Database Archiving in the E-ARK Project

www.eark-project.eu

Dr Janet Delve, University of Portsmouth
Dr Rainer Schmidt, Austrian Institute Technology
Anders Bo Nielsen, Danish National Archives
Boris Domanjko, Slovenian National Archives
Luis Faria, KEEP Solutions
Manfred Thaller, Jan Rörden, University of Köln

PASIG, 17 September 2014
THE E-ARK PROJECT IS CO-FUNDED BY THE EUROPEAN COMMISSION UNDER THE ICT-PSP PROGRAMME

www.eark-project.eu
Outline

• E-ARK Objectives
• Current database archiving practice
  – Danish National Archives
  – Slovenian National Archives
  – RODA
• Transactional (OLTP) vs Analytical (OLAP / Data Warehousing) techniques *(Whistlestop tour!)*
Outline

• Database archiving in E-ARK:
  – Overview of E-Ark Concepts
  – Extract – Transform – Load
  – “Indexing” - Full Text Search
  – Online analytical processing (OLAP)
  – Data Mining
  – Proposed Architecture
  – Tasks and Components
E-ARK Objectives

• Reduce the cost of transfer, preservation and access to digital information by
  – **Standardising** how agencies **export** and send information to digital archives
  – Offering new ways for the long-term **preservation** of various content
  – Exploring needs for **accessing** archives and providing novel interfaces for these

• Long-term vision: Improve semantic and technical interoperability to a level which allows any system developer to deliver out-of-the-box archiving functionality...
E-ARK Objectives

• E-ARK aims to preserve structured and unstructured data:
  – Records
  – Databases

• How?

• Building on current practice (e.g. Denmark, Slovenia, and RODA)

• Using OLAP/ Data Warehousing techniques
Ingest and Preservation

SIP – AIP Conversion

Digital preservation systems

AIP - DIP Conversion

Archival Search, Access and Display Tools

Content and Records Management Systems

Data Mining Showcase

Scalable Computation

Content and Records Management Systems
Archival records submission in Denmark

• Archival records are submitted every 5th year (or sometimes annually)
• The SIP structure is based on all data being in a relational database format
• The SIP is a simple container containing the database in the SIARDDK format along with metadata about the producer and the system creating the records
Extracting the archival records into the SIARDDK format

• The producer extracts the archival records from the database and transforms them into the **SIARDDK** format
  – The archival records consist of the database and the related documents
  – The tools for extracting and transforming are created in closed source by the producer or its supplier and are often inefficient due to the long delay between submissions for each producer
Slovenian Database archiving example
Slovenian Database archiving example

**Objective**

**production:**

<table>
<thead>
<tr>
<th>carSoldID</th>
<th>carForSaleID</th>
<th>salePrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>15000</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>23500</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8600</td>
</tr>
</tbody>
</table>

**archive:**

- end-user access to information in the db
- fulfill the requests for information
Slovenian Database archiving example

**test AIP contents**

1. **Structure (SQL 1999)**
   - `create_script.sql`

2. **Data (CSV files)**
   - `atributi_isvora.csv`
   - `atributi_tabel.csv`
   - `clanstva_zbornic.csv`
   - `clanstva_zbornic_sgod.csv`
   - `clanstva_zdruzenj.csv`
   - `clanstva_zdruzenj_sgod.csv`
   - `dejavnosti_ekd.csv`
   - `dejavnosti_skd.csv`
   - `drzave.csv`
   - `e_svese.csv`
   - `e_svese_posl_subjektor.csv`
   - `ekonomske_skupine.csv`
   - `hisne_stevilke.csv`
   - `izvori_kapitala.csv`
   - `kontinenti.csv`
   - `naselja.csv`
   - `naslovi_s_napakami.csv`
   - `obcine.csv`
   - `oblike_lastnine.csv`
   - `obmoce_zbornice.csv`
   - `organizacijske_oblike.csv`
   - `podrocja_skd.csv`
   - `poste.csv`
   - `ps_vlajnost_isvor.csv`
   - `ps_vlajnost_isvor_sgod.csv`
   - `registrski_organizi.csv`
   - `sobrani_angle.csv`
   - `skripta_archivrs.sql`
   - `spol.csv`
   - `tabele_sa_kontrolo.csv`
   - `tipi_zdruzenj.csv`
   - `ulice.csv`
   - `upravne_enote.csv`
   - `velikosti.csv`
   - `vrste_sprememb.csv`
   - `vrste_tabel.csv`
   - `vrste_trans_racunov.csv`
   - `zbornice.csv`
   - `zdruzenja.csv`

3. **Metadata**
   - `list.txt`
Database Preservation Toolkit

- Developed within RODA project
- Now stand-alone open-source project

- [http://keeps.github.io/db-preservation-toolkit/](http://keeps.github.io/db-preservation-toolkit/)

- Imports and exports between DBMS and DB formats
- Supports preservation formats: DBML, SIARD
db-preservation-toolkit architecture

Import modules:
- MySQL
- Oracle'12
- SQL Server
- PostgreSQL
- DB2
- MS Access
- ODBC
- DBML
- SIARD

Streaming data model

Export modules:
- MySQL
- SQL Server
- PostgreSQL
- DB2
- MS Access
- ODBC
- DBML
- SIARD
- PhpMyAdmin

ICTPSP

E-ARK
Pre-ingest / Production / SIP creation

Origin formats:
- MySQL
- Oracle'12
- SQL Server
- PostgreSQL
- DB2
- MS Access
- ODBC

db-preservation-toolkit

Preservation formats:
- DBML
- SIARD
Access

Preservation formats:
- DBML
- SIARD

Access format / viewer:
- db-preservation-toolkit
- PhpMyAdmin
Transactional Processing (OLTP)
Online Analytical Processing (OLAP)

<table>
<thead>
<tr>
<th>Property Type</th>
<th>City</th>
<th>Time</th>
<th>Total Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
<td>Glasgow</td>
<td>Q1</td>
<td>15056</td>
</tr>
<tr>
<td>House</td>
<td>Glasgow</td>
<td>Q1</td>
<td>14670</td>
</tr>
<tr>
<td>Flat</td>
<td>Glasgow</td>
<td>Q2</td>
<td>14555</td>
</tr>
<tr>
<td>House</td>
<td>Glasgow</td>
<td>Q2</td>
<td>15888</td>
</tr>
<tr>
<td>Flat</td>
<td>Glasgow</td>
<td>Q3</td>
<td>14578</td>
</tr>
<tr>
<td>House</td>
<td>Glasgow</td>
<td>Q3</td>
<td>16004</td>
</tr>
<tr>
<td>Flat</td>
<td>Glasgow</td>
<td>Q4</td>
<td>15890</td>
</tr>
<tr>
<td>House</td>
<td>Glasgow</td>
<td>Q4</td>
<td>15500</td>
</tr>
<tr>
<td>Flat</td>
<td>London</td>
<td>Q1</td>
<td>19678</td>
</tr>
<tr>
<td>House</td>
<td>London</td>
<td>Q1</td>
<td>23877</td>
</tr>
<tr>
<td>Flat</td>
<td>London</td>
<td>Q2</td>
<td>19567</td>
</tr>
<tr>
<td>House</td>
<td>London</td>
<td>Q2</td>
<td>28677</td>
</tr>
<tr>
<td>……</td>
<td>……</td>
<td>……</td>
<td>……</td>
</tr>
<tr>
<td>……</td>
<td>……</td>
<td>……</td>
<td>……</td>
</tr>
</tbody>
</table>

Diagram showing data for different property types in different cities over time.
Typical Architecture for MOLAP Tools

- Relational database server and/or legacy systems
- MOLAP server
- End-user access tools

Load -> Database and application logic layer

Data request -> Result set

Presentation layer
Data warehousing

- *Snapshots of DB* (Useful for DB archiving)
- **Add Updates** from DB
- DW is never changed (only added to)
- Can be denormalised
- Main purpose is analysis
- Star schema – dimensional model
Star Schema

http://www.1keydata.com/datawarehousing/star-schema.html
Very preliminarily ...: E-ARK AIP layer 1

- `<E-ARK-Id><generation-number>/`
  - `data/
  - manifest-???.txt
  - bagit.txt
  - signature.???
  - E-ARK-Id-Record.xml
    - `bitstreams-1/
      - premis.xml
      - database binary
    - `bitstreams-2/
      - premis.xml
      - SIARD representation
  - `metadata/
    - EAD fragment???
    - ...

  ...
Very preliminarily ...: E-ARK AIP layer 2

- `<E-ARK-Id><generation-number>/`
  - `data/`
  - `manifest-???.txt`
  - `bagit.txt`
  - `signature.???
  - `E-ARK-Id-Record.xml`
    - `bitstreams-1/`
      - `premis.xml`
      - `database binary`
    - `bitstreams-2/`
      - `premis.xml`
      - `OLAP representation`
    - `metadata/`
      - `EAD fragment???
      - `...

...
Overview of E-Ark Concepts

• Archiving of databases in different layers:
  – primary format,
  – semantic representation,
  – representation for analytical processing.

• Data intensive technology for AIP storage and processing.
  – Hadoop, HDFS, HBase, Lily, SolR.

• Support for database transformation and analysis
  – such as denormalization, aggregation, indexing.

• Levels of DIP format and display
  – Access to archived records,
  – based on OLAP queries and reports,
  – as dynamically reconstructed RDBs.
Extract – Transform – Load

• Goal: Integrate data from multiple applications into a database/warehouse.
  – Extracting data from source systems like RDBMS and flat files.
  – Transform: derive, extract, aggregate data.
  – Load data into target: Overwrite cumulative information or add new data.

• Important pre-processing step for data mining/analytics.
  – Involves data cleaning and data integration.

• Result: Structured data, random access based on data based indexes (e.g. RDBMS, NOSQL).

• E-Ark Approach: Automated transformation of archived databases into flat, star-schema like structures.
  – Denormalized, connected fact and auto-detected dimension tables.
“Indexing” - Full Text Search

• Searching documents based on full text
  – distinguished from searches based on metadata
  – Returns (ranked) list of document IDs

• Involved Information Retrieval methods
  – Building an inverted index
  – Scoring and weighting Results
  – Text classification
  – Evaluation

• Approach in E-Ark
  – Denormalization / star schema transformation and ingestion into Apache HBase.
  – Auto detection of full text relevant data
  – Repository and faceted search on records based on NGDATA’s Lily repository and Apache SolR.
Online analytical processing (OLAP)

- OLAP Database / Data Warehouse
  - Aggregated, historical data, low transaction rate
  - Resource-intensive and complex queries
- Analyse multi-dimensional data in a read efficient manner (Web analytics, sales)
  - View metrics by combination of dimensions
  - Time vs. Space: Pre-aggregates data to build cube
  - Dynamically analyze data from multiple perspectives
    - roll-up, drill-down, and slicing and dicing
- Approach in E-Ark
  - Data analytics based on pre-processed database representation arranged along dimensions.
  - Data loaded and queried through Apache HBase.
  - Access (DIPs) supported by additional use of OLAP tools.
Proposed Architecture

Data Management Application

ESS Arch Preservation Platform

Scalable Computation Staging Area
Lily, Hadoop, HBase, HDFS

Archive Storage (WORM)

EARK-AIP

Data Connector API

AIP Storage
T6.2, MS10

Data Management Integration
T6.1, MS06, D6.2

CRUD API

Query API

Query and Indexing
T6.3, MS04, D6.1

Data Mining API

Data Mining Showcase
T6.4, D6.3

Re-use and Data Mining
T6.4

Data Management Integration
Tasks and Components

• Archival Storage
  – Store APIs on HDFS using ESS Preservation Platform
  – Bulk-load, permanent and replicated storage

• Data Integration
  – Extract data from archival information package.
  – ETL data into Lily/HBase, keep AIP in HDFS (don’t touch)

• Query and Indexing
  – Metadata on AIP level stored in HBase for basic retrieval
  – Faceted search based on Apache SolR

• Data Mining and Analytics
  – Load OLAP structure from package
  – Data sets stored on record level into HBase
  – Query for facts based on different dimension and levels.