Data Warehousing
Metadatenverwaltung

Frühlingssemester 2009
Dr. Andreas Geppert
Credit Suisse
geppert@acm.org

CREDIT SUISSE

Gliederung der Vorlesung

- Einführung
- DWH-Architektur
- Multidimensionale Datenmodelle
- DWH-Entwurf
- Extraktion, Transformation und Laden (ETL)
- ⇨ **Metadaten**
- Datenqualität
- Analytische Anwendungen
- Implementierungs- und Performance-Aspekte

Inhalt

1. Motivation
2. Metadatenverwaltung & DWH
3. DWH-Metadatenstandards

Motivation

Metadaten-Management ist Voraussetzung für zwei Hauptziele:

1. Aufwand für Entwicklung und Betrieb eines DWHs zu minimieren (bzw. überhaupt zu ermöglichen)
   - Integration: Information über Bedeutung der Quellen und des DWH, Werkzeugintegration
   - Prozessautomatisierung: Scheduling, Konfiguration, Protokolle
   - Flexibler (Software-)Entwurf: Anpassbarkeit und Wiederverwendung von Transformationsregeln
   - Schutz- und Sicherheitsaspekte: einheitliche Zugriffsrechte

Motivation

2. Optimalen bzw. korrekten Informationsgewinn ermöglichen
   - Datenanalyse: beruht auf bekannter und einheitlicher Semantik von Daten, Kennzahlensystemen
   - einheitliche Terminologie
   - Datenqualität: Qualitätsdefinitionen, Ueberprüfungsregeln, Reparaturregeln, Qualitätsansprüche etc.

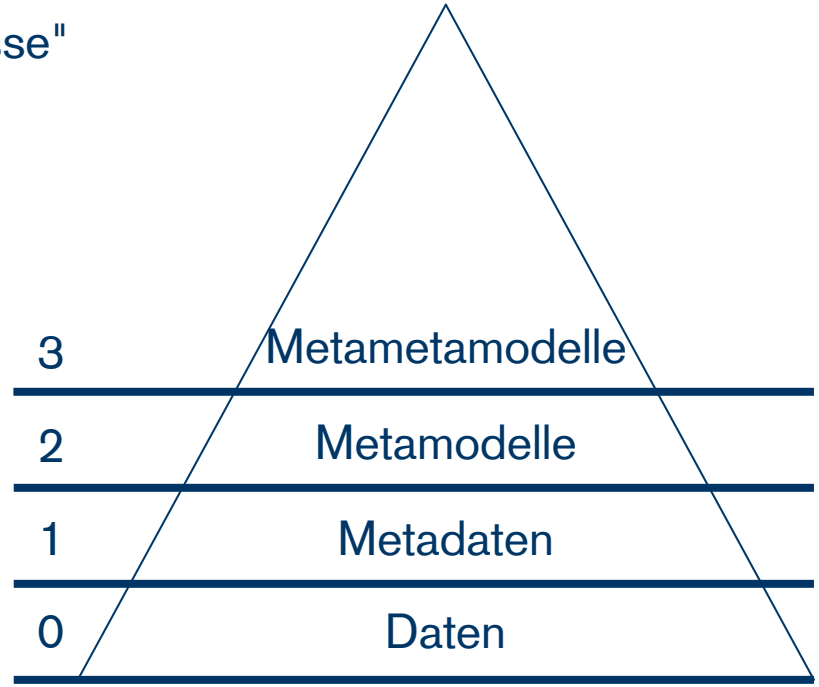
Metadaten

- "Daten über Daten": Daten, die andere Daten beschreiben
- Bedeutung, Syntax, etc.
- "jede Art von Information, die für den Entwurf, die Konstruktion und die Benutzung eines Informationssystemes benötigt wird" (Bauer & Günzel)
- Speicherung und Verwaltung der Metadaten in einem eigenen Informationssystem

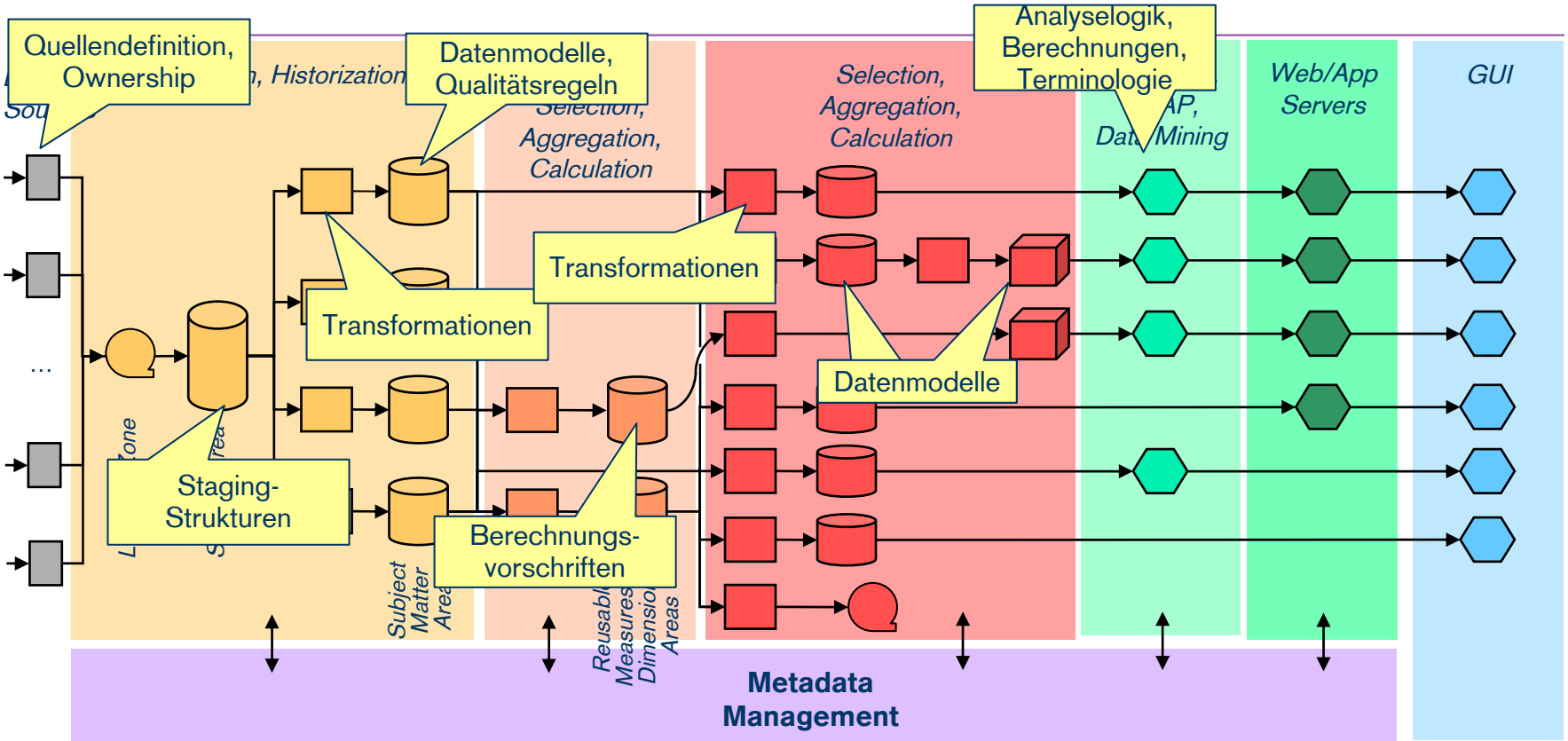
  ⇨ Repositorium (Repository)

Instanz- und Metadaten

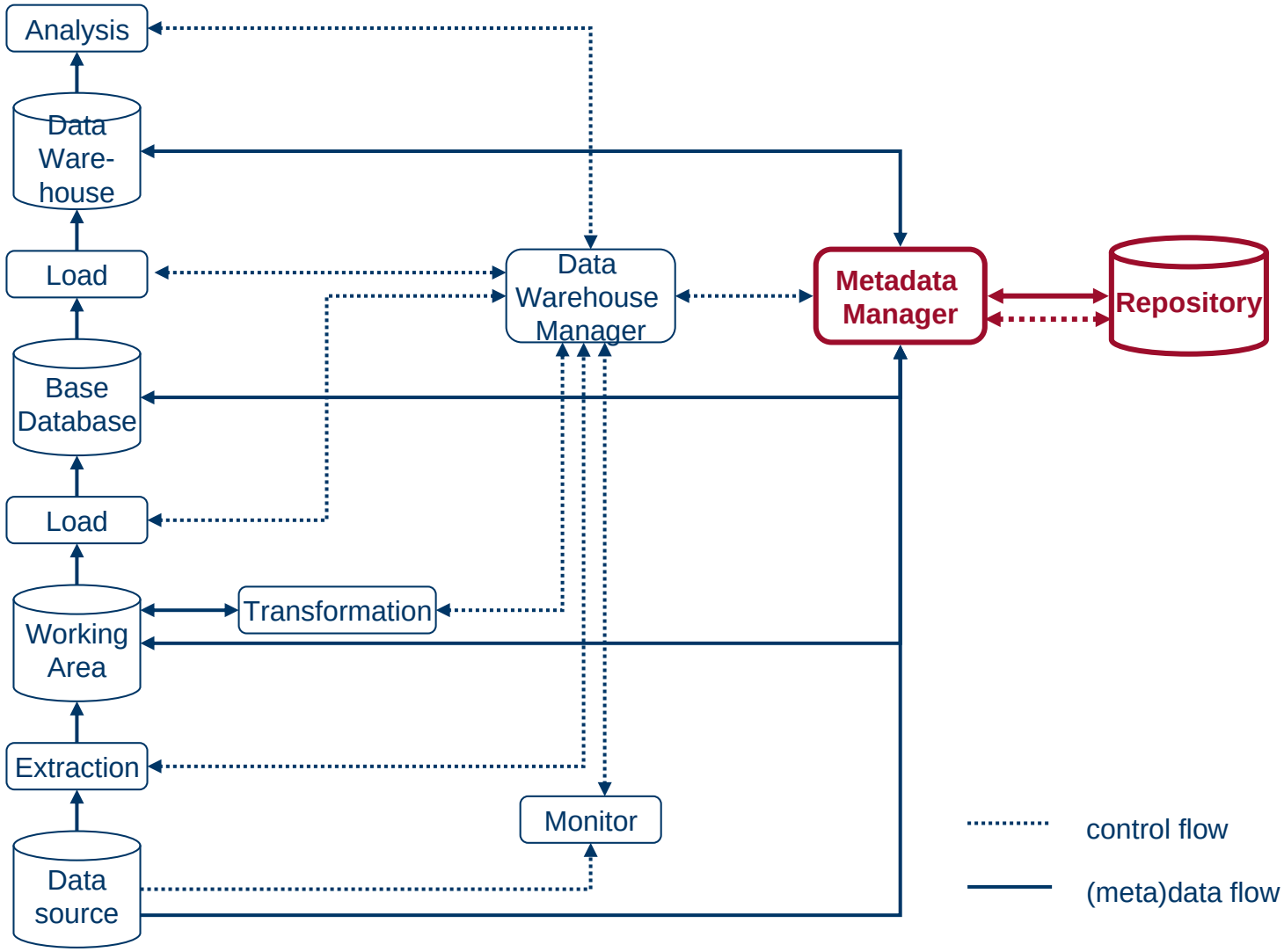
- Instanzdaten, Objektdaten
  - repräsentieren Gegenstände des UoD
  - Bsp. Kunde "Müller", Adresse "Bachgasse"
- Metadaten
  - repräsentieren Schema bzw. Typen
  - Bsp. Typ/Klasse oder Tabelle "Kunde", Attribut "Strasse"
- Metamodelle
  - repräsentieren Schema der Metadaten
  - Bsp. "Tabelle", "Attribut"
- Metametamodelle



Metadatenverwaltung: Fundament der DWH-Architektur



Metadaten in der Referenzarchitektur nach [Bauer & Günzel]



Anforderungen an die Metadatenverwaltung

- Automatisierung
  - Metadaten müssen (wann immer möglich) automatisch im Zuge der DWH-Entwicklung erfasst und nachgeführt werden
  - "Nachdokumentation" (Entwicklung mit anschliessender Dokumentation in der Metadatenverwaltung) funktioniert nicht; Metadaten und Code werden sehr schnell inkonsistent
- Integration
  - Metadaten müssen integriert werden und eine Gesamtsicht (end-to-end) bieten
  - Beziehungen zwischen Metadaten (unterschiedlicher Arten) müssen erfasst, verwaltet und benutzt werden
  - z.B. sind Transformationen nicht alleinstehende Metadaten, sondern beschreiben die Abbildung eines Schemas in ein anderes Schema
  - ohne (durchgängig!) integrierte Metadaten sind Data Lineage und Impact Analyse nicht möglich

Anforderungen an die Metadatenverwaltung (2)

- Anwenderzugriff
  - Bereitstellung der benötigten Informationen für alle Anwender(gruppen)
  - anwendergerechte Schnittstellen
  - adäquate Sprachen und Konstrukte (möglichst formal und für Benutzer verständlich)
- Werkzeugunterstützung
  - DWH-Werkzeuge müssen angebunden werden können (Analyse, etc.)
- Interoperabilität
  - Metadaten müssen mit anderen Systemen ausgetauscht werden können

Metadatenverwaltung: Klassifikation (1)

- Kriterium "Erstellungs- und Verwendungszeitpunkt"
- Entwurfsmetadaten
  - Schemadefinitionen, Transformationsregeln, Datenqualitätsregeln, Begriffsdefinitionen
- Aufbaumetadaten
  - Protokolldateien, Statistiken, Qualitätsprüfungsresultate
- Benutzungsmetadaten
  - Verwendungshäufigkeit, Zugriffsmuster

Metadatenverwaltung: Klassifikation (2)

- Kriterium "Anwendersicht"
- technische Metadaten
  - verwendet von Entwicklern und Administratoren
  - Data Dictionaries, Schemadefinitionen, Code von Transformationsregeln
- Business-Metadaten
  - Begriffsdefinitionen, Berechnungsvorschriften

Metadatenverwaltung: Klassifikation (3)

- Kriterium "Typ"
- Metadaten über Primärdaten
  - Metadaten über Datenbestände der Quellsysteme, des DWHs, der Data Marts
- Prozessmetadaten
  - Regeln und Transformationen der ETL-Prozesse
  - Protokolle, Ausführungspläne

Metadatenverwaltung: Klassifikation (4)

- Kriterium "Herkunft"
- Werkzeuge
  - Schema-Designer, ETL-Werkzeug
- Quellen
- Anwender

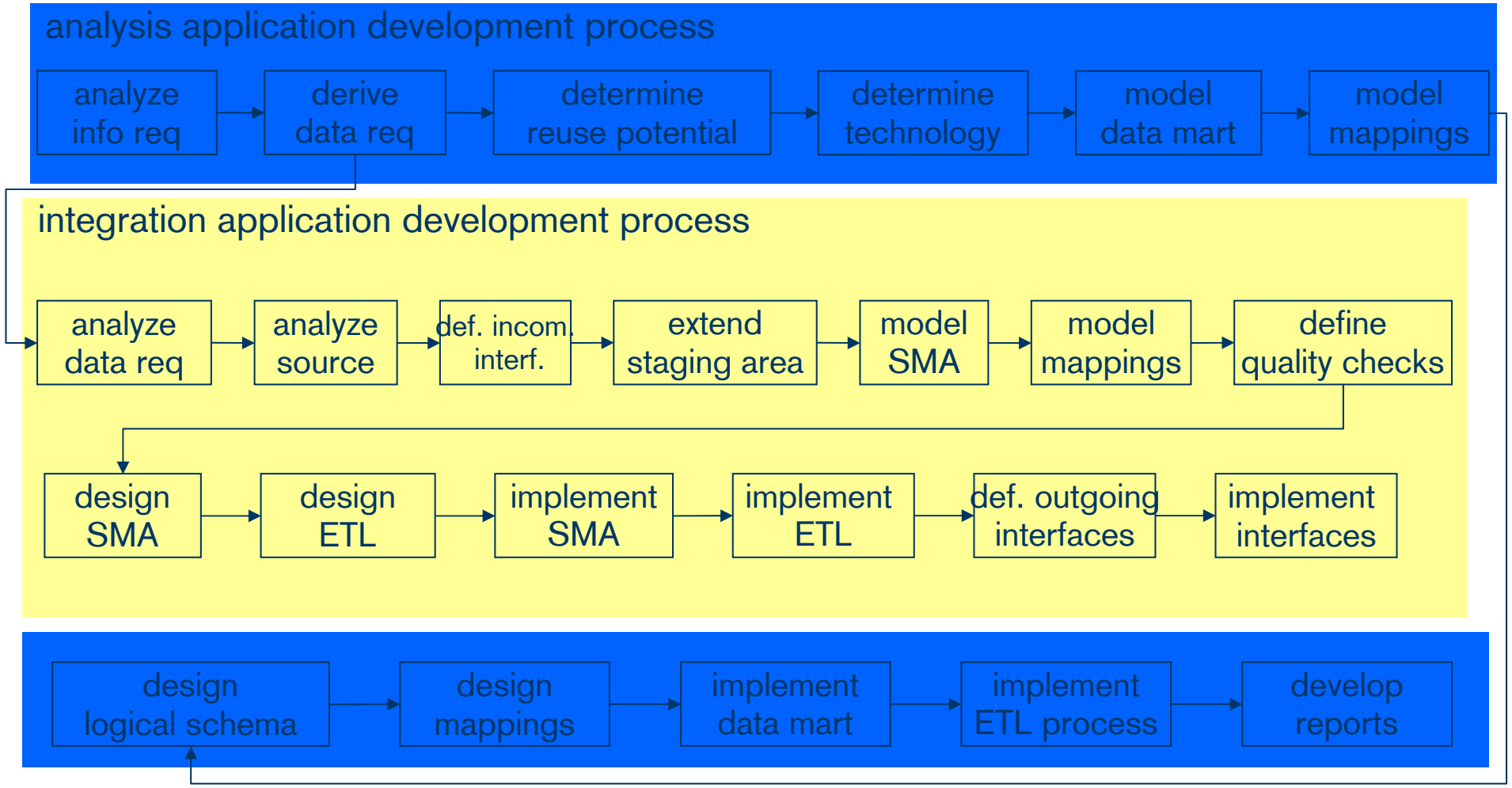
Metadatenverwaltung: Klassifikation (5)

- Kriterium "Abstraktion"
- konzeptuell
  - abstrakte Beschreibung, implementierungsunabhängig
  - teilweise auch natürlichsprachlich
  - für Anwender verständlich
- logisch
  - Beschreibung in formaler Sprache
  - z.B. Datenbankschema, Formeln
- physisch
  - Implementierung
  - z.B. SQL-Code

Nutzung von Metadaten

- passiv: Dokumentation der verschiedenen Aspekte eines DWHs. Nutzung durch Anwender, Entwickler, Administratoren
- aktiv: Interpretation von Metadaten durch Werkzeuge (z.B. Transformationsregeln, Qualitätsregeln)
  - metadatengetriebene Prozesse
- semi-aktiv: Verwendung von Metadaten durch Werkzeuge zur Ueberprüfung von Sachverhalten (z.B. Schemadefinitionen)

Nutzung von Metadaten: Entwicklungszeit



Erzeugung von Metadaten: Entwicklungszeit (2)

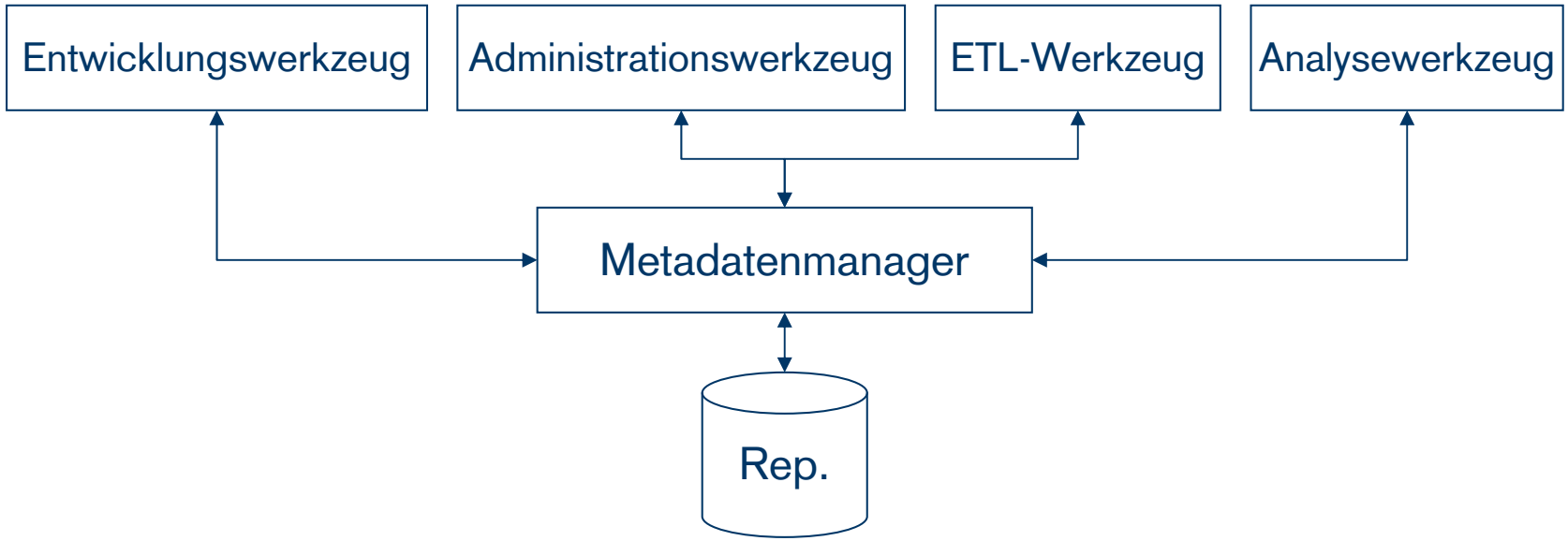
- Analyse der benötigten Daten: Anforderungen
- Quellanalyse: Anforderungen, Quellmetadaten (Datenmodelle)
- Definition der "eingehenden" Schnittstellen: Schnittstellenkontrakte
- Erweiterung der Staging Area: logisches Datenmodell
- Modellierung der SMA-Strukturen: konzeptuelles Datenmodell
- Modellierung der Mappings: konzeptuelles ETL-Modell
- Definition von Qualitästregeln: Qualitätsregeln
- logische Modellierung und Implementierung der SMA und der Transformationen: logische und physische Daten- und Prozessmodelle, Job-Netze
- Definition von "ausgehenden" Schnittstellen: Schnittstellenkontrakte

Erzeugung von Metadaten: Entwicklungszeit (3)

- Analyse der Informationsbedürfnisse: Anforderungen, Business Metadata/Glossare, Ownership
- Ableitung der benötigten Daten: Anforderungen, Analysemodell
- Bewerte Wiederverwendungspotential der Dimensionen und Masse: Anwendungsportfolio, BI-Strategie
- Modellierung des Data Mart: konzeptuelles Datenmodell, Rollen, Zugriffsregeln, Datenqualitätsanforderungen- und regeln
- Modellierung der Mappings: konzeptuelles ETL-Modell
- logische Modellierung und Implementierung des Data Marts und der Transformationen: logische und physische Daten- und Prozessmodelle, Job-Netze
- Report-Entwicklung: Semantic Layer

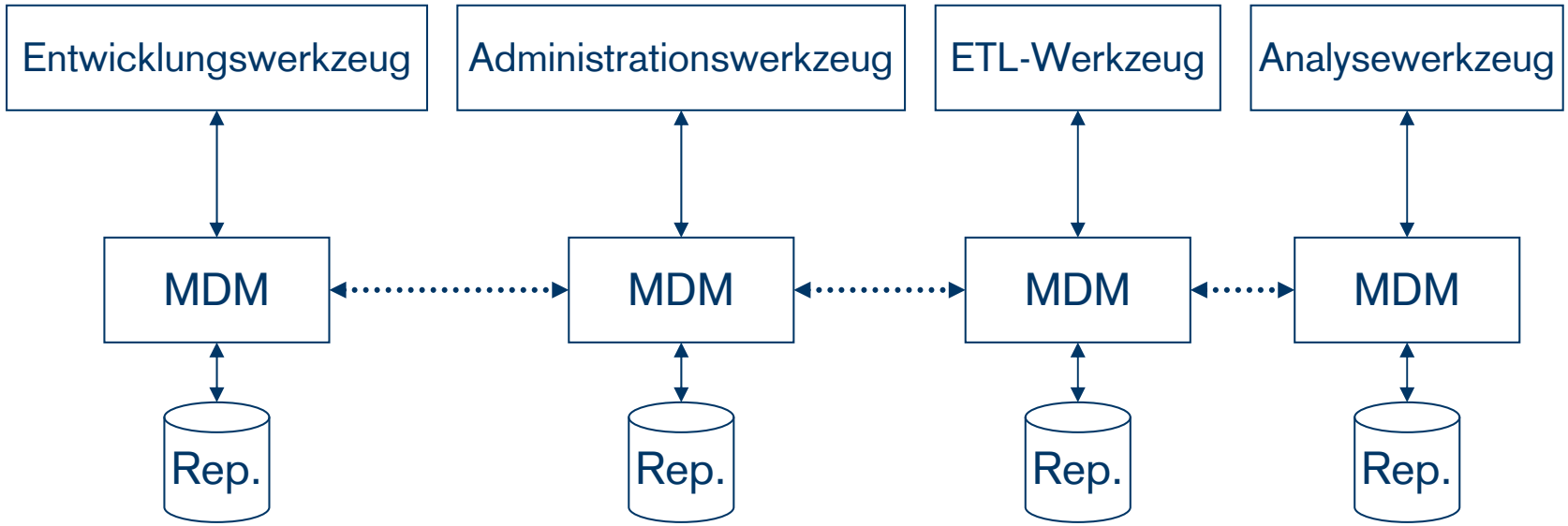
Metadatenverwaltung: zentralisierte Architektur

- eine einheitliche Metadatenverwaltung
- i.d.R. nur möglich, wenn alle Komponenten vom gleichen Hersteller (vs. *best of breed*)



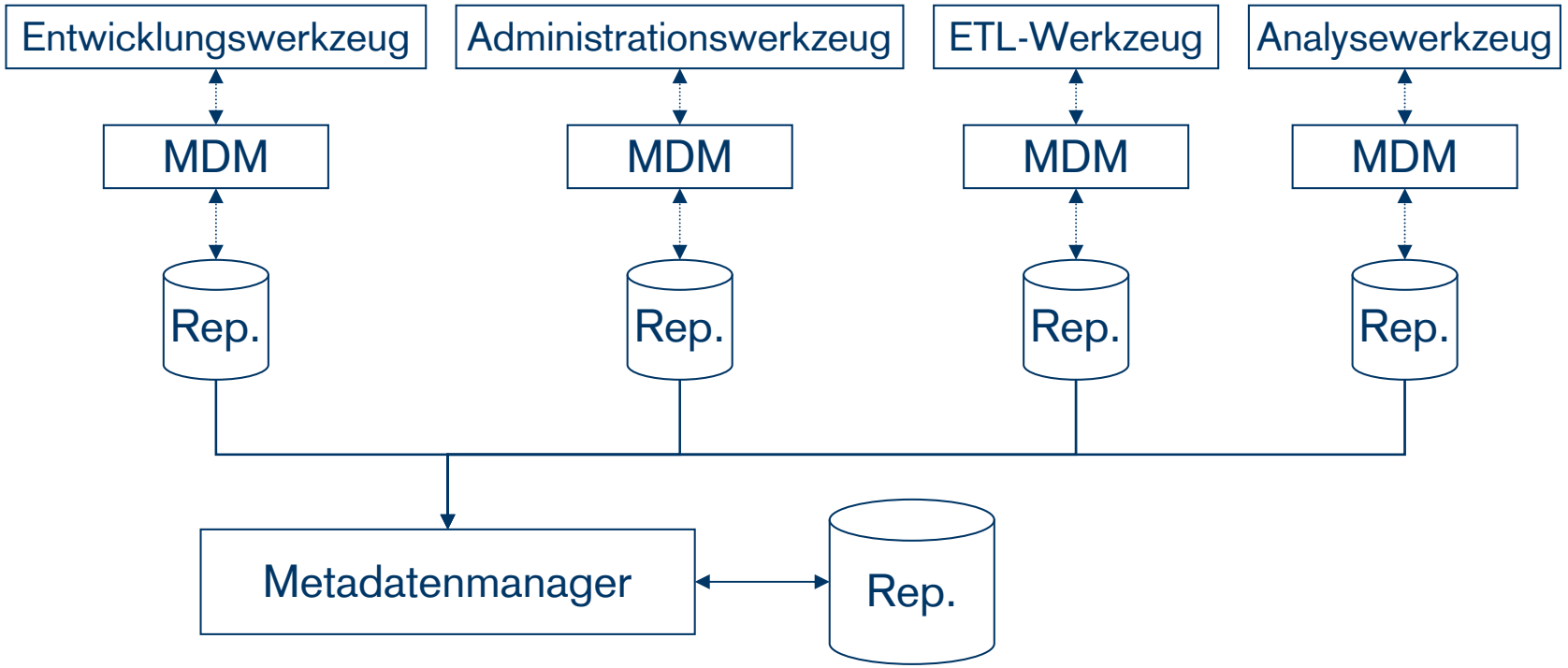
Metadatenverwaltung: dezentralisierte Architektur

- mehrere Metadatenmanager (z.B. werkzeugspezifisch)
- Austausch von Metadaten bilateral (max. n*(n-1)/2 Schnittstellen)

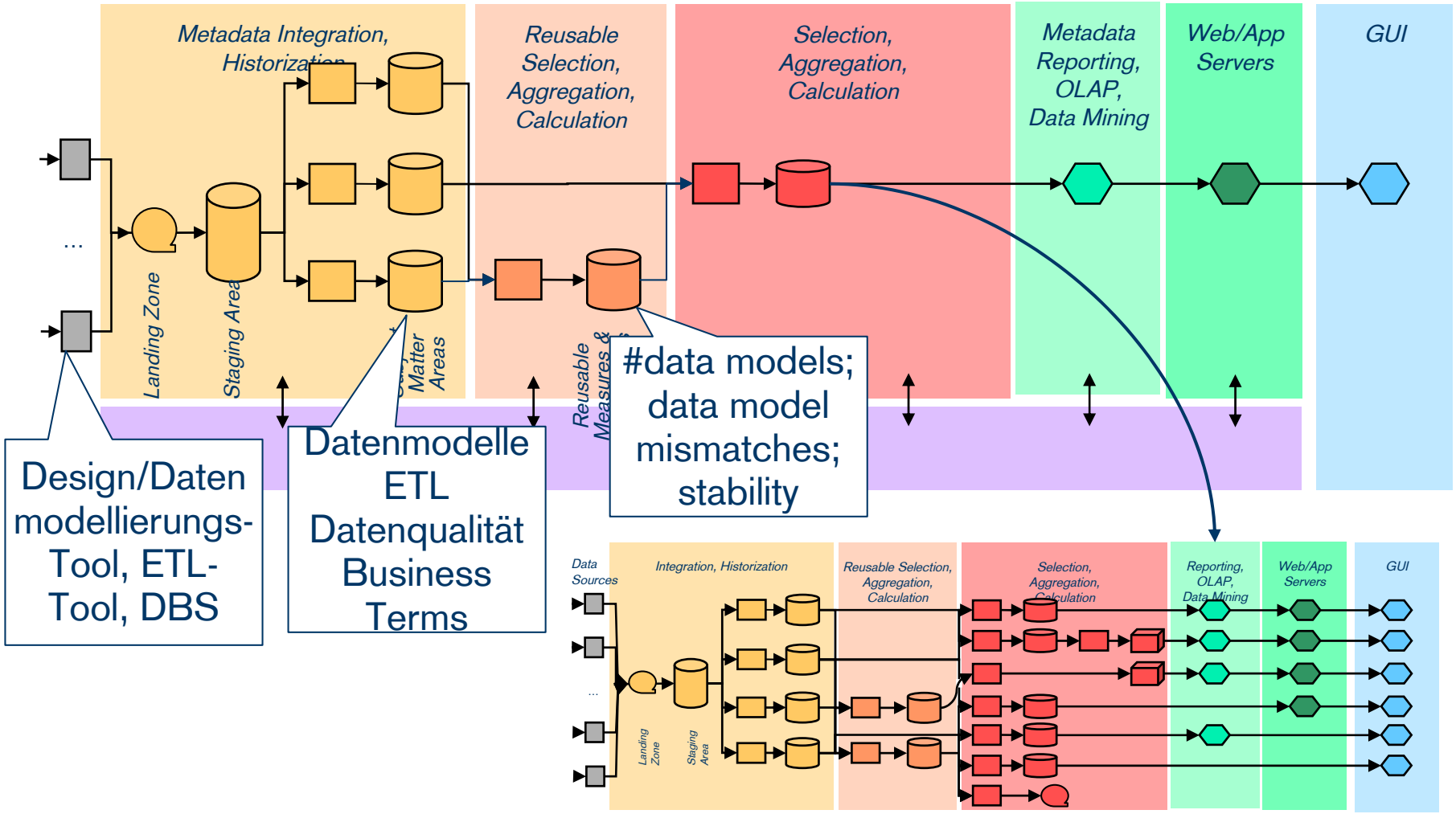


Metadatenverwaltung: DWH-Architektur

- "DWH-Ansatz": autonome Repositorien
- integrierte globale Sicht durch gemeinsames Repositorium

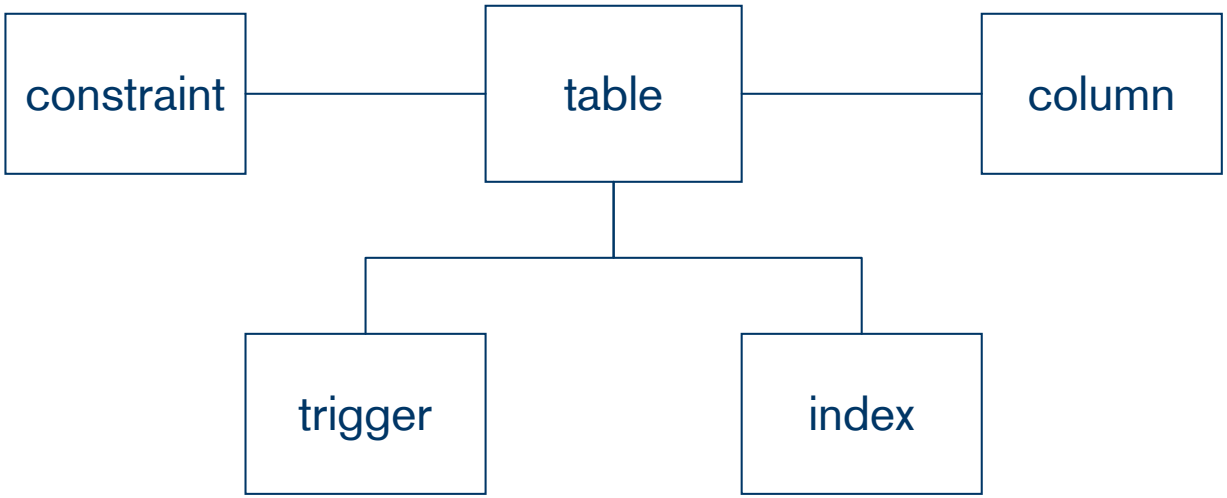


Metadatenverwaltung: DWH-Architektur



Quell-Metadaten: Schema eines DB-Repositories

- DB-Katalog enthält Datenbankmetadaten
- Schemas des Katalogs definiert Metadatenstrukturen (Metametamodel)
- DB2: Schema syscat, enthält ~ 70 Katalogsichter
- Oracle: Schemauser sys, enthält > 3000 Sichten



ALL_TABLES	
OWNER	VARCHAR2 (30)
TABLE_NAME	VARCHAR2 (30)
TABLESPACE_NAME	VARCHAR2 (30)
CLUSTER_NAME	VARCHAR2 (30)
IOT_NAME	VARCHAR2 (30)
STATUS	VARCHAR2 (8)
PCT_FREE	NUMBER
PCT_USED	NUMBER
INI_TRANS	NUMBER
MAX_TRANS	NUMBER
INITIAL_EXTENT	NUMBER
NEXT_EXTENT	NUMBER
MIN_EXTENTS	NUMBER
MAX_EXTENTS	NUMBER
PCT_INCREASE	NUMBER
FREELISTS	NUMBER
FREELIST_GROUPS	NUMBER
LOGGING	VARCHAR2 (3)
BACKED_UP	VARCHAR2 (1)
NUM_ROWS	NUMBER
BLOCKS	NUMBER
EMPTY_BLOCKS	NUMBER
AVG_SPACE	NUMBER
CHAIN_CNT	NUMBER
AVG_ROW_LEN	NUMBER
AVG_SPACE_FREELIST_BLOCKS	NUMBER
NUM_FREELIST_BLOCKS	NUMBER
DEGREE	VARCHAR2 (10)
INSTANCES	VARCHAR2 (10)
CACHE	VARCHAR2 (5)
TABLE_LOCK	VARCHAR2 (8)
SAMPLE_SIZE	NUMBER
LAST_ANALYZED	DATE (7)
PARTITIONED	VARCHAR2 (3)
IOT_TYPE	VARCHAR2 (12)
TEMPORARY	VARCHAR2 (1)
SECONDARY	VARCHAR2 (1)
NESTED	VARCHAR2 (3)
BUFFER_POOL	VARCHAR2 (7)
ROW_MOVEMENT	VARCHAR2 (8)
GLOBAL_STATS	VARCHAR2 (3)
USER_STATS	VARCHAR2 (3)
DURATION	VARCHAR2 (15)
SKIP_CORRUPT	VARCHAR2 (8)
MONITORING	VARCHAR2 (3)
CLUSTER_OWNER	VARCHAR2 (30)
DEPENDENCIES	VARCHAR2 (8)
COMPRESSION	VARCHAR2 (8)
COMPRESS_FOR	VARCHAR2 (18)
DROPPED	VARCHAR2 (3)
READ_ONLY	VARCHAR2 (3)

Metadaten über Metadaten: Schema eines Repositories

- s. Marco 2000

Inhalt

1. Motivation
2. Metadatenverwaltung & DWH
3. **DWH-Metadatenstandards**

Metadatenstandards

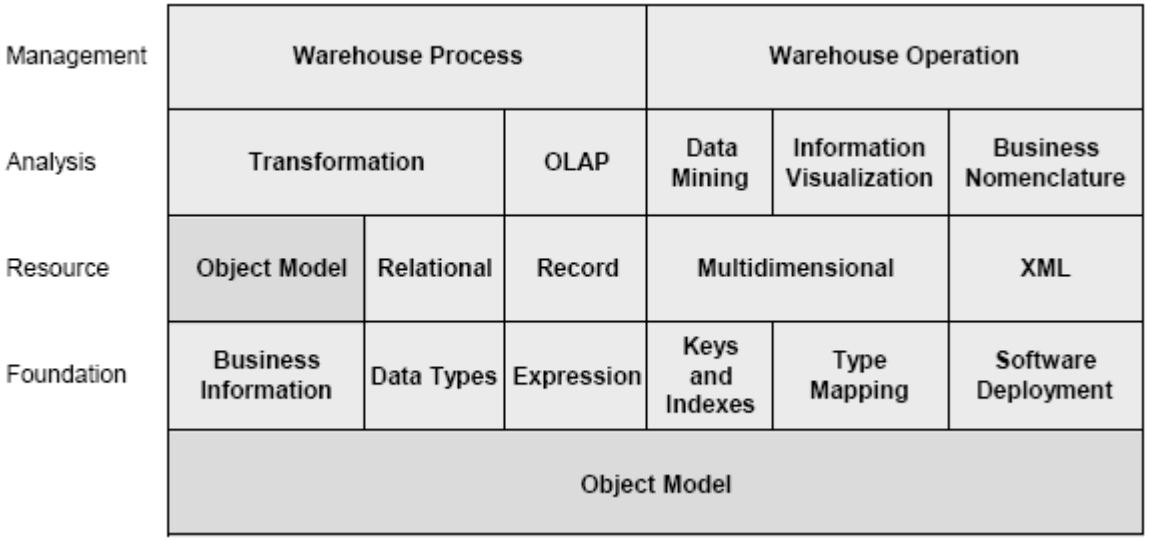
- Open Information Model (OIM)
  - Version 1.0 definiert in 1999 durch Metadata Coalition (MDC)
- Common Warehouse Model (CWM)
  - erste Version definiert in 1999 durch Object Management Group (OMG)
  - einfacher Austausch von DWH-Metadaten zwischen Werkzeugen und Repositorien
  - Modularität, so dass 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



- jedes Paket kann Pakete der gleichen Schicht oder der unteren Schichten referenzieren

CWM: Layer "Foundation"

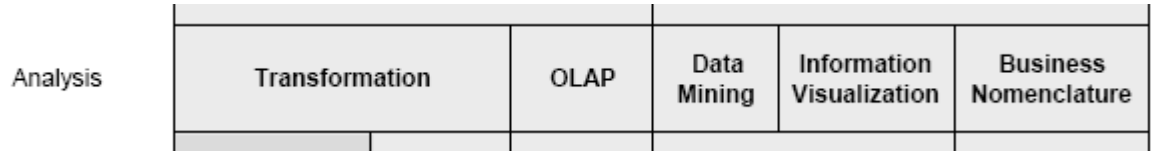
- "Foundation" bietet CWM-spezifische Dienste für andere Packages auf höheren Schichten
- Data Types: Klassen und Assoziationen für die Definition von Datentypen
- Expressions: K&A für die Repräsentation von Ausdrucksbäumen
- Keys and Indexes: K&A, die Schlüssel und Indexe repräsentieren
- Software Deployment: 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

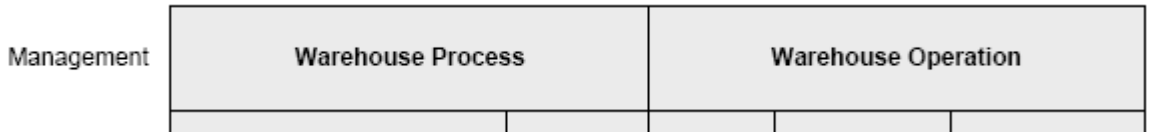
CWM: Layer "Analysis"

- 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



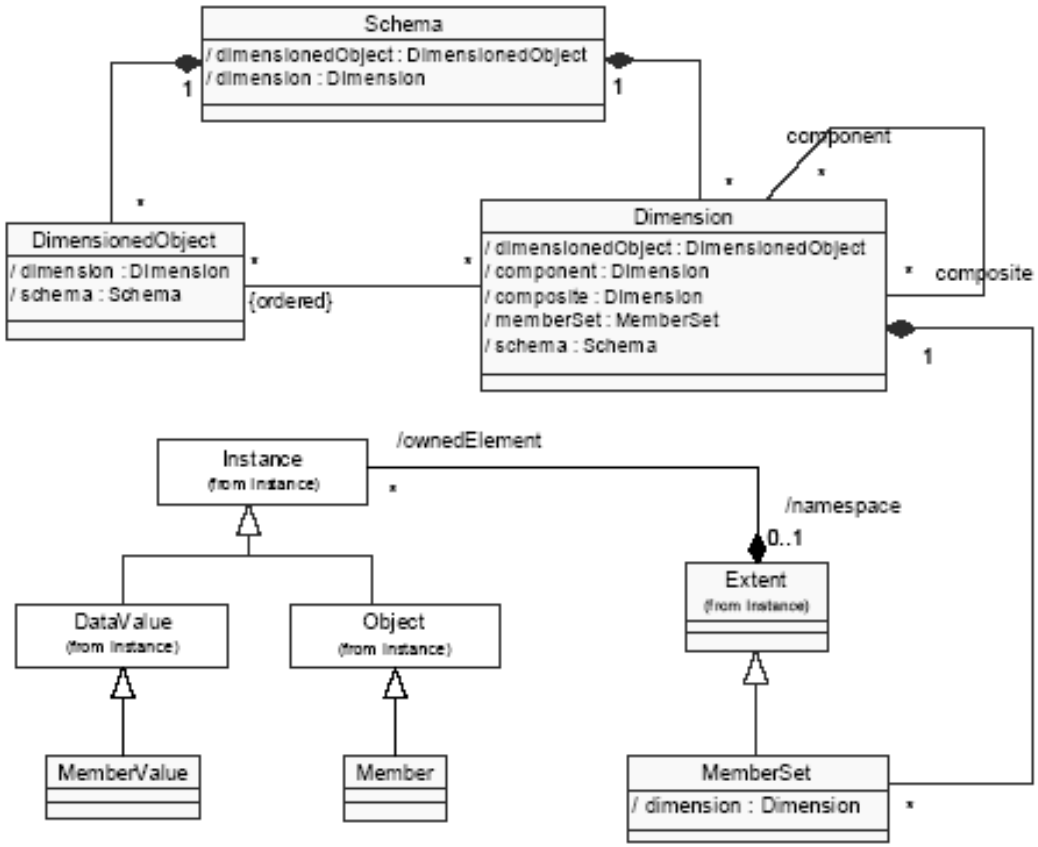
CWM: Layer "Management"

- Warehouse Process: Metadaten über DWH-Prozesse
- Warehouse Operation: Metadaten über DWH-Betrieb (Ergebnisse)



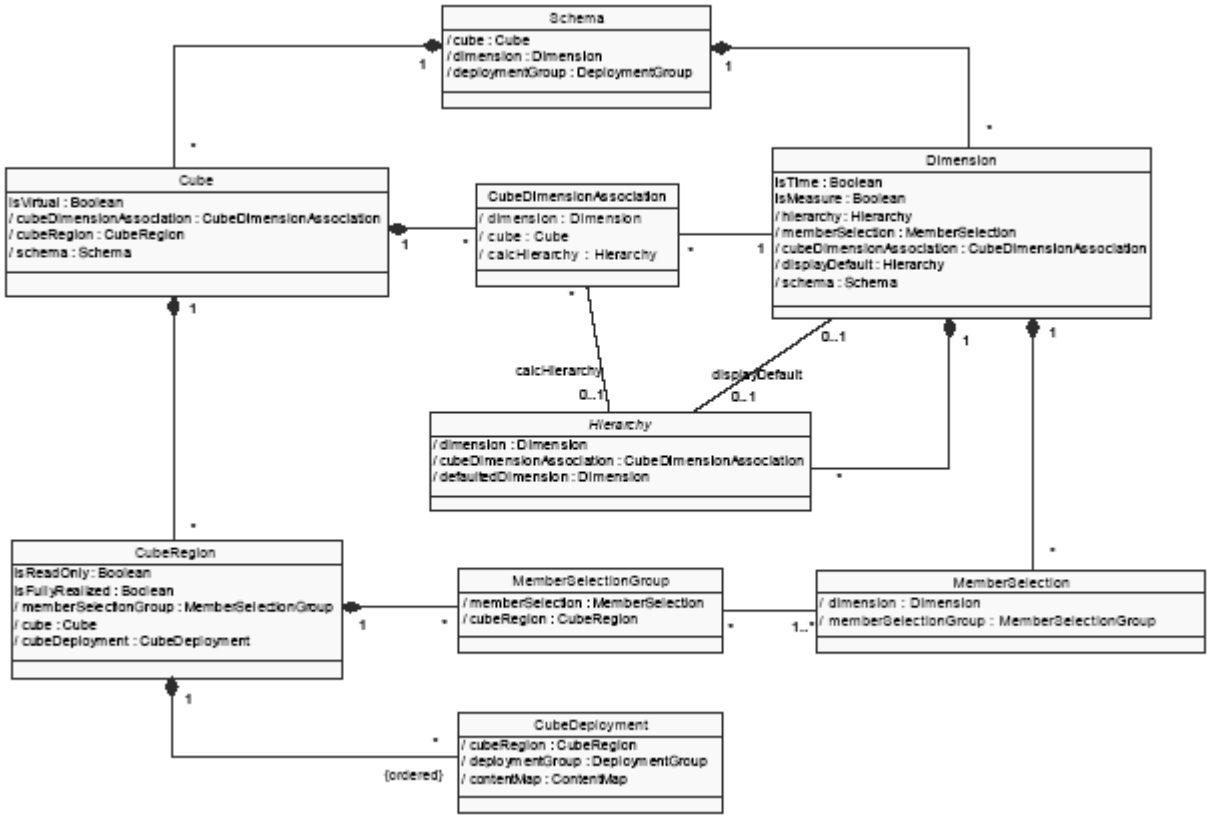
CWM: "Multidimensional" Package

- generische Repräsentation einer multidimensionalen Datenbank



CWM: "OLAP" Package

- generische Repräsentation von OLAP-Konzepten



Zusammenfassung

- Metadaten fallen in allen relevanten Phasen und Komponenten des Data Warehousings an
- umfassende und adäquate Metadatenverwaltung ist Voraussetzung für ein erfolgreiches DWH
- integrierte Metadatenverwaltung ist besonders problematisch bei einer heterogenen Werkzeuglandschaft

⇨ Metadatenverwaltung ein Kriterium bei der Definition von DWH-Plattformen