Data Warehousing
Metadata Management

Spring Term 2018
Dr. Andreas Geppert
geppert@acm.org
Outline of the Course

- Introduction
- DWH Architecture
- DWH-Design and multi-dimensional data models
- Extract, Transform, Load (ETL)
- Metadata
- Data Quality
- Analytic Applications and Business Intelligence
- Implementation and Performance
Content

1. Motivation
2. Metadata
3. Metadata Creation and Usage
4. Metadata Management System Architecture
5. Appendix: DWH Metadata Standards
Motivation

Metadata management is the prerequisite for two major objectives:

1. Enable and minimize effort for the development and operations of a data warehouse

2. Enable optimal and correct information extraction
   - data analysis: requires known and agreed semantics of data
   - uniform terminology
   - data quality; quality definition, quality checks, correction rules, quality requirements
Metadata

- "data about data": data that describe other data
- Meaning/semantics, syntax, etc.
- "any kind of information that is needed for the design, development and use of an information system" (Bauer & Günzel)
- metadata management: gathering, storage, administration, and providing of metadata
- storage in an own information system -> Repository
Instance and Metadata

- Meta meta models
- Meta models
  - define metadata schema
  - concept "table", "attribute"
- Metadata (data / object model)
  - represent schemas, types
  - Type/class "Customer", attribute "Main Street"
- Instance data, object data
  - represent elements of the UoD and their properties
  - Customer "John Doe", address "Main Street"
Metadata: Examples in the DWH-Architecture

- Source data models, Ownership
- Staging structures
- Transformations
- Data models, quality rules
- Transformations
- Calculations
- Data models
- Selection, Aggregation, Calculation
- Analysis logic, calculations, terminology
- Reports
- Web/App Servers
- GUI
- Metadata Management

- Subject Matter Areas
- Reusable Measures & Dimensions Areas
- Metadata Management
- Source data models, Ownership
- Staging structures
- Transformations
- Data models, quality rules
- Transformations
- Calculations
- Data models
- Selection, Aggregation, Calculation
- Analysis logic, calculations, terminology
- Reports
- Web/App Servers
- GUI
Requirements to Metadata Management

### Availability and existence
- Metadata must exist (e.g., data models of sources and the DWH)

### Automated capture
- Metadata must (whenever possible) be captured and updated in the course of DWH development
- Documentation after the fact ("Nachdokumentation") – development plus subsequent documentation in the metadata repository does not work, metadata and code become inconsistent very fast

### Integration
- Metadata must be integrated and provide an end-to-end view
- Relationships between metadata (of different types) must be captured, managed, and used
  - For instance, transformation rules are not independent metadata, they describe the mapping of a schema onto another one
- Without seamlessly integrated metadata it is impossible to support impact analysis and data lineage
Requirements to Metadata Management (2)

**User access**
- metadata support for all users and user groups
- adequate user interfaces, languages, etc.

**Tool support**

**Interoperability**
- metadata exchange between different systems should be possible
Metadata: Classification: Creation and usage time

- design metadata
  - schema definitions/data models, transformation rules, data quality rules, terms and definitions
- runtime metadata
  - log files, statistics, quality check results
- usage metadata
  - usage frequencies, access patterns
Metadata: Classification (2)

**User**
- technical metadata
  - used by developers, administrators (IT staff in general)
  - data dictionaries, database schemas, transformation rule code
- business metadata
  - definitions of terms, calculation formulas

**Type**
- metadata about primary data
  - metadata about data in source systems, data warehouse, data marts
- process metadata
  - rules and transformations of ETL-processes
  - logs, execution plans
Metadata: Classification (3)

**Origin**
- tools
  - schema designer, ETL-tool, ...
- sources
- users

**Abstraction**
- conceptual
  - abstract description,
    implementation independent
  - understandable for users
- logical
  - description in formal language
  - e.g., database schema, formulas
- physical
  - implementation
  - e.g. SQL-code
Content

1. Motivation
2. Metadata
3. Metadata Creation and Usage
4. Metadata Management System Architecture
5. Appendix: DWH Metadata Standards
Metadata Usage

- passive use: documentation of various aspects of a DWH. Used by users, developers, administrators
- active use: interpretation of metadata through tools (transformation rules, quality rules)
  - metadata-driven processes
- semi-active use: use of metadata through tools in order to check something (e.g., schema definitions)
Metadata Use: Build Time

Analysis application development process:
- Analyze info req
- Derive data req
- Determine reuse potential
- Determine technology
- Model data mart
- Model mappings

Integration application development process:
- Analyze data req
- Analyze source
- Def. incom. interf.
- Extend staging area
- Model SMA
- Model mappings
- Define quality checks
- Design SMA
- Design ETL
- Implement SMA
- Implement ETL
- Def. outgoing interfaces
- Implement interfaces
- Design logical schema
- Design mappings
- Implement data mart
- Implement ETL process
- Develop reports

© Andreas Geppert
Spring Term 2018  Slide 15
Metadata Use: Build Time (2)

- analysis of required data: requirements
- source analysis: requirements, source metadata (data models)
- definition of "incoming" interfaces: interface contracts
- extension of staging area: logical data model
- modeling of SMA-structures: conceptual data model
- modeling of mappings: conceptual ETL-model
- definition of quality rules: quality rules
- logical modeling of SMA-structures, SMA- and ETL-implementation: logical and physical data and ETL-models, job nets
- definition of "outgoing" interfaces: interface contracts
Metadata Use: Build Time (3)

- Analysis of information requirements: requirements business metadata/glossaries, ownership
- Derivation of required data: requirements, analysis model
- Identification of reuse potential of dimensions and measures: application portfolio business intelligence (application) strategy
- Data mart modeling: conceptual data model, roles, access rules, data quality requirements and rules
- Modeling of mappings: conceptual ETL-model
- Logical modeling and implementation of data mart and transformations: logical and physical data and process models, job nets
- Report development: semantic layer, business metadata/glossaries
End-to-end Metadata Use: Build Time

- Impact analysis determines the impact of changes in one place to downstream components.
- It thus requires an end-to-end view of metadata.
- In particular, data models and transformations must be complete.
- Development techniques such as generation or derivation of lower-level artifacts out of higher-level (more abstract) ones foster impact analysis.
- Development techniques that do not have an integral treatment of metadata compromise impact analysis (in particular, stored procedures).
End-to-end Metadata Use: Impact Analysis

Layered Architecture

Data Sources

Reference/Master Data

Staging Area

Domain Integration and Enrichment

Integration, Aggregation, Calculation

Federated Integration

Selection, Aggregation, Calculation

Data Marts

Reporting and Analysis Services

Reporting, OLAP, Data Mining

Front End

GUI

Legend:

- relational database
- multidimensional database
- file
- logic; extract, transform, load
- logic (no ETL)
- data flow
Metadata Use: Runtime

- metadata use depends on the kind of user access
- ad-hoc querying: user must first formulate query → use metadata to determine the data to analyze and query (data models, semantic layer)
- possibly use metadata search engine to determine data to query
- reporting: users consume generated reports → use metadata about reports (report metadata, semantic layer)
- OLAP: querying and analyzing multidimensional data structures → data models, KPI definitions
Metadata Use: Runtime

- report on various aspects of the DWH (metadata reporting)
- DWH content: DWH data models, tables, table rows, transformations, reports, ...
- DWH processing: report on DWH load processes (number of load jobs, number of feeder files processed, number of rows loaded, etc.)
- DWH usage: number of reports consumed, (number of) queries, n most complex queries, etc.
- DWH security: number of users, roles, privileges, number of logins, number of failed login attempts, ...
- data quality reports: number of DQ rules, checks, and DQ-issues
- DWH performance: database size, response times, storage and server utilization, ...
Content

1. Motivation
2. Metadata
3. Metadata Creation and Usage
4. Metadata Management System Architecture
5. Appendix: DWH Metadata Standards
Metadata Management Architecture: Centralized

- single, centralized metadata management (system)
- usually only feasible if all components are from the same vendor (vs. best of breed)
Metadata Management Architecture: decentralized

- multiple metadata management systems (e.g. tool-specific)
- bilateral metadata exchange (max. $n^2 - n/2$ interfaces)
Metadata Management Architecture: DWH-like

- "DWH-approach": autonomous metadata management systems and repositories are responsible for local tasks
- metadata are integrated in "global" metadata DWH, which is responsible for all tasks requiring integrated metadata

Diagram:

![Metadata Management Architecture Diagram](image-url)
Metadata Management Architecture: DWH-like

- **Metadata Integration, Historization**
- **Reusable Selection, Aggregation, Calculation**
- **Selection, Aggregation, Calculation**
- **Metadata Reporting, OLAP, Data Mining**
- **Web/App Servers**
- **GUI**

**Data Sources**

- **Landing Zone**
- **Staging Area**
- **Metadata Areas**
- **Design, Modeling, and ETL Tools, DBS**
- **ETL**
- **Data Quality Business Terms**

#data models; data model mismatches; stability

- **#data models**
- **data model mismatches**
- **stability**

**Metadata reports and analysis**
Example Source Metadata: Schema of a DB-Repository

- database catalog contains database metadata
- schema of the catalog defines metadata structure (meta meta model)
- DB2: schema syscat, contains ~ 70 catalog views
- Oracle: schema user sys, owns > 3000 views
Metadata about Metadata: Schema of a Repository

- s. Marco 2000
Summary

- Metadata are created and used in all phases and layers of data warehousing
- Complete and adequate metadata management is a success factor
- Integrated metadata management can be a challenge in a heterogeneous environment (i.e., best-of-breed)
Content

1. Motivation
2. Metadata
3. Metadata Creation and Usage
4. Metadata Management System Architecture
5. Appendix: DWH Metadata Standards
Metadatenstandards

- **Open Information Model (OIM)**
  - Version 1.0 definiert in 1999 durch Metadata Coalition (MDC)

- **Common Warehouse Model (CWM)**
  - erste Version definiert in 1999 durch Object Management Group (OMG)
  - einfacher Austausch von DWH-Metadaten zwischen Werkzeugen und Repository
  - Modularität, so dass auch nur relevante Teile des Models implementiert werden können
**CWM: Struktur**

- CWM erlaubt Repräsentation von Metadaten über ...  
- Quellen, Targets, und Transformationen  
- Analysen  
- Prozesse und Operationen, die Warehouse-Daten erzeugen und verwalten sowie Lineage der Verwendung erlauben

- CWM basiert auf UML  
- weitgehende Wiederverwendung des Object Models (Teil von UML)  
- CWM verwendet UML-Packages und eine hierarchische Package-Struktur aus Gründen der Komplexität, Verständnis und Wiederverwendbarkeit
CWM: Struktur

<table>
<thead>
<tr>
<th>Management</th>
<th>Warehouse Process</th>
<th>Warehouse Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transformation</td>
<td>OLAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Mining</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information Visualization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Business Nomenclature</td>
</tr>
<tr>
<td>Resource</td>
<td>Object Model</td>
<td>Relational</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multidimensional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XML</td>
</tr>
<tr>
<td>Foundation</td>
<td>Business Information</td>
<td>Data Types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keys and Indexes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type Mapping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Deployment</td>
</tr>
</tbody>
</table>

- jedes Paket kann Pakete der gleichen Schicht oder der unteren Schichten referenzieren
CWM: Layer "Foundation"

- "Foundation" bietet CWM-spezifische Dienste für andere Packages auf höheren Schichten
- Data Types: Klassen und Assoziationen für die Definition von Datentypen
- Expressions: K&A für die Repräsentation von Ausdrucksbäumen
- Keys and Indexes: K&A, die Schlüssel und Indexe repräsentieren
- Software Deployment: K&A, mit denen repräsentiert werden kann, wie Software in einem DWH "deployed" wird
- Type Mapping: K&A für die Abbildung von Datentypen zwischen verschiedenen Systemen
CWM: Layer "Ressourcen"

- Relational: Metadaten relationaler Systeme
- Record: Metadaten satzorientierter Systeme
- Multidimensional: Metadaten multidimensionaler Systeme
- XML: Metadaten von XML-Daten
CWM: Layer "Analysis"

- Transformation: Metadaten über Transformationen (aus Transformationswerkzeugen)
- OLAP: Metadaten aus OLAP-Werkzeugen
- Data Mining: Metadaten aus Data Mining-Werkzeugen
- Information Visualization: Metadaten aus Werkzeugen für die Informationsvisualisierung
- Business Nomenclature: Metadaten über Business-Taxonomien und -Glossare
CWM: Layer "Management"

- Warehouse Process: Metadaten über DWH-Prozesse
- Warehouse Operation: Metadaten über DWH-Betrieb (Ergebnisse)
CWM: "Multidimensional" Package

- generische Repräsentation einer multidimensionalen Datenbank
CWM: "OLAP" Package

- generische Repräsentation von OLAP-Konzepten