Data-Warehouse-, Data-Mining- und OLAP-Technologien

Data Warehouse Architecture
Overview

- Data Warehouse Architecture
- Data Sources and Data Quality
- Data Mart
- Federated Information Systems
- Operational Data Store
- Metadata
  - Metadata Repository
  - Metadata in Data Warehousing
Architecture

End User Data Access

Data Warehouse Manager

Metadata Manager

Metadata Repository

Data Warehouse

Data Staging Area

Data Staging Area

Data Sources

Load

Transformation

Extraction

Monitor

(A. Bauer, H. Günzel: Data Warehouse Systeme, 2001)
Data Sources

- Characteristics of source systems:
  - narrow, "account-based" queries
  - no queries in a broad and unexpected way, like DW
  - maintain little historical data
  - no conformed dimensions (product, customer, geography, ...) with other legacy systems
  - use keys (production keys) to make certain things unique (product, customer, ...)

- Important issues in selecting data sources:
  - Purpose of the data warehouse
  - Quality of data sources (consistency, correctness, completeness, exactness, reliability, understandability, relevance)
  - Availability of data sources (organizational prerequisites, technical prerequisites)
  - Costs (internal data, external data)
Data Quality

- Are there contradictions in data and/or metadata?
- Do data and metadata provide an exact picture of the reality?
- Are there missing attributes or values?
- Are exact numeric values available? Are different objects identifiable? Homonyms?
- Is there a Standard Operating Procedure (SOP) that describes the provision of source data?
- Does a description for the data and coded values exist?
- Does the data contribute to the purpose of the data warehouse?
Dimensions of Data Sources

- **origin**
  - internal vs. external data

- **time**
  - current vs. historic data

- **usage**
  - data vs. metadata

- **type**
  - number, string, time, graphic, audio, video, ...
  - numeric, alphanumerical, boolean, binary, ...
  - ASCII, EBCDIC, UNICODE, ...

- **character set**
  - left to right, right to left, top-down

- **orientation**
  - strictly confidential, confidential, public, ...

- **confidentiality**
Monitoring

- Goal: Discover changes in data source incrementally
- Approaches:

<table>
<thead>
<tr>
<th></th>
<th>Based on ...</th>
<th>Changes identified by ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>triggers defined in source DBMS</td>
<td>trigger writes a copy of changed data to files</td>
</tr>
<tr>
<td>Replica</td>
<td>replication support of source DBMS</td>
<td>replication provides changed rows in a separate table</td>
</tr>
<tr>
<td>Timestamp</td>
<td>timestamp assigned to each row</td>
<td>use timestamp to identify changes (supported by temporal DBMS)</td>
</tr>
<tr>
<td>Log</td>
<td>log of source DBMS</td>
<td>read log</td>
</tr>
<tr>
<td>Snapshot</td>
<td>periodic snapshot of data source</td>
<td>compare snapshots</td>
</tr>
</tbody>
</table>
Data Staging Area (DSA):

Data Staging Area:
A storage area and a set of processes that clean, transform, combine, deduplicate, household, archive, and prepare source data for use in the data warehouse.

- Data is temporarily stored in the data staging area before it is loaded into the data warehouse.
- All transformations are performed in the DSA.
  - Preprocessing does not influence data sources or data warehouse.
- DSA is the central repository for ETL (Extraction - Transformation - Load) processing.

Extraction

- Transfer data from data source into the data staging area.
- Extracted subset of data sources and schedule of the extraction depends on the kind of analysis that should be supported.
- Method depends on the monitoring strategy used:
  - Read data from a file written by triggers.
  - Read data from replication tables.
  - Select data based on the timestamp.
  - Read data from log.
  - Read output of snapshot comparison.

- Multiple extract types:
  - periodic
  - started by the admin/user
  - event-driven
  - immediate after changes in data sources
Transformation

- Convert the data into something representable to the users and valuable to the business.
  - Transformation of structure and content
- Typical transformations:
  - denormalization, normalization
  - data type conversion
  - calculation, aggregation
  - standardization of strings and date values
  - conversion of measures
  - cleansing (missing, wrong, and inconsistent values)
Load

- Transfer data from the data staging area into the data warehouse.
- Data in the warehouse is rarely replaced. The history of values/changes is stored instead.
- Mainly based on bulk load tools of the DBMS.
- Offline vs. online load.
- Parallel load may be required.
Data Warehouse Manager

- Controls all components of the data warehouse system:
  - **Monitor**: Discover changes in data sources
  - **Extraction**: Select and transfer data from data sources to the data staging area
  - **Transformation**: Consolidate data
  - **Load**: Transfer data from data staging area to the data warehouse
  - **End User Data Access**: Analysis of data in the data warehouse
### Basic Elements of the Data Warehouse

<table>
<thead>
<tr>
<th>Source Systems</th>
<th>Data Staging Area</th>
<th>&quot;The Data Warehouse&quot; Presentation Servers</th>
<th>End User Data Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage:</td>
<td></td>
<td><strong>Data Mart #1:</strong> OLAP query services;</td>
<td>Ad Hoc Query Tools</td>
</tr>
<tr>
<td>Flat files; RDBMS; other</td>
<td></td>
<td>dimensional! subject oriented; locally implemented; user group driven; may store atomic data; may be frequently refreshed; conforms to DW Bus</td>
<td></td>
</tr>
<tr>
<td>Processing:</td>
<td></td>
<td><strong>Data Mart #2:</strong> Conformed dimensions;</td>
<td>Report Writers</td>
</tr>
<tr>
<td>clean; prune; combine; remove duplicates; household; standardize; conform dimensions; store awaiting replication; archive; export to data marts</td>
<td></td>
<td>Conformed facts</td>
<td></td>
</tr>
<tr>
<td>No user query services</td>
<td></td>
<td><strong>Data Mart #3:</strong> Conformed dimensions;</td>
<td>End User Applications</td>
</tr>
<tr>
<td>upload cleaned dimensions</td>
<td></td>
<td>Conformed facts</td>
<td>Models</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>forecasting; scoring; allocating; data mining; other downstream systems; other parameters; special UI</td>
</tr>
</tbody>
</table>

Anwendersoftware

Data Warehouse Architecture

Architecture

Central architecture
- only one data model
- performance bottleneck
- complex to build
- easy to maintain

Federal architecture
- logically consolidated
- separate physical databases that store detailed data
- faster response time

Tiered architecture
- physical central data warehouse
- separate physical databases that store summarized data
- faster response time

(M. Jarke et al., Fundamentals of Data Warehouses, 2002)
Data Marts

 dependent data marts

 independent data marts
Data Marts

### dependent data marts (tiered architecture)
- Central data warehouse (DW) is build first
- Extracts of the data warehouse are provided as data marts (materialized views)
- Establish ETL process for DW only
- Consistent analysis on DW and DM

### independent data marts (federated architecture)
- Several data marts (DM) are build first
- Data marts are integrated by means of a second transformation step
- Establish ETL process for each DM and the central DW
- Inconsistent analysis is possible
- Virtual data warehouse possible (federated architecture)
Federated Information Systems

- Federated DBMS
  - Transparent access to a collection of heterogeneous and semi-autonomous data sources.

- Complete, extensible database engine
  - Function compensation
  - Powerful (global) query optimizer (pushdown analysis, cost-based optimization, query rewrite)

Diagram:
- Presentation layer
- Federation layer
  e.g. uniform access language, uniform access schema, uniform metadata set
- Foundation layer
  (data sources)
- Local Applications

[Diagram depicting layers and flows]
Architecture for a Federated Database Server

Client \rightarrow SQL API \rightarrow Federated Database Server \rightarrow Wrapper

\rightarrow Back-end Data Source \rightarrow Data

\rightarrow Back-end Data Source \rightarrow Data

Catalog \rightarrow Data
Federated DBMS: Processing Scenario

- Knowing what the data source can do is a good idea!
Operational Data Store (ODS)

- Term has taken many definitions. For example:
  - **Point of integration for operational systems**
    - refreshed within a few seconds after the operational data sources are updated
    - very little transformations are performed
    - Example: Banking environment where data sources keep individual accounts of a large multinational customer, and the ODS stores the total balance for this customer.
    
    true operational system separated from the data warehouse
  
  - **Decision support access to operational data**
    - integrated and transformed data are first accumulated and then periodically forwarded to the ODS
    - involves more integration and transformation processing
    - Example: Bank that stores in the ODS an integrated individual bank account on a weekly basis
    
    part of the data warehouse or separate system?
Classes of Operational Data Stores

- Tables are copied from the operational environment
- Transactions are moved to the ODS in an immediate manner (range of one to two seconds)
- Activities in the operational environment are stored, integrated, and forwarded to the ODS
- ODS is fed aggregated analytical data from the data warehouse
- Combination of integrated data from the operational environment and aggregated data from the analytical environment

Distribution of DW Project Costs

DW Design Costs
- disk storage: 30%
- processor costs: 20%
- integration and transformation: 15%
- network costs: 10%
- DBMS: 10%
- access/analysis tools: 6%
- metadata design: 5%
- activity monitor: 2%
- data monitor: 2%

Recurring DW costs
- DW refreshment: 55%
- servicing data mart requests for data: 21%
- monitoring of activity and data: 7%
- end-user training: 3%
- metadata management: 2%
- periodic verification of the conformance to the enterprise data model: 2%
- data archiving: 1%
- reorganization of data: 1%
- capacity planning: 1%
- summary table usage analysis: 2%
- security administration: 1%

(M. Jarke et al., Fundamentals of Data Warehouses, 2002)
A repository is a shared database of information about engineering artifacts, such as software, documents, maps, information systems, and manufactured components and systems.

Functions of a repository:
- Object management
- Dynamic extensibility
- Relationship management
- Notification
- Version management
- Configuration management

(P. Bernstein: Repositories and Object Oriented Databases, 1998)
Metadata in Data Warehousing

- What data is available in the warehouse and where is the data located?
- **Data dictionary**: Definitions of the databases and relationship between data elements
- **Data flow**: Direction and frequency of data feed
- **Data transformation**: Transformations required when data is moved
- **Version control**: Changes to metadata are stored
- **Data usage statistics**: A profile of data in the warehouse
- **Alias information**: Alias names for a field
- **Security**: Who is allowed to access the data

Stored in a metadata repository

Need for a standard interchange format
Metadata in Data Warehousing

- Criteria to identify important classes of metadata in data warehousing:
  - Type of data
  - Abstraction
  - User
  - Origins
  - Time

- Usage of metadata in data warehousing:
  - passive
  - active
  - semi-active

- Main goals:
  - Support development and operation of a data warehouse
    - system integration
    - processes for DW administration
    - flexible application development
    - access rights
  - Provide information for data warehouse users
    - quality of data
    - consistent terminology
    - support for data analysis
Metadata Management

centralized - decentralized - federated

(A. Bauer, H. Günzel: Data Warehouse Systeme, 2001)
Summary

• Basic Components:
  ▪ Data Staging Area: Extraction, Transformation, Load
  ▪ Data Warehouse Database
  ▪ Data Warehouse Manager
  ▪ Metadata Repositories and Metadata Manager

• Data Marts: Distributed Data Warehouse

• Data Warehouse vs. Federated Information Systems

• Metadata is important to:
  ▪ Support development and operation of a data warehouse
  ▪ Provide information for data warehouse users

• Metadata standards are important to interchange metadata between warehouse tools, warehouse platforms and warehouse metadata repositories.