Database Preservation Evaluation Report - SIARD vs. CHRONOS

Preserving complex structures as databases through a record centric approach?

Andrew Lindley
AIT - Austrian Institute of Technology GmbH
Donau-City-Strasse 1, 1220 Vienna, Austria
andrew.lindley@ait.ac.at

ABSTRACT

Preserving information systems is one of the greatest challenges in digital preservation. In this paper we outline the existing strengths and shortcomings of a record-centric driven preservation approach for relational databases by lining up a state-of-the art industry database archiving tool CHRONOS[1] against SIARD[2] one of the most popular products in the GLAM (galleries libraries archives museums) world. A functional comparison of both software products in the use cases of database retirement, continuous and partial archiving as well as application retirement is presented. The work focuses on a technical evaluation of the software products - organizational and process aspects of digital preservation are out of scope. We explain why preserving complex structures as databases through a record centric approach does not only depend on the amount of information captured in the preservation package and present a brief overview on available functional aspects in CHRONOS that help to address the challenges of application decommissioning. The paper at hand presents the results of a case study which was undertaken 2012 at AIT - Austrian Institute of Technology GmbH.

Categories and Subject Descriptors

H.2.m [**Database Management**]: Database Applications—*Miscellaneous*; D.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*System issues, User issues*

General Terms

Verification, Experiment, Performance, Reliability, Management, Human Factors

[1] `http://www.csp-sw.de`
[2] `http://www.bar.admin.ch/`

Keywords

Digital Preservation, Database Archiving, Case Study, Technical Evaluation, Decommissioning, Application Retirement

1. INTRODUCTION

Sustained information to our scientific and cultural heritage world is stored digitally. The term digital preservation (DP) summarizes methods and techniques to secure long-term access to digital information. Every information management system, data warehouse, or even the simplest online webstore is backed by a database system. For the last decades relational databases have been the dominant technology in this area mainly due to broad vendor adoption and acceptance of the SQL standard for the relational model. ACID (Atomicity, Consistency, Isolation, Durability) provides principals governing how changes are applied to a database. In the decade of big data some of these principles are loosened with respect to high data volumes and high traffic throughput and niche products as NoSQL databases, key value and tripple stores found their place.[1]

Within the last ten years the digital preservation community was able to achieve a solid understanding of issues and provided solutions and guidance in the domain of document preservation. The currently ongoing European initiatives widen the domain of digital preservation taking on from memory institutions and include scenarios such as healthcare, data with direct commercial value and web-based data and focus on aspects such as data collection, scalability, reconfigurability and lifecycle management. [2], [3]

Preserving information systems is one of the greatest challenges in digital preservation. The paper at hand presents the results of case study which was undertaken 2012 at AIT. The technical evaluation and comparison of the database preservation tools SIARD [4] and CHRONOS [5] targeted at the use cases of database retirement, partial and continuous archiving as well as application retirement. Besides presenting a functional tool comparison we highlight strengths, shortcomings and white spots in general. A goal is to broaden the discussion on database preservation by comparing one of the most popular tools for database preservation in the GLAM domain against CHRONOS a leading industry product. This work focuses on a technical evaluation of the software products and only briefly covers organizational and process oriented aspects of digital preservation. CHRONOS is a commercial product owned by CSP

and emerged through a joint research cooperation between the department of computer science at the university for applied science in Landshut. SIARD (Software Independent Archiving of Relational Databases) is owned by the Swiss National Archives (BAR) and is both an open format to express relational database archives as well as a software product SIARD suite. It is available under closed-source license and was originally developed by Trivalis.

1.1 Continuous Archiving and Application Retirement

Solutions for database archiving are not part of a standard relational database systems. According to Forrester [3] only 15% percent of business data are actively required to serve a company's day-to-day requirements while the vast amount of data could already be moved into an archived state.

> [..]Terabyte-size transactional databases are harder to manage, increase costs for hardware capacity and database licenses, and drive up requirements for database administrators (DBAs). Yet 85% of production data is inactive, so information and knowledge management professionals should devise a database archiving strategy that moves inactive data to lower-cost storage and servers, thus improving the manageability, performance, and security of critical production applications[..]

A typical data life-cycle can be categorized in an

a) Active State, in which data is generated and modified as part of the production system

b) Archiving State, in which a dataset is no longer altered but still needs to be kept active for fulfilling existing business processes

c) Long-Term Archiving state, in which only selected parts of a dataset are kept for retention

Effects that are achievable with a continuous database archiving strategy are for example the reduction in database license fees, easier adherence to SLAs, efficient system consolidation or a noticable reduction of maintenance effort. Such a system can also be constructed to adhere to different requirements as for example given legal regulations on data retention.

Preserving complex structures as databases through a record centric approach does not solely depend on the amount of information captured in the preservation package but requires a surrounding process to capture all required metadata as additional documentation and understanding of the underlying data. This is shown by a case study of the National Archives of the Netherlands [6] in 2011 on longterm preservation of relational database systems in coherence with the legal mandate to archive public data records (content, reports, applications) and records from public institutions. They came to the insight that even though an acceptable number of available tools and technology was available to address the challenge, there was a lack on sufficient knowledge on the relevance of the archived data and its contextual relationship within given business processes.

Decommissioning is the process of a planned shut-down and removal from operational use. Decommissioning as well as application retirement are challenges that an archive or library is confronted with. In the ECM podcast [7] on practical digital preservation Adrian Brown, director of the Parliamentary Archives in London mentions the 'blurring of the boundary' between digital objects and the applications that they are held in as key challenge the institutions are confronted with. Digital preservation initiatives and projects made great progress in tackling the problem of how to preserve the file formats and the objects themselves but now faces the more complex problem of how to preserve the information that an application has about the objects it holds? How to enable digital objects to move from one application to another without losing that information? Within this paper we present technical issues and generic challenges we discovered when transforming a database into a long-term database archive by using the tools SIARD and CHRONOS and conclude with a brief overview on the features that CHRONOS is able to deliver for the application retirement scenario.

1.2 Paper Outline

In the first part of the paper we raise relevant research questions and point out existing limitations of a record-centric driven preservation approach for relational databases. In the second part we present the experiment setup and detailed evaluation results in a functional comparison. We conclude with a brief overview on functional aspects provided by CHRONOS for the challenges of application decommissioning.

1.3 Related Work

Burda et al.[8] present a semantic literature review of 122 publications in the domain of digital preservation with respect to different aspects as drivers, stakeholders and applied research methods in the field. The authors disclose the gap of a DP reference model that addresses organizational concerns, considering aspects such as costs, risks, decision criteria, etc. The ISO standard 'Reference Model for an Open Archival Information System' (OAIS) which guarantees cross-organisational concepts and terminology has impact in the construction of a preservation package and the 'Model Requirements Specification for the Management of Electronic Records' (MoReq2) which provides principles to guide institutions in the implementation of electronic record management systems are both seen as relevant in the domain of database preservation. Preservation Planning Tools as PLATO [9] support the process of cost-benefit analysis within digital preservation decision making but to the author's knowledge no case study on database preservation was conducted to date. Digital preservation projects co-funded by the European Commission under the sixth and seventh framework programs are given in [10] which presents objectives, developments and major outcomes of the projects. The intellectual property rights of SIARD lays at the Swiss Federal Archives and development was stimulated through

[3]http://www.forrester.com/Database+Archiving

the Planets project [11]. A different approach than extracting and describing relational data through generic and vendor independent XML formats as DMBL[12] or SIARD for archival and cross compatibility purposes is the preservation of relational data through RDF triples as implemented in the Semantic Archive and Query (SAQ) system where access is provided via A-SPARQL queries.[13]

Preservation of databases and database records has always been an an important task for national archives which in many cases is based on a legal mandate to preserve governmental records. Activities in this area for the Danish National Archives started in 1973. In 2008 all of the approximately 3.600 Archival Information Packages (AIPs) held in their collection were exports from database systems, whereby content from both business systems and record management systems are transferred as relational databases. The Access project was completed 2008 and since September 2010 archival records, which are structured according to the Danish archival standard for digital records are delivered in a modified version of the SIARD format which also includes contextual documentation. A general query building system for archival records has been developed to support unknown needs for retrieving data. [14]

2. EXPERIMENT SETUP

Work presented in this paper is based on a case study which was undertaken by AIT in 2012. The report is split into three major sections, a generic evaluation of the underlying tools and their technical features, a ISO 25010:2011 driven evaluation of software quality aspects based on ISO/IEC TR9126 'quality in use' metrics in the areas of efficiency, productivity, security and satisfaction within a very specific usage context and staging environment, and finally an interpretation of the research results based on the customer's requirements. Please note that part two and three of the report itself are confidential as they contain customer sensitive information and therefore are not presented in this paper. Aspects as licensing or pricing information from part one of the report which are protected by NDA agreements are also left out.

Database preservation strategies heavily depend on the nature of the underlying data where typically three main categories are distinguished: administrative, scientific and document management databases[15]. Part one of the tests which are presented within this paper were executed on a virtualized standard desktop hardware infrastructure running Windows-7 with a local copy of the tools and all required software dependencies together with an Oracle 11gR2 database filled with Transaction Processing Performance Council (TPC)-C "'Entry-Order"' records that were enriched with BLOB and CLOB data. The aim is not to provide benchmark information but rather accompanying documentation on technical features and unique selling points - no entitlement of functional completeness.

3. EVALUATION RESULTS

A quick overview of the product driven evaluation results is given in Table 1. More detailed explanations on the individual items and resulting issues are given within this paper.

Evaluated Categories	Siard	Chronos
Supported Preservation Scenarios	3/10	8/10
Exported Elements of an Archived Database	6/10	8/10
Pre- and Postprocessing via Database Scripts and Markertables	5/10	10/10
Data Retention and Data Controls	1/10	10/10
Support of UDTs and Oracle Specifics	3/10	5/10
Rights, Roles and User Management	0/10	9/10
Archive Data Access and Performance	2/10	10/10
Syntactic and Semantic Data Changes	0/10	9/10
Existing APIs and Interfaces	3/10	8/10
Scalability and Limitations	7/10	9/10
Risk Behavior and Dependencies	9/10	8/10
Referential Dependencies	3/10	10/10
Standard and Compliance	4/10	4/10
Data Exchange Formats	5/10	5/10
Structure, Setup and Size of the physical Archive	7/10	7/10
Specification of Information Lost	3/10	3/10
Installation and Delivered Components	10/10	9/10
License Models, Costs and Reference Customers	5/10	5/10

Table 1: Overview of the Product Driven Evaluation Results

Supported Preservation Scenarios

The evaluation is based on the support of the three classification scenarios: 'database retirement', 'ongoing or partial database archiving' and 'application retirement'. Questions addressed are to which degree do the products offer support for database retirement (including database independent transformation, understandability of the physical archive, SQL data access, etc.), continuous or partial archiving (inc. schema changes over time, data retention, etc.) and application retirement (incl. available support for the recreation of business objects, functions as reporting, data access roles and programmatic access, etc.).

CHRONOS is able to deliver an extensive package with support for all three database preservation scenarios. Especially 'database retirement' and 'continuous/partial archiving' are seen as core use cases which are covered out of the box in the requested and required complexity. A key feature of CHRONOS regarding data access is the possibility to execute SQL92 compliant reporting through queries on top of the archived datasets. Even though the content is exported and physically stored in basic text files the query performance is comparable to the one of a relational database. The scenario of 'application retirement' is backed through the Chronos software module Archive Explorer that allows recreating relevant business objects, custom views and reporting workflows based on the archival records. All modules adhere to data access and role policy models. CHRONOS software suite can in addition leverage positive secondary effects as quicker backup and restoration time, as an easy way of generating snapshot data, performance improvements within the production database and reduction of storage and licensing costs as typically database system are licensed by number of cores.

SIARD is defined to fully support the 'database retirement' use case for a huge number of relational database systems. Support for the scenarios application retirement or continuous / partial archiving are not envisioned for the SIARD Suite. Even though it is possible to re-import a SIARD database archive by restoring its primary tabular data into a RDBMS in order to execute complex queries and even though it is possible to manually rebuild or ignore the lost metadata such as views, procedures, triggers, etc., the system it is not meant to re-vive a database for continuously exporting data.

Exported Elements of an Archived Database

A relational database and RDBMS is a complex product that technically speaking consists out of Tables, Views, Materialized Views, Indices, Packages, Triggers, Stored Procedures, Functions, Sequences, Scheduler, Check Constraints and Triggers, Queues, Database Links, User Management Access Privileges and Roles to mention the most important constructs. Which database elements are extracted into a database archive by the preservation tools at hand? Which of these elements remain functional after re-importing them into a RDBMS and which of them solely serve the purpose of documentation within an archive?

The main focus in CHRONOS lays on exporting primary data and datatypes. Tables, Views, Indices, Packages, Procedures, Functions, Triggers, Sequences, Materialized Views, Scheduler and Check Constraints are supported elements when transferring data into a database archive. Queues are not preserved as they only serve for communication purposes and no value is seen in keeping them. Database Links are not supported. Jobs are deprecated and are not archived by CHRONOS. *User management and definition of roles are not preserved by CHRONOS as there is no access mechanism through standard interfaces.* In many cases user and rights management however is not depicted at database level anyway. On a functional level CHRONOS offers extensive support for integrating with central policy and access permission systems as LDAP. *Triggers, Procedures and Views are exported from the production system but remain unsupported elements when re-importing the archived data into a RDBMS.* This can be seen as a security feature as cross mapping between different database vendors and also between versions of the same product (e.g. Oracle version 10 and 11) would have the potential to cause serious inconsistencies.

SIARD exclusively supports archiving of core SQL:1999 elements. Procedures and Functions are minimally supported and documented in a SIARD archive, depending on accessibility of the pertinent metadata information. The tool does not support functional long-term preservation of code but concentrates rather on preserving primary data. Triggers are supported by the SIARD format as they are defined in SQL:1999 but are not archived by SIARD Suite as they are only seen useful for 'live' databases where activities occur that trigger them. Materialized Views are not defined in SQL:1999 and in most database systems they are just (temporary) tables. Check Constraints are supported by the SIARD format as they are defined in SQL:1999 but usually are not archived as they are not easily accessible in most database systems. Users and Roles are archived by SIARD Suite. Standard 'scalar' SQL data types (Strings, Numbers, Dates) are supported by SIARD. User-defined data types (UDTs) at the moment are not archived, because no real life database system supported them when the design and development of SIARD started. There are plans to enhance the SIARD format to accommodate UDTs, however backward compatibility between the different versions of the SIARD format is a major requirement! Database links and packages are not supported. Packages are not defined in SQL:1999 and are not supported by all relational database systems. Indices are not supported, as indices in SQL:1999 are not defined as database elements but only serve as performance enhancers. Also Queues and Sequences are neither defined in SQL:1999 nor supported by all relational database systems or SIARD. *When re-importing a SIARD archive into a RDBMS, solely tables and tabular content is restored. Constraints are attempted to be restored as primary and foreign keys. Views, procedures, users, triggers, and check constraints are not restored as they could cause problems between different database instances.* From a SIARD perspective views, procedures, triggers, etc. are just considered as metadata. This information is therefore only depicted within the metadata.xml file, which is located in the header and not in the content folder for primary data. SIARD concentrates on restoring primary table data in RDBMS for the purpose of executing complex queries on it.

Pre- and Postprocessing via Database Scripts and Markertables

In the process of creating an archival package, especially in the scenario of partial and ongoing archiving, it might be necessary to execute pre- and post processing steps on the database as for example preparation or cleanup tasks. Supporting a smooth and integrated continuous archival workflow might require logging some kind of state or placings process markers within a production system. To which degree do the tools offer support for interacting with a production environment as executing pre- or post processing scripts or documenting archival state within the database itself?

CHRONOS allows to directly interact with a database system via shell commands, database scripts and marker tables. Documentation within a database system is possible via marker tables at the granularity of individual records. *SIARD, by design, never writes to a database and can therefore be executed with read-only permissions.* In the SIARD archive date and the circumstances of the download are recorded. SIARD Suite does not directly support pre- or post-processing of database scripts as this is highly dependent on the database system in use. Due to the fact that the SIARD Suite not only provides a GUI application but also supports the command-line interface for up- and download of archives, there is a workaround for calling a script or batch file via sqlplus for static pre- and post processing.

Data Retention and Data Controls

Due to legal regulations for example on handling of personal or sensitive data it might be required to keep and/or delete records after a given period of time from an archive. Other forms of data retention concern the periodical refreshment of expiration dates. The following questions are taken into account: Do the systems easily allow to classify and separate

archival data from master data items. Which mechanisms are in place to handle data retention and deletion controls and at which degree of granularity. Is it for example possible to connect to external storage systems that ship with built in mechanisms for data retention? Which security mechanisms for supervising deletion control mechanisms are in place?

CHRONOS ships with modules for creating archival data retention policies and fully applies to the requirements of implementing legal hold within a repository. There are mechanisms in place for interacting with database environments themselves but primarily data retention policies are enforced on the exported data. The software allows to enforce retention and deletion policies across different storage media and provides a central interface for maintaining distributed archival packages. CHRONOS offers adapters for interacting with dedicated storage facilities as for example provided by EMC[2]. The system not only takes advantage and closely integrates with these advanced storage technologies but also provides retention mechanisms for standard file volumes which don't offer out of the box capabilities for defining update strategies, expiration dates, etc. The degree of granularity on which the system is able to operate upon is a single archival package. The actual process of marking data for deletion and enforcing the physical deletion of data from the media is a two step process and is safeguarded by human approval with dedicated access rights.

SIARD, by design, exclusively offers support for the database retirement scenario. All information the application is able to access within a database gets archived and it is up to the user to provide adequate visibility and access right policies to the targeted data sets via the database's management component. SIARD never writes or deletes information to or from a database system as it is executed with read-only access permissions. All information is written to the standard file-systems with no SIARD internal support for data retention or different storage connectors. Data integrity as written to the file system is guaranteed by the SIARD-Suite, but it is up to the archivist to take care of everything beyond.

Support of UDTs and Oracle Specifics

Clarifies the degree of support for custom Oracle database features such as user-defined datatypes (UDTs) or Oracle specific extensions as PL/SQL, Oracle Spatial and custom built applications with Oracle Forms.

CHRONOS is able to archive cascading Oracle user-defined datatypes in a preliminary form and CSP has announced further support for upcoming releases together with comparable constructs of other database vendors. However UDTs are seen problematically given their inconsistency and incompatibility across different versions of Oracle databases. UDTs are only available for current Oracle product versions and only when the JDBC driver offers support, no cross vendor mapping is possible when re-importing archived data. To gain performance in Oracle it is possible to temporarily disable the checking of foreign key constraints when importing a large datasets. This state is not reflected in an exported CHRONOS archive and therefore falsely enabled as active foreign key when re-imported. *In the process of data export CHRONOS makes use of native dialects for querying the individual database systems.* CHRONOS itself delivers a SQL92 interface for running queries on archived data. *Procedural Language SQL (PL/SQL) is neither supported for querying nor for archival purposes.* Additional Oracle specific extensions as Oracle Spatial are currently not supported. For form based applications such as created through Oracle Forms CHRONOS is able to act as middleware through its provided APIs.

SIARD supports standard 'scalar' SQL data types (Strings, Numbers, Dates). There are *plans to enhance the SIARD format to accommodate user-defined data types (UDTs) in a SQL99 standardized way*, in order to fulfill backward compatibility requirements of the SIARD format. There are no plans to further support other Oracle-specifics with one exception, the export of Oracle table and column comments as metadata comments. SIARD's product design focuses solely on standardized data content, which in a SIARD understanding is the only amenable way to long-term preservation. Additionally, vendor lock-in of any kind is avoided in this way.

Rights, Roles and User Management

Access controls and user management is a core component of a running database environment. This section focuses on the capabilities of the tested database archiving products to offer rights, roles and user management functionality on top of the extracted database archive.

CHRONOS delivers a mature user, rights and access management layer out of the box. It is tightly integrated throughout all delivered CHRONOS components and is highly customizable to individual needs. Integration of central user management systems like LDAP is possible. The provided level of granularity allows to protect sensitive data in the archive at the level of database columns.

SIARD itself does neither provide user, rights or permission management nor custom application views within the user interface on top of the underlying data but rather makes extensive use of the underlying RDBMS user management component. Visibility and access rights of the archiving user determines the scope of harvested data as SIARD performs a full database export of all 'seen' objects.

Archive Data Access and Performance

One of the core features of CHRONOS is the system's possibility to execute SQL statements directly on top of the archived data located on the file system with performance measures comparable to those of standard database systems. *To overcome the bottleneck of finding, accessing and processing archival packages from the file system CHRONOS makes use of a hybrid approach of a custom SQL92 interpreter, global search index and a local BTree index on column level, as well as H2 and hsqldb in-memory database systems for SQL JOIN operations.* SQL queries without pre-processed indices i.e. a full archive search, are possible but not very performant therefore the data selection for pre-indexation is essential. A core parameter for adjusting performance in CHRONOS is the archive package split size. This allows to decide how to allocate tabular content into different physical

zip containers. Finding an optimum balance between data access and search is highly use case dependent.

With the CHRONOS database archiving product suite it is possible to both create a database export in a vendor independent generic format that from a technology point of view does not contain any crucial dependencies but roughly just data and corresponding schematic structure. At the same time the software suite delivers added value on top of the physical archive which is crucial for the management and use of such information. For example performing queries over different revision of data, i.e. search operations on content at a given structure and point in time which are performed directly upon archival packages on the file system, expressed in SQL92 and in performance that we're used from database operations. And all without having to re-import and revive archived data in a dedicated database environment and even if in the meantime modifications on the database schema have been undertaken.

SiardEdit is a graphical user interface application for exploring SIARD archive files. *SiardEdit is the central instrument with which SIARD formatted data is processed.* It allows to display, sort and browse primary data in a SIARD archive and to add to or change the archival metadata. Primary data cannot be changed. However the tool is not suitable for complex research or research within large archives. In this case it is recommended to load a SIARD archive into a database system and use database techniques for exploration.

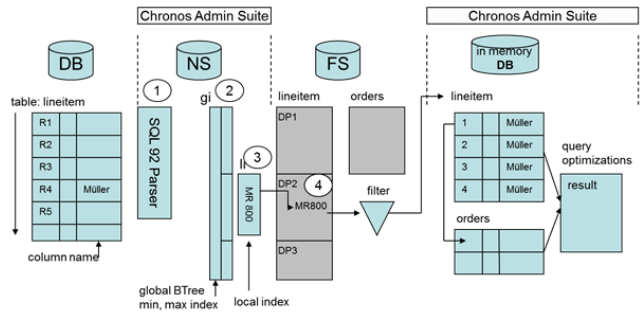

Figure 1: Simplified scenario of a SQL JOIN operation between two tables within CHRONOS depicting the interaction between the custom CHRONOS SQL92 parser, BTree indices for data retrieval, in memory databases for the JOIN operation. Only the indices requires to be on a near storage (NS as disk) next to the search server to execute the query, the actual archival data may be distributed across multiple backends and far storage units (FS as tape).

Syntactic and Semantic Data Changes

In the case of continuous archiving partial datasets remain within the production environment. Therefore a common scenario which needs to be dealt within is the reaction to syntactic and semantic changes over time. Which form of support or traceability do the systems provide for this kind of temporal changes?

Structural changes in the schema as adding additional columns, are automatically detected by CHRONOS. Data is exported into a separate revision and for more complex changes the user is given tools to administrate them. When running a query against a given revision CHRONOS only takes the structure and data into account which was present at that time. *Semantic changes always require manual treatment as there is no way for detection. CHRONOS offers support to automatically transform deposited data via customizable operations for an entire revision.* Those script based operations are written in Java and allow to use the full richness of the JDK for data manipulation. The actual physical content within the long-term archive however stays authentic and untouched as semantic transformations are only reflected within the CHRONOS middleware. The content of a given revision is therefore always properly and consistently reflected on the file system in the state it was extracted from the original database. Duplication of data between revisions is deliberately accepted.

SIARD cannot be evaluated against this use case as it exclusively offers support for the database retirement scenario. The tools is neither designed to cope with semantic or syntactic changes of the underlying data nor does it provide support for handling modifications in the archived packages within SIARD Suite.

Existing APIs and Interfaces

The scenarios archiving, data access and search were evaluated with respect to available programming interfaces.

All of the CHRONOS server modules offer programmatic access via JDBC, Java RMI and web-services and allow deep system interaction. JDBC drivers provide unified access to database systems out of a Java environment. CHRONOS provides direct access to previously archived content on the file system through a JDBC class 4 driver and therefore allows to easily select and process data. Data manipulation is not possible via JDBC. From a functional point of view available interfaces have been tested to programmatically support the entire process of setting up and running a database export and re-importing a CHRONOS archive into a database system. The range of available interfaces differs depending on licensing. Beyond this there is out of the box support for a variety of external facilities such as job schedulers as crontab or taskmanager, storage solutions like EMC^2 centera.

SIARD is a generic platform-independent JAVA program that achieves a lot of independence from the individual database system by being bound to the JDBC interface. As interfaces to SIARD Suite the two command line applications SiardFromDb and SiardToDb are provided for extracting a database archive within the SIARD format or vice-versa. Although the applications' functionality is identical with the functions available via SiardEdit it is recommended using the command line versions especially when downloading large databases as they are designed for scalability. All settings for those tools can be provided via a configuration file, so using the two applications within scripting solutions allows to achieve a certain degree of automation as e.g. scheduling via cron jobs. All surrounding dependencies need to be configured externally.

Scalability and Limitations

This point takes into account scalability aspects as size, throughput, access time as well as any form of limitations that could influence the products usage as hardware and software prerequisites.

CHRONOS is a product which in all aspects is designed to deliver performance and scalability via Java multithreading. In our testing environment with standard hardware running 4 CPUs and 6 GB of RAM we were able to constantly export two thousand tuples per seconds from the database even running the archival packages indexing operations aside. The bottle neck in this case was the performance of the underlying local file system. Scenarios with limited memory resource allocation of the Search- and LocalIndexJobs can noticeable bring down the response time of the system whereat only 128 MB of assigned Java heap memory still were sufficient to properly execute operations on the SQL search server without any erratic behavior.

Both CHRONOS and SIARD are self documented archives of primary data. External documentation, artifacts, process documentation, approval or decisions taken are not part of a created archival package even though this information is partially available through the software suite. Due to integrity checks of the archival zip packages it is not possible to add this information externally.

The ZIP 64 standard accommodates files with sizes up to 18'446'744'073'709'551'616 Bytes (i.e. 16 Exabyte). *SIARD uses ZIP 64 without compression to generate a one-file container for the archived database and is therefore limited by this size.* The SIARD Suite runs within a JVM of typically 500-2000 MB of heap space. It uses the heap space for holding all metadata in memory as well as one row of data. This JVM setup has been sufficient for any database tested. SIARD does not make use of JAVA multithreading or multiple DB sessions due to the imposed number of integrity problems! While the Java Swing application SiardEdit had memory problems when downloading a large number of items, the provided command line applications showed consistent performance.

Risk Behavior and Dependencies

Whats the degree of underlying dependencies for a given database archive in subject to system dependencies, vendor / tool locking, or similar objectives?

Both tools follow the approach of clearly separating the composition and description of the data structure from the actual primary data - this is also reflected on file system level. CHRONOS describes the structure in XML and provides a fully interpretable XSD schema file while the content itself is stored in a delimiter file. In theory all information to properly read and interpret this data and therefore possibly manually revive it into a RDBMS in case of a vendor crash is available without any direct dependencies. In practice this step is non trivial and not possible out of the box without a previous data transformation process due to the fact that both SIARD - which makes this fact implicit by proposing a central data exchange format and representation based on SQL99 - but also CHRONOS require a mapping between their internal form of data representation and the corresponding database datatype configuration and mapping. While within SIARD this commitment and scope is based on SQL99 datatypes to guarantee a full round-trip scenario, CHRONOS explicitly documents the supported datatypes for every vendor and database version but however treats the cross-vendor and inter-version representation as industrial secret.

A main aspect in digital preservation is to keep the stack of software dependencies as low as possible. For CHRONOS they can be mainly summarized as Java + JVM, XML and Zip32 Deflate. The zip compression deflate is public domain and widely used as for example within the Portable Network Graphics (PNG) or ISO Open Office XML-Format. Additional system configurations, documentation regarding the technical approval processes as the underlying user, role and rights management are not part of an archival package but partially are reflected in the applications settings in XML form. It should be possible to enable manual database recovery within a fair amount of time.

Referential Dependencies

In many cases the database does not contain full referential integrity as this is often depicted by external documentation or reflected within a different software layer. In some use cases it many be required to export a given dataset including all referential dependencies? *CHRONOS allows to automatically detect referential dependencies for master tables and has tools to decide how to deal with cyclic references and to with depth those references need to be respected. External dependencies can be remodeled.*

SIARD has the ability to archive an entire database, but without the possibility of selecting individual tables. However the 'entire database' refers to the collection of all objects that the database user which is used to export the archive has read access to. Therefore to exclude certain tables from the archival process the only additional step required is to create a database user with specific read access rights limited to the tables that should be archived. All foreign keys are resolved if the SIARD file was generated from a database which had constraints enabled. SIARD does not censor data or ensure integrity.

Standards and Compliance

Currently there is no standard in the field of long-term archiving for databases. The SIARD format has become a widely accepted format for the exchange of relational database content within GLAMs.

Is there a chance for an SQL standard for Archiving, based on a subset of the ISO-9075-SQL, similar to the PDF/A for archiving? To increase acceptance by vendors the SQL standard defines three levels of conformance and implementation: entry, intermediate and full level. The mandatory part of SQL99 is called core and is described in part 2 (foundation) and part 11 (schemata) of the standard. Since most RDBMS are based on SQL and most vendors claim compliance with the standard one should assume that relational database definitions are independent of any specific RDBMS product. Unfortunately this is far from the truth. Even though the SQL standard today comprises over 2000 pages it is far from being fully self-contained. In contrast,

SQL99 explicitly identifies 381 so called implementation-defined items. Most of today's RDBMS implement (and sometimes faultily) only the core and the entry level of the standard completely. To this often large number of non-standard, product-specific enhancements are added which leads to many different SQL flavors. *SIARD Suite currently adheres to SQL:1999 "Core Features" in terms of supported functionality and mapping of data types.* Future versions may be extended to make use of additional SQL99 components as Packages.

The OAIS model is a reference model for a repository where a SIARD archive would be a Digital Object held within an OAIS repository. The SIARD archive therefore is not a stand-alone single file that can be though of as an AIP. A SIARD file should be treated as a single object – like a word file – in an archival system, which itself may or may not adhere to the OAIS model. It is assumed that a retired database in the SIARD format is archived as part of a larger archive package with additional documentation. In the case of SIARD, it is important to separate the discussion of the format from the discussion of the tool. The formats huge advantage is that it is solely based on existing international standards and independent from any single database vendor or the specific infrastructure of a particular customer. The SIARD tool has more disadvantages. It makes assumptions and decisions about the mapping of real live databases to the standard. These may be questioned. However, this is not a failure specific to the SIARD software. The author is not aware of any tool that explicitly guarantees the preservation of primary data values and idempotent up- down and -uploading. The tool creators have made the decision to prefer moderate performance over database or operating system dependence. The existence of this "reference implementation" does not prevent the implementation of other solutions with higher performance or even with vendor lock-in.

Structure, Setup and Size of the physical Archive

The Transaction Processing Performance council database dump was used to get measures and comparison on the physical size of an exported database archive. Not taken into account in this comparison are parameters which are built up within a database environment that are not easily uniquely assignable. The size of the original source of a database is not a defined value i.e. there is no measurement on the size of an Oracle schema or database index in bytes?

While a SIARD archive required +338% on disc space compared to the database dump a CHRONOS archive is able to decrease the required space by -41%. This comparison took into account operational artifacts which are understood and processable by SQL-Developer, SIARD-Suite and CHRONOS Administration Suite. SIARD uses a zip container but does not apply any compression algorithm. By applying a post-compression (deflate, 32K word size, standard compression) the size of a SIARD archive can be brought down by 30%.

For CHRONOS in average we measured a 40-60% reduction in required file size compared to the database dump depending on the underlying tabular data. Further room for improvement lays in the use of different checksum algorithms. MD5 is applied out of the box and tends to blow up small records. As the 32-Bit version of a zip container is only able to support container file sizes up to 2 GB the system splits up archival packages. Per default 20 MB is the standard package split size which also shows the best performance stats regarding searchability, indexing and query response time. The file structure of an exported database archive within CHRONOS separates the actual tabular data from its structural description. Partial retirement scenarios are built up based on temporal events, either static or based on temporal markers within the database. Elements as Binary Large Objects (BLOBs) or CLOBs are stored in separate clusters of binary objects within the archive and are referred to via data pointers in the tabular data. Data integrity against the original is checked by the system after harvesting as well as after moving the archival data into the storage component.

For primary data SIARD chooses to use XML short tags. In our TPC-C test data we were able to notice a factor of 1:3 of increase in data size. According to SIARD's official statements the size of the SIARD file should be similar to the size of the Oracle dump from which it was downloaded, if the primary data represents the majority of information. Even though zip64 packaging is used to create the container file, no compression algorithm is applied to avoid any dependencies for the long-term. The SIARD format is not configurable in the sense of being able to add additional fields. The Swiss Federal Archive feels that an international standard is better served by uniformity than by flexibility. The 'technical metadata' describing the database structure is dictated by the SQL standard. Once a SIARD archive is exported its consistency with the underlying database's data is verified and the number of archived records is documented. Any modification to the database during the process of creating the SIARD export leads to an error in the exporting process. Regarding SIARD's structure on the file system, all database contents such as schema definitions and primary data are stored in a collection of XML files which conform to the SQL99 definition. The only exceptions are binary large object (BLOB) and character large object (CLOB) elements which allow holding larger sets of data. These are stored in separate binary files having referential pointers in the corresponding XML entries. Data is stored in a Unicode character set. While extracting databases that support different character sets, a mapping to the corresponding Unicode characters is carried out. For this reason, SIARD generally translates national character string types in the database software (NCHAR, NVARCHAR and NCLOB) into non-national types (CHAR, VARCHAR and/or CLOB). This convention is well supported by XML and independent of whether an XML file is stored in the UTF-8 or UTF-16 representation. Characters with a special meaning to XML are substituted by entity references in the SIARD archive files. If a string is longer than 4000 characters then „clobType" and „xs:string" are replaced by an external reference to a text file. If a binary array is longer than 2000 bytes then „blobType" and „xs:hexBinary" are replaced by an external reference to a binary file. Characters that cannot be represented in UNICODE as well as the 'escape character' and multiple space characters are escaped as 0̆0<xx> in the corresponding XML, 'greater than', 'less than' and ampersand characters are represented as entity references in XML.

Specification of Information Lost

Which audit trail capabilities does the system offer for logging and tracking modifications over time. Is there a way of specifying the amount of information lost when exporting data into a long-term archive? One example on a measure which could be applied is the Oracle SQL Minus operation after re-importing a database archive to determine the correct structure and item count against the original data.

The amount of information available in the database's metadata is debatable and cannot be quantified. Both SIARD and CHRONOS can be classified as idempotent in terms of that an upload – download – upload produce delivers the exact same data types and values on the second upload as on the first one. Checking this idempotence is part of the SIARD build script. However there is no support for statements that declare what information is actually lost during export (as e.g. UDTs, disabled Oracle foreign key constraints, etc.), lost during cross database or cross db-version re-import or lost by a mapping from the native type to SQL99. Both CHRONOS and SIARD support program logging with various log levels to track down system behavior but no persistent logging of the history of changes is implemented. SIARD per definition does not support schema changes or ongoing database archiving over time and takes an archived/retired database as a final and unmodifiable constructs there is no need for data modification audit trails or similar tracking features at this level. Features like these are more in the realm of the enclosing archival process/system and therefore a feature which one would expect in system like CHRONOS.

4. CONCLUSION

Archiving databases either means preserving information or preserving functionality or both, so the tools SIARD and CHRONOS were evaluated within the scenarios of database retirement, continuous/ongoing and partial retirement as well as application retirement. In the underlying case study both tools proved stable and technically mature in creating a database archive in a vendor independent long-term preservation format for a rich number of relational database system. The tools proved mature and were able to deliver solid performance. There are small differences in the number of supported database vendors, SQL elements and the internal data representation. A clear recommendation which product the community should adopt is almost impossible as the supported scope and use cases both tools are able to deliver are highly diverse. SIARD suite was designed as reference implementation for the SIARD format and exclusively offers tool support for the use case of database retirement. CHRONOS on the other hand, a commercial product well designed for scalability and industrial needs, provides a rich set of tools and end-user applications that allow both to export a physical database archive and to operate on top. CHRONOS provides all mandatory bits at the required level of complexity to accomplish the challenges of the ongoing/continuous and partial archiving scenario. Core features include running SQL92 queries on top of the archived data with database like performance, support for revisions, syntactical and semantical schema modifications, resolving cyclic dependency, external referential integrity handling, a full blown access control and data retention layer, etc. Shortcomings of CHRONOS are the limited support of complex objects as Oracle UDTs and lacking support of audit trails for classification and documentation of information lost. Besides the core functionality CHRONOS provides support for the use case of application retirement with tools that allow re-modeling of business objects, application logic and reporting functionality and by being able to directly serve as middleware layer for legacy applications. The rich set of programmatic interfaces allows both to integrate with most of the system's functionality as well as to grant access to data via standard mechanism as JDBC. Finally we presented examples why preserving complex structures as databases through a record centric approach does not only solely depend on the amount of information captured from a database itself but why it is important to create full preservation packages which cover contextual information.

Acknowledgments

The author would like to thank Mario Günther, Mario Täubler (CSP GmbH Co. KG) and Thomas Hartwig (Enter AG) for their input on the underlying study. Their comments, written feedback and interviews helped to clarify open issues in the process of evaluating CHRONOS and SIARD. Any remaining misinterpretations or mistakes are those of the author.

5. REFERENCES

[1] Agrawal, R., et al.: The claremont report on database research. SIGMOD Rec. **37**(3) (September 2008) 9–19

[2] Edelstein, O., Factor, M., King, R., Risse, T., Salant, E., Taylor, P.: Evolving domains, problems and soluations for long term digital preservation. iPres (2011)

[3] Schmidt, R.: An architectural overview of the scape preservation platform. iPres (2012)

[4] Heuscher, S., Stephan, J., Peter, K.M., Frank, M.: Providing authentic long-term archive access to complex relational data. CoRR (2004) DL/0408054

[5] Brandl, S., Keller-Marxer, P.: Long-term archiving of relational databases with chronos. First International Workshop on Database Preservation (March 2007)

[6] van Essen, M., de Rooij, M., Roberts, B., van den Dobbelsteen, M.: Database preservation case study: Review. IST-2006-033789 Planets Deliverable PA/6-D13 (2011)

[7] Brown, A., Lappin, J.: Ecm talk 17: Practical digital preservation (2013) `http://traffic.libsyn.com/ecmtalk/ECM_Talk_017.mp3`.

[8] Burda, D., Teuteberg, F.: Sustaining accessibility of information through digital preservation: A literature review. Journal of Information Science (2013) 1–19

[9] Becker, C., Rauber, A.: Decision criteria in digital preservation: What to measure and how. Journal of the American Society for Information Science and Technology (JASIST) (2011)

[10] Strodl, S., Petrov, P., Rauber, A.: Research on digital preservation within projects co-funded by the european union in the ict programme (2011)

[11] Farquhar, A., Hockx-Yu, H.: Planets: Integrated services for digital preservation. International Journal of Digital Curation **2**(2) (2007) 88–99

[12] Rammalho, J.C., Ferreira, M., Faria, L., Castro, R.: Relational database preservation through xml modelling. Extreme Markup Languages (2007)
[13] Stefanova, S., Risch, T.: Scalable long-term preservation of relational data through sparql queries. Semantic Web Journal
[14] The Danish State Archives: Symposium about the transfer, preservation of and access to digital records based on the danish experiences (2008)
[15] Ribeiro, C., David, G.: Database preservation briefing paper
[16] von Suchodoletz, D., Rechert, K.: Migrating of complex original environments - verification and quality assurance challenges. JCDL (2013)