

Integriertes Metadatenmanagement für das Data Warehousing

Grundlagen, Nutzenpotenziale, Architektur

Verfasser: Florian Melchert, Gunnar Auth,
Clemens Herrmann

Lehrstuhl: Prof. Dr. Robert Winter
Bericht Nr.: BE HSG/CC DW2/03
Version: 2.0
Datum: 2002-04-10

**Universität St. Gallen –
Hochschule für Wirtschafts-, Rechts- und Sozi-
alwissenschaften (HSG)**

Institut für Wirtschaftsinformatik
Müller-Friedberg-Strasse 8
CH-9000 St. Gallen
Tel.: + 41 (0) 71 224 2420
Fax: + 41 (0) 71 224 2777

Prof. Dr. A. Back
Prof. Dr. W. Brenner
Prof. Dr. E. Fleisch
Prof. Dr. H. Österle
Prof. Dr. R. Winter (geschäftsführend)

Das Kompetenzzentrum Data Warehousing 2 (CC DW2)

In grösseren Unternehmen existieren eine Vielzahl verschiedener und häufig sehr heterogener Informationssysteme. Neue Problemstellungen und sich dynamisch verändernde Geschäftsmodelle machen es jedoch erforderlich, dass vorhandene Datenquellen auch integriert, das heisst unabhängig von ihrem operativen Einsatzzweck genutzt werden können. Die Zielvorstellungen reichen von einer einheitlichen Kundensicht bis hin zu schnell verfügbaren Führungsinformationen.

Während im technischen Bereich des Data Warehousing in den vergangenen Jahren erhebliche Fortschritte erzielt worden sind, fehlt es noch immer an gesicherten Erkenntnissen in Bereichen wie:

- Applikationsintegration,
- Entwicklung von Strategien zur Nutzung operativer Applikationen im Data Warehouse,
- Integration des Operational Data Stores (ODS) ins Data Warehouse,
- Kooperation des ODS mit dem Data Warehouse,
- Metadatenmanagement,
- Datenqualität.

Im Rahmen des Kompetenzzentrums Data Warehousing 2 (CC DW2) entwickelt das Institut für Wirtschaftsinformatik der Universität St. Gallen (IWI-HSG) zusammen mit namhaften Partnerunternehmen Methoden und Referenzlösungen für die genannten Bereiche. Die folgenden Unternehmen sind Partner im CC DW2:

- | | |
|------------------------------------|----------------------------------|
| • ARAG Lebensversicherungs-AG (DE) | • SwissRe (CH) |
| • Credit Suisse (CH) | • UBS AG (CH) |
| • Mummert + Partner (DE) | • Winterthur Versicherungen (CH) |
| • Rentenanstalt/Swiss Life (CH) | • W & W AG (DE) |

Durch die entstehenden dedizierten Problemlösungen kann das Data Warehouse als wichtige Komponente langfristig und in wirtschaftlicher Weise in das betriebliche Informationsmanagement integriert werden. Der vorliegende Arbeitsbericht ist dem Themenbereich integriertes Metadatenmanagement für das Data Warehousing zuzuordnen.

Inhaltsverzeichnis

Das Kompetenzzentrum Data Warehousing 2 (CC DW2).....	i
Inhaltsverzeichnis	ii
Abbildungsverzeichnis.....	iii
Tabellenverzeichnis	iii
Abkürzungsverzeichnis	iv
1 Einleitung	1
2 Metadaten und ihre Rolle im Data Warehousing.....	3
2.1 Das Metadatenverständnis in der Domäne Data Warehousing.....	3
2.2 Einsatzmöglichkeiten und Nutzenpotenziale von Metadaten im Data Warehouse.....	8
2.3 MetaMIS als Beispiel für die Nutzung von Metadatenmanagement im DWH-Entwicklungsprozess	10
2.4 Nutzenpotentiale einzelner Metadatenkategorien aus Sicht der Praxis	13
3 Technische Metadatenintegration im Data Warehousing	16
3.1 Metadatenmanagementsysteme	16
3.2 Architekturvarianten für Metadatenmanagementsysteme	16
3.2.1 Zentrale Architektur	17
3.2.2 Dezentrale Architektur	19
3.2.3 Hybride Architektur – Metadata Warehouse	20
3.3 Technische Metadatenintegration am Beispiel einer dezentralen Architektur	23
4 Standardisierung als Basis für die Metadatenintegration	26
4.1 Der Nutzen von Standards für die Integration von Metadaten	26
4.2 Das Common Warehouse Metamodel als Standard für DWH-Metadaten	28
4.2.1 Einführung in das CWM-Konzept	28
4.2.2 Vorstellung der CWM-Packages.....	30
4.3 Möglichkeiten der Erweiterung des CWM	36
4.4 Fazit	39
5 Zusammenfassung und Ausblick	40
Literatur	42

Abbildungsverzeichnis

Abb. 2-1:	Realität, ER-Modell und ER-Metamodell	6
Abb. 2-2:	Modellierung des Informationsbedarfs mithilfe des Werkzeugs MetaMIS Designer	12
Abb. 2-3:	Einschätzung der CWM-Packages aus Sicht der Praxis	14
Abb. 3-1:	Zentrale Architektur für ein Metadaten-Repository.....	18
Abb. 3-2:	Dezentrale Architektur für ein Metadaten-Repository.....	19
Abb. 3-3:	Hybride Architektur für ein Metadaten-Repository.....	21
Abb. 3-4:	Architektur eines DWH mit integriertem Metadata Warehouse.....	22
Abb. 3-5:	Architektur des Showcase-Systems für Metadaten austausch	24
Abb. 4-1:	Vergleich der Schnittstellenanzahl zwischen bilateralem und standardbasiertem Metadaten austausch	27
Abb. 4-2:	Ebenen der CWM-Modellierung	29
Abb. 4-3:	Vererbungshierarchie aller Klassen des CWM.....	31
Abb. 4-4:	Beispiel für die Mehrfach-Verwendung von Assoziationen im CWM.....	32
Abb. 4-5:	Die Packages des CWM.....	33

Tabellenverzeichnis

Tab. 2-1:	Pragmatische Aufzählung von Kategorien für DWH-Metadaten.....	4
Tab. 2-2:	Einsatz- und Nutzenpotenziale von Metadaten.....	9
Tab. 2-3:	Priorisierung von Metadaten-Kategorien.....	15

Abkürzungsverzeichnis

ACM	Association of Computing Machinery
API	Application Programming Interface
BIG	Business Information Guide
CC DW2	Competence Center Data Warehousing 2
CWM	Common Warehouse Metamodel
DB	Datenbank
DBMS	Datenbankmanagementsystem
DSS	Decision Support System
DTD	Document Type Definition
DWC	Data Warehouse Catalog
DWH	Data Warehouse
ER	Entity Relationship
ETL	Extraktion, Transformation, Laden
MD	Metadaten
MDM	Metadatenmanagement
MDMS	Metadatenmanagementsystem
MMS	Meta-Metasprache
MOF	Meta Object Facility
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OMG	Object Management Group
UML	Unified Modelling Language
WWW	World Wide Web
XMI	XML Metadata Interchange
XML	Extensible Markup Language

1 Einleitung

Seit einiger Zeit hat sich das Data Warehousing als fester Bestandteil der IT-Infrastruktur grosser Unternehmen etabliert und ermöglicht heute meist schon eine gute Versorgung der Fach- und Führungskräfte mit entscheidungsrelevanten Informationen. Die Basis für das Funktionieren dieser Informationsversorgung bildet eine Vielzahl unterschiedlichster Metadaten, die im Rahmen von Erstellung, Betrieb und Nutzung des Data Warehouse von allen beteiligten Systemkomponenten fortlaufend produziert und konsumiert werden.

Über ihren eigentlichen Verwendungszweck hinaus wird Metadaten aber auch erhebliches Potenzial zugesprochen, das Data Warehousing sowohl im Hinblick auf die Effizienz bei der Entwicklung als auch hinsichtlich einer effektiven Nutzung entscheidend zu verbessern. Aus Nutzersicht kann die Effektivität erhöht werden, indem das Auffinden und Interpretieren relevanter Daten bei der Analyse durch Metadaten unterstützt werden. Effizienzsteigerungen werden bei Entwicklung und Betrieb des Data-Warehouse-Systems erzielt, wenn Metadaten zur Wiederverwendung von Entwicklungsbausteinen, Automatisierung von Prozessen sowie konsistenten und integrierten Dokumentation des Data-Warehouse-Systems genutzt werden und dadurch zu einer Reduzierung des Entwicklungsaufwandes beitragen.

Um das Potenzial der Metadaten in der beschriebenen Weise ausnutzen zu können, ist es erforderlich, die meist lokal genutzten und verwalteten Metadatenbestände einzelner Komponenten zu integrieren, d. h. sie allen potenziellen Nutzern in einer einheitlichen Form zur Verfügung zu stellen. Die Gesamtheit diesbezüglicher Prozesse zur Integration, Verwaltung, Nutzung aber auch Generierung von Metadaten wird als Metadatenmanagement bezeichnet. Als Enabler des Metadatenmanagement kommen wie auch im Data Warehousing verschiedene Softwarekomponenten zum Einsatz, die als Metadatenmanagementsystem bezeichnet werden.

Ziel dieses Arbeitsberichtes ist es, das integrierte Metadatenmanagement im Data Warehousing hinsichtlich konzeptioneller, technischer und organisatorischer Aspekte näher zu beleuchten. Hierbei werden theoretische Erkenntnisse mit den Erfahrungen kombiniert, die die Arbeitsgruppe des Kompetenzzentrums Data Warehousing 2 im Rahmen mehrerer Workshops zum Thema Metadatenmanagement sowie in bilateralen Praxisprojekten sammeln konnte.

Ausgehend von einer grundlegenden Diskussion des Metadatenbegriffes werden zunächst die Einsatzmöglichkeiten und Nutzenpotenziale von Metadaten im Data Warehousing spezifiziert. Im Anschluss daran werden verschiedene Varianten für die Architektur von Metadatenmanagementsystemen vorgestellt und hinsichtlich ihrer Vor- und Nachteile diskutiert. Ein eigenes Kapitel wird dem Thema Standards für Metadatenintegration gewidmet, wobei insbesondere das Common Warehouse Metamodel der Object Management Group näher vorgestellt und auf mögliche Schwachstellen bzw. erweiterbare Bereiche untersucht wird. Die gewonnenen Erkenntnisse werden schliesslich in einer Zusammenfassung überblicksartig dargestellt.

2 Metadaten und ihre Rolle im Data Warehousing

2.1 Das Metadatenverständnis in der Domäne Data Warehousing

Metadaten und ihre integrierte Verwaltung gelten sowohl in der Wissenschaft als auch in der Praxis als viel versprechendes Mittel zur Verbesserung der Nutzung von Data-Warehouse-Systemen durch Endbenutzer sowie zur Senkung des Entwicklungsaufwands bei Betrieb, Pflege und Wartung dieser Systeme [vgl. z. B. Widom 1995; Hurwitz 1998; Tozer 1999, Augusta 2001]. Hinsichtlich einer Definition des Metadatenbegriffes besteht weitgehend Einigkeit darüber, dass der oft zu lesende Satz „Metadaten sind Daten über Daten“ für eine differenzierte Betrachtung des Themenkomplexes zu kurz greift [vgl. z. B. ComputerWire 1996; Schwarz 2000, S. 103, Wiener 2000], da suggeriert wird, dass Metadaten auf die Beschreibung von Objektdaten beschränkt sind. Metadaten, die bspw. der Beschreibung von Prozessen oder Organisationsstrukturen dienen, werden hingegen völlig vernachlässigt [vgl. Devlin 1997, S. 52]. Darüber hinaus liefern sehr allgemeine Definitionen wie die genannte keine Kriterien, mithilfe derer zwischen Daten und Metadaten unterschieden werden kann [vgl. Inmon 2001, S. 3]. Aufgrund der Tatsache, dass Metadaten im Data Warehousing eine Fülle unterschiedlichster Aufgaben unterstützen, fällt es allerdings schwer, eine besser geeignete Definition zu entwickeln, die die Problemdomäne einerseits exakter umreißt, andererseits aber auch alle inhaltlichen Aspekte abdeckt [vgl. Staudt/Vaduva/ Vetterli 1999]. Ohne dieses Problem zu lösen behilft man sich bei der näheren Erläuterung des Metadatenbegriffs häufig dadurch, dass der relativ aussageschwachen Definition eine Klassifikation oder Enumeration von Metadatentypen anbeigestellt wird [vgl. z. B. Inmon 2001; Barkin/Edelstein 1997f.; Kimball 1998, S. 22f.]. Tab. 2-1 zeigt ein Beispiel einer solchen Aufzählung, welche im Rahmen eines Workshops des CC DW2 erarbeitet wurde. Zu seiner Erstellung benutzte die beteiligte Arbeitsgruppe das Common Warehouse Metamodel (CWM) [s. hierzu auch Kap. 4] der Object Management Group als Diskussionsgrundlage und entwickelte darauf aufbauend weitere Kategorien, die nach ihrer Meinung nicht durch die Teilmodelle des CWM abgedeckt werden. Die Übersichten dieser Art vermitteln durchaus ein besseres Verständnis des Problemereichs, jedoch eignen sie sich nicht als Bestandteil einer Definition, da ihre Vollständigkeit aufgrund des Facettenreichtums des Metadatenbegriffes als unwahrscheinlich eingestuft werden muss.

Metadatenkategorie	Beschreibung des Inhalts
Business Nomenclature	Fachbegriffsmodelle, Ontologien, Taxonomien
Warehouse Operation	Dokumentation technischer DWH-Prozesse, Messkriterien
Warehouse Process	Schedules, Ausführungslogs technischer DWH-Prozesse
Software Deployment	Architekturbeschreibungen von Metadatenssystemen
Analytische Informationssysteme	Technische Beschreibung der Analysestrukturen des DWH (Dimensionen, Kennzahlen), mathematischer Modelle zur statistischen Datenanalyse sowie der verfügbaren Varianten zur Darstellung von Analyseergebnissen
Warehouse Content	Verzeichnis verfügbarer Berichte, Verantwortlichkeiten
Datenmodell	Modelle der DWH-Datenbank auf konzeptioneller, logischer und physischer Ebene
Autorisierung	Rollen, Zugriffsrechte
Transformation	Konzeptionelle Beschreibung technischer Extraktions-, Transformations- und Ladeprozesse
Business Rules	Datenqualitätsanforderungen, Dokumentation organisatorischer DWH-Prozesse, Regeln zur Automatisierung organisatorischer Regelungen
Data Type	Technische Beschreibung von Datentypen, Ausdrücken, Schlüsseln und Indizes

Tab. 2-1: Pragmatische Aufzählung von Kategorien für DWH-Metadaten

Die einzige geeignete Variante, sich einer möglichst treffenden *Definition* für Metadaten zu nähern, sehen die Autoren hier jedoch im Rückgriff auf theoretische Überlegungen zum Begriff Metadaten. Im Folgenden wird daher zunächst die Sprachstufentheorie vorgestellt, um darauf aufbauend die grundlegenden Eigenschaften von Metadaten zu entwickeln.

Die Sprachstufentheorie als Basis für eine Definition von Metadaten

Im Umfeld der Wirtschaftsinformatik und in verwandten Disziplinen deutet die Verwendung der Vorsilbe „Meta“ (aus dem Alt-Griechischen u. a. für „danach“, „darüber hinaus“) darauf hin, dass Erkenntnisse über ein Erkenntnisobjekt verallgemeinert und auf eine höhere Abstraktionsstufe übertragen werden. Somit werden diese Erkenntnisse selbst zum Gegenstand der Betrachtung [Strahringer 1999, S. 1]. Beim Übergang von einem Betrachtungsgegenstand zu einem abstrakteren, „darüberliegenden“ Betrachtungsgegenstand vollzieht der Betrachter einen Wechsel der Bezugsebene. Ein oft genutztes Beispiel für einen solchen Wechsel der Bezugsebene stellt der Begriff des Metamodells dar. Dabei werden Modelle und Modellbildung selbst zum Gegenstand der Modellierung [Strahringer 1998, S. 1]. Laut Ferstl und Sinz

legt ein Metamodell „das zur Spezifikation von Modellsystemen verfügbare Begriffssystem fest“ [Ferstl/Sinz 2001, S. 122]. Der Hinweis auf ein Begriffssystem in dieser Definition macht deutlich, dass Aussagen über einen zu untersuchenden Gegenstandsbereich, wie sie bei der Modellierung getroffen werden, typischerweise in Sprache formuliert werden [vgl. z. B. Ortner 1998; Strahringer 1999]. Nachfolgend wird daher die Sprachstufentheorie als Grundlage zur Erklärung von Meta-Begriffen und schliesslich zur Definition von Metadaten herangezogen.

Bei einem Wechsel der Bezugsebene werden die in Sprache gefassten Aussagen selbst zum Erkenntnisobjekt einer Untersuchung, in deren Verlauf in Sprache über Sprache gesprochen wird. Strahringer weist auf die Gefahr der Verwirrung hin, die entstehen kann, „wenn Aussagen nicht nur über aussersprachliche Gebilde, sondern auch über sprachliche Gebilde selbst gemacht werden“ [Strahringer 1999, S.1]. Sie führt weiter aus, dass zur Vermeidung dieser Gefahr in der Logik verschiedene Sprachebenen, sog. semantische Stufen, unterschieden werden. Demnach wird die Sprache, mit der Aussagen über das anfängliche Erkenntnisobjekt spezifiziert werden, als Objektsprache bezeichnet. Wird die Objektsprache selbst wiederum Gegenstand einer Untersuchung auf einer Metaebene, heisst die Sprache, in der diese Untersuchung erfolgt, Metasprache. Demzufolge kann eine Sprache immer nur in Bezug auf eine andere Sprache (die Objektsprache) Metasprache sein. Die Abstraktion von einer Objektsprache zu einer Metasprache lässt sich rekursiv anwenden, wodurch eine Hierarchie von Sprachebenen entsteht. Zu einer Metasprache lässt sich erneut eine Metasprache bilden, mit der Aussagen über die erste Metasprache formuliert werden. Diese Metasprache ist bezüglich der ursprünglichen Objektsprache eine Meta-Metasprache. Um eindeutig bezeichnet werden zu können, werden die Sprachen mit einem Index i versehen, man spricht von Metasprachen i -ter Stufe (M^iS). Nach dieser Konvention würde beispielsweise in einer 3-stufigen Sprachebenen-Hierarchie die Objektsprache mit S bzw. M^0S bezeichnet, die Metasprache mit MS bzw. M^1S und die Meta-Metasprache mit MMS bzw. M^2S . Ein konkretes Anwendungsbeispiel für eine Sprachebenen-Hierarchie ist die Meta-Object Facility (MOF) der Object Management Group (OMG)¹ [vgl. OMG 2001].

¹ Die OMG (<http://www.omg.org>) ist ein 1989 gegründetes Industriekonsortium mit dem Ziel, herstellerunabhängige Spezifikationen für die Softwareindustrie zu entwickeln. Bekannt ist sie vor allem für ihre CORBA- und UML-Spezifikationen. Der OMG gehören mittlerweile über 800 Organisationen an, darunter die führenden Software-Hersteller. Die MOF (<http://www.omg.org/technology/cwm>) ist ein objektorientierter Information Modeling Framework zur Integration heterogener Applikationen und zum Entwurf homogener Applikationen.

Die Unterscheidung in Objekt- und Metasprache stellt jedoch keine Klassifikation im strengen Sinne dar, sondern dient lediglich der Bildung von Bezugsebenen. Das Merkmal Objekt- oder Metasprache ist kein absolutes Merkmal, sondern ausschliesslich relativ zu einer anderen Sprache zu verstehen. Die Unterscheidung ist also stets abhängig vom Kontext, in dem die Sprache verwendet wird. Dabei kann es sich auch um ein und dieselbe Sprache handeln. Zum Beispiel lässt sich das Metamodell für die Modellierungssprache Entity-Relationship-Modell wiederum als Entity-Relationship-Modell beschreiben.

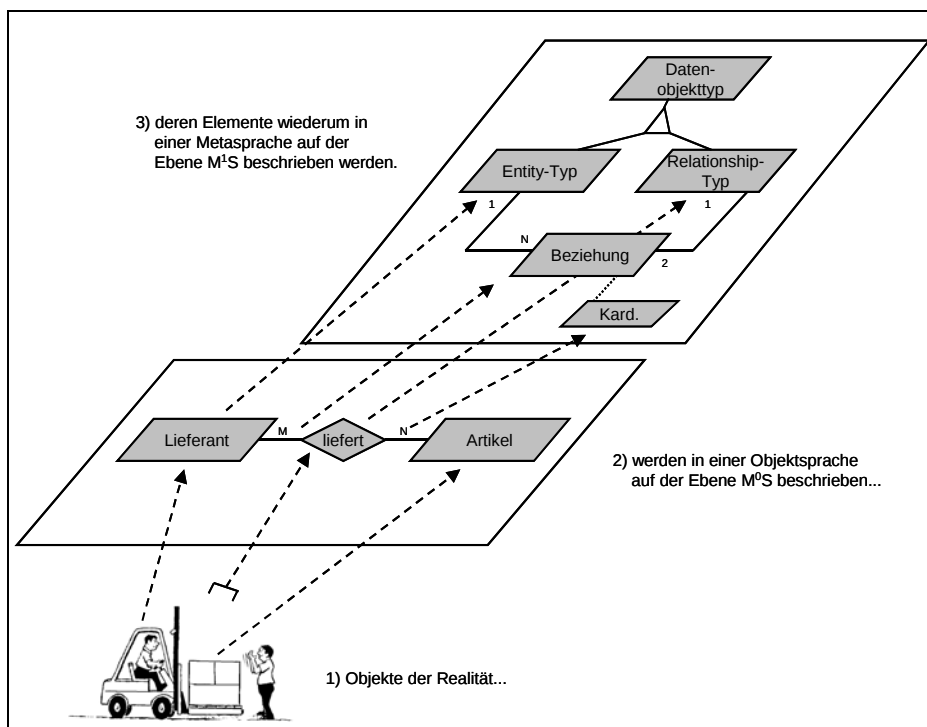


Abb. 2-1: Realität, ER-Modell und ER-Metamodell

Definition des Metadaten-Begriffs auf Basis der Sprachstufentheorie

Hilfreich sowohl für das Verständnis von Metadaten als auch für eine spätere Modellierung ist es, sich den Prozess der Abstraktion bei der Erzeugung von Metadaten bewusst zu machen. Mit der Vorsilbe „Meta“ werden diejenigen Daten gekennzeichnet, die auf einer höheren Abstraktionsebene die Daten einer darunterliegenden Ebene beschreiben. Bei den von den Metadaten dokumentierten Daten kann es sich bspw. um solche Daten handeln, die von Applikationen zur Unterstützung der Geschäftsprozesse gespeichert werden. Zur schärferen Abgrenzung von den sie beschreibenden Metadaten werden diese Daten in Anlehnung an die Terminologie der Sprachstufentheorie als Objektdaten bezeichnet. Objektdaten repräsentieren Objekte und deren Beziehungen untereinander bezogen auf einen bestimmten Gegenstandsbe-

reich. Neben der Sprachstufentheorie ist der Begriff durch das Objekt-Paradigma motiviert, dessen Ziel Khoshafian und Abnous angeben als: „modeling the real world as close to a user’s perspective as possible“ [Khoshafian/Abnous 1995, S. 3]. Nachfolgend wird der Begriff Objektdaten dann verwendet, wenn die Verwendung des Begriffs Daten unklar lässt, ob Meta- oder Objektdaten gemeint sind. In allen anderen Fällen wird der Begriff Daten abkürzend für Objektdaten verwendet. Die Unterscheidung zwischen Objektdaten und Metadaten soll anhand eines digital gespeicherten Auftrags verdeutlicht werden: Bei den Auftragspositionen, den Mengenangaben zu den Positionen und der Adresse des Auftraggebers handelt es sich um Objektdaten. Die Struktur und die Semantik des Auftrags wird durch Metadaten beschrieben, bspw. durch eine Record- oder Tabellendefinition oder eine XML Document Type Definition (DTD).

Auf Basis der theoretischen Vorarbeit lassen sich nun drei Charakteristiken identifizieren, die Metadaten von Objektdaten differenzieren:

- Metadaten stehen immer in Bezug zu Objektdaten. Sofern für einen als Metadaten ausgewiesenen Datenbestand keine korrespondierende Menge an Objektdaten identifiziert werden kann, handelt es sich nicht Metadaten.
- Metadaten beschreiben niemals Objekte der Realwelt sondern nur Daten, die diese Objekte repräsentieren
- Metadaten befinden sich auf einer höheren Abstraktionsstufe als Daten. Im Objekt-paradigma entspricht dies dem Unterschied zwischen Objekten und Objekttypen oder Objektklassen.

Ein wichtiges Charakteristikum des Metadatenbegriffs ist zudem seine Relativität. Metadaten einer Abstraktionsstufe e stehen immer in Bezug zu Objektdaten auf der nächst niedrigeren Abstraktionsstufe $e-1$, können darüber hinaus aber auch selbst durch Meta-Metadaten beschrieben werden, die auf einer noch höheren Abstraktionsstufe $e+1$ liegen. Betrachtet man nur die beiden höheren Abstraktionsebenen e und $e-1$, so stellen die Metadaten auf der Ebene e zugleich Objektdaten dar, die durch Metadaten auf der Ebene $e+1$ beschrieben werden, die wir ursprünglich als Meta-Metadaten bezeichnet haben.

Die bisherigen Ausführungen haben gezeigt, dass sich der Begriff Metadaten zwar hinsichtlich seiner theoretischen Eigenschaften recht gut beschreiben lässt, dass es allerdings auch schwer fällt, das Anwendungsszenario der Metadaten im Data Warehousing klar abzustecken.

Statt einer nur bedingt hilfreichen Definition von DWH-Metadaten wird im Folgenden näher untersucht, zu welchem Nutzen Metadaten im Data Warehousing überhaupt führen.

2.2 Einsatzmöglichkeiten und Nutzenpotenziale von Metadaten im Data Warehouse

Ein Grundgedanke des Einsatzes von Metadaten ist, durch sie das Verständnis von Datenelementen und ihren Beziehungen zu verbessern. In Verbindung mit dem betrieblichen Informationsmanagement unterstützen und ermöglichen Metadaten bereits seit langem die Administration von Datenhaltungssystemen (Datenbanken, Dateisystemen usw.), die Organisation von Benutzer- und Zugriffsberechtigungskonzepten und die Dokumentation von Verknüpfungen zwischen Datenfeldern, Programmen und Programm- oder Systemschnittstellen [vgl. Narayan 1988]. In diesem Zusammenhang werden Metadaten überwiegend in der Daten(bank)administration oder der Software-Entwicklung zielgerichtet genutzt, d. h. im operativen Bereich des Informationsmanagements. Das Bewusstsein über die Notwendigkeit und den Nutzen von Metadaten hat jedoch im Rahmen des Data Warehousing eine neue Bedeutung erlangt, da die integrierte Datenbasis des DWH-Systems beliebige Analysen ermöglichen soll und nicht wie bei operativen Systemen auf die Durchführung gleich bleibender Abläufe ausgerichtet ist [vgl. Mucksch/Behme 2000, S. 15].

Staudt, Vaduva, Vetterli unterscheiden drei unterschiedliche Arten der Nutzung von Metadaten im Data Warehousing [vgl. Staudt/Vaduva/Vetterli 1999, S. 5 f.]:

1. *Passive Nutzung*

Hierbei werden Metadaten genutzt, um eine umfassende und konsistente Dokumentation für Aufbau, Entwicklung und Nutzung des DWH-Systems zur Verfügung zu stellen. Diese Dokumentation ist mit einem herkömmlichen Hilfesystem für eine Desktop-Anwendung (z. B. Textverarbeitung) zu vergleichen. Zielgruppe sind sämtliche am Data Warehousing beteiligten Personen, vom Entwickler bis zum Endbenutzer. Eine solche Dokumentation kann darüber hinaus den Wissenstransfer zwischen Mitarbeiter fördern, indem Wissensträger explizit in Zusammenhang mit ihren Spezialgebieten benannt werden.

2. *Aktive Nutzung*

Metadaten können aktiv genutzt werden, indem semantische Beschreibungen von Komponenten des DWH-Systems in Form von Metadaten gespeichert werden (z. B. Transformationsregeln). Zur Laufzeit des DWH-Systems werden diese Metadaten interpretiert und

steuern bspw. die Ausführung von Datenbewegungsprozessen. In diesem Zusammenhang wird auch vom metadatengesteuerten DWH-System gesprochen [vgl. Volck 2000].

3. Semi-aktive Nutzung

Von semi-aktiver Nutzung wird gesprochen, wenn Metadaten zur Laufzeit zwar gelesen, aber nicht zur Steuerung genutzt werden. Hierbei handelt es sich normalerweise um primär statische Metadaten wie z. B. Strukturdefinitionen oder Konfigurationspezifikationen.

Über diese drei Nutzungsarten hinaus lassen sich bei der Betrachtung von Nutzenpotenzialen für Metadaten im Data Warehousing grundsätzlich zwei Sichten unterscheiden [vgl. Vaduva/Dittrich 2001, S. 5]: Aus Sicht der Endbenutzer dienen Metadaten zur Steigerung der Effektivität und Effizienz bei der Nutzung des DWH-Systems. Aus Sicht der für Aufbau und Betrieb des DWH-Systems verantwortlichen Mitarbeiter dienen Metadaten zur Senkung des Aufwands bei der Durchführung ihrer Aufgaben durch Wiederverwendung und Automatisierung. Tab. 2-2 zeigt, welche Teilnutzenaspekte diese beiden übergeordneten Nutzenaspekte beinhalten.

Endbenutzersicht	Entwicklersicht
Verbesserung der DWH-Nutzung durch: ■ Verbesserung der Datenanalyse, insbesondere Auffinden und Interpretieren relevanter Daten ■ Verbesserung der Interaktion mit dem DWH-System ■ Forcieren eines einheitlichen Begriffssystems ■ Reduzierung des Schulungsbedarfs ■ Verbesserung der Datenqualität ■ Wissenstransfer durch Experten	Reduzierung des Entwicklungsaufwandes durch: ■ Automatisierung von Administrationsprozessen ■ Verbesserung der Wiederverwendbarkeit von Datenstrukturen und Code-Modulen ■ Konsistente und integrierte Dokumentation ■ Unterstützung bei der Datenintegration ■ Auswirkungsanalysen ■ Unterstützung bei der Zugriffskontrolle ■ Verbesserung der Datenqualität ■ Wissenstransfer durch Experten

Tab. 2-2: Einsatz- und Nutzenpotenziale von Metadaten

Durch das permanente Wachstum der DWH-Datenbasis wird es für den Endbenutzer zunehmend schwierig, die Orientierung zu behalten und die gesuchten Daten aufzufinden. Ebenfalls stark eingeschränkt wird die Nutzung des DWH-Systems, wenn die Bedeutung der Daten unklar oder unbekannt ist. Durch die oft komplexen Transformationsprozesse, welche die Daten

auf ihrem Weg von den operativen Systemen über die Data-Warehouse-Datenbank bis hin zu den für den Endbenutzer-Zugriff verfügbaren, analytischen Applikationen durchlaufen, ist es für den Endanwender oft nicht mehr nachvollziehbar, aus welchen Quellsystemen die Daten stammen und von welcher Qualität sie sind. Solche Daten sind nicht geeignet, den Benutzer bei Entscheidungen zu unterstützen. Dies führt dazu, dass das eigentliche Ziel des Data Warehousings, integrierte Daten für die Unterstützung von Entscheidungsprozessen bereitzustellen [vgl. Inmon 1996, S. 33], nicht erreicht werden kann. Durch die gezielte Verwaltung und Bereitstellung von Metadaten kann dagegen die Effektivität und Effizienz des Data Warehousings gesteigert werden.

Entwickler und Administratoren profitieren vor allem von metadatengesteuerten Systemarchitekturen, die ein erhöhtes Mass an Wiederverwendbarkeit von Komponenten durch bessere Dokumentation bieten, die Automatisierung von Administrationsprozessen ermöglichen und Auswirkungsanalysen vor geplanten Systemänderungen erlauben. Dadurch wird der Entwicklungsaufwand insgesamt signifikant gesenkt [vgl. Vaduva/Vetterli 2001; Vaduva/Dittrich 2001, S. 5; Wu/Miller/Nilakanta 2000, S. 118]. Werden technische Metadaten zusammengeführt und für menschliche Nutzer zugänglich gemacht, können sie als konsistente Dokumentation der Datenstrukturen und -flüsse des DWH-Systems genutzt werden [Staudt/Vaduva/Vetterli 1999b, S. 43].

Ein reales Beispiel dafür, wie Metadaten eingesetzt werden könnten, um den Entwicklungsaufwand für DWH-basierte Führungsinformationssysteme zu verringern und gleichzeitig die Nachhaltigkeit des realisierten DWH-Systems zu verbessern, wird im folgenden Abschnitt vorgestellt.

2.3 MetaMIS als Beispiel für die Nutzung von Metadatenmanagement im DWH-Entwicklungsprozess

Ein wesentlicher Schritt des DWH-Entwicklungsprozesses besteht in der Erfassung der Informationsbedarfe der späteren DWH-Nutzer, die durch das DWH abgedeckt werden sollen. Hierzu werden i. d. R. zunächst Informationsbedarfsanalysen durchgeführt und deren Ergebnisse in einer Form dokumentiert, die auf den jeweiligen Anwendungszweck zugeschnitten ist. So werden bspw. Stellenbeschreibungen, aus denen sich Informationsbedarfe ableiten lassen, häufig in Form von Textdokumenten festgehalten, für das Layout konkreter Berichte werden evtl. beispielhafte Excel-Spreadsheets angelegt. Bei der Implementierung werden sich die Entwickler dann zwar auf diese Dokumente stützen, um das zu konstruierende DWH an

den Bedürfnissen der Nutzer auszurichten. Eine automatische Überführung der Ergebnisse der Informationsbedarfsanalyse in logische oder gar physische DWH-Strukturen wird i. d. R. allerdings nicht erfolgen. Der Medienbruch zwischen Analyseergebnissen und DWH-Implementierung führt nicht nur zu einem hohen manuellen Überführungsaufwand seitens der DWH-Entwickler sondern auch zu einer Entkopplung der Entwurfs- und der Implementierungsphase. Diese Entkopplung stellt eine potenzielle Quelle für Inkonsistenzen dar, da jede Änderung in einer der beiden Phasen die manuelle Anpassung in der jeweils anderen Phase nötig macht. Eine Lösung für diese Problematik kann in der konsequenten Umsetzung eines Metadatenmanagement gesehen werden.

In seinem Ansatz zur metadatengestützten Entwicklung von Führungsinformationssystemen stellt HOLTEN eine Methode vor, mit der Informationsbedarfe zunächst auf Fachkonzeptebene und unabhängig von der technischen Realisierung modelliert und anschliessend mit weitgehender Rechnerunterstützung in technische Implementierung überführt werden können. Zur informationstechnischen Unterstützung dieses Prozesses dient der Software-Prototyp MetaMIS Toolset.

Abb. 2-2 zeigt ein die Komponente MetaMIS Designer, die es erlaubt, Informationsbedarfe auf Basis einer an der ER-Notation orientierten Modellierungstechnik zu dokumentieren, welche auf die speziellen Anforderungen der FIS-Entwicklung zugeschnitten ist. Hierbei werden zunächst Auswertungsdimensionen definiert (Dimension), thematisch in Gruppen (Dimension Groupings) zusammengefasst sowie Kennzahlssysteme (Ratio System) erstellt. Kennzahlssysteme und Dimensionen werden dann zu einem sog. Informationsobjekt (Information Object) kombiniert, welches die relevanten Auswertungsmöglichkeiten für einen speziellen Aufgabenbereich definiert. Durch die zentrale Verwaltung aller Informationsobjekte einer Unternehmung lässt sich nach Ansicht HOLTENS zunächst Konsistenz der an das DWH-Entwicklungsteam gestellten Anforderungen erreichen. Mithilfe weiterer Softwarekomponenten des MetaMIS Toolset lassen sich die Informationsobjektmodelle der Fachkonzeptebene dann schrittweise in technische Strukturen und Beschreibungen des DWH-Systems überführen. Die in einem zentralen Metadaten-Repository gespeicherten Informationsobjektmodelle werden dazu mithilfe von Bridge-Modulen in die Metadatenbestände der eingesetzten DWH-Standardsoftware propagiert. Durch diese weitgehende Automatisierung werden Inkonsistenzen zwischen fachkonzeptioneller, logischer und technischer Ebene von vornherein vermieden.

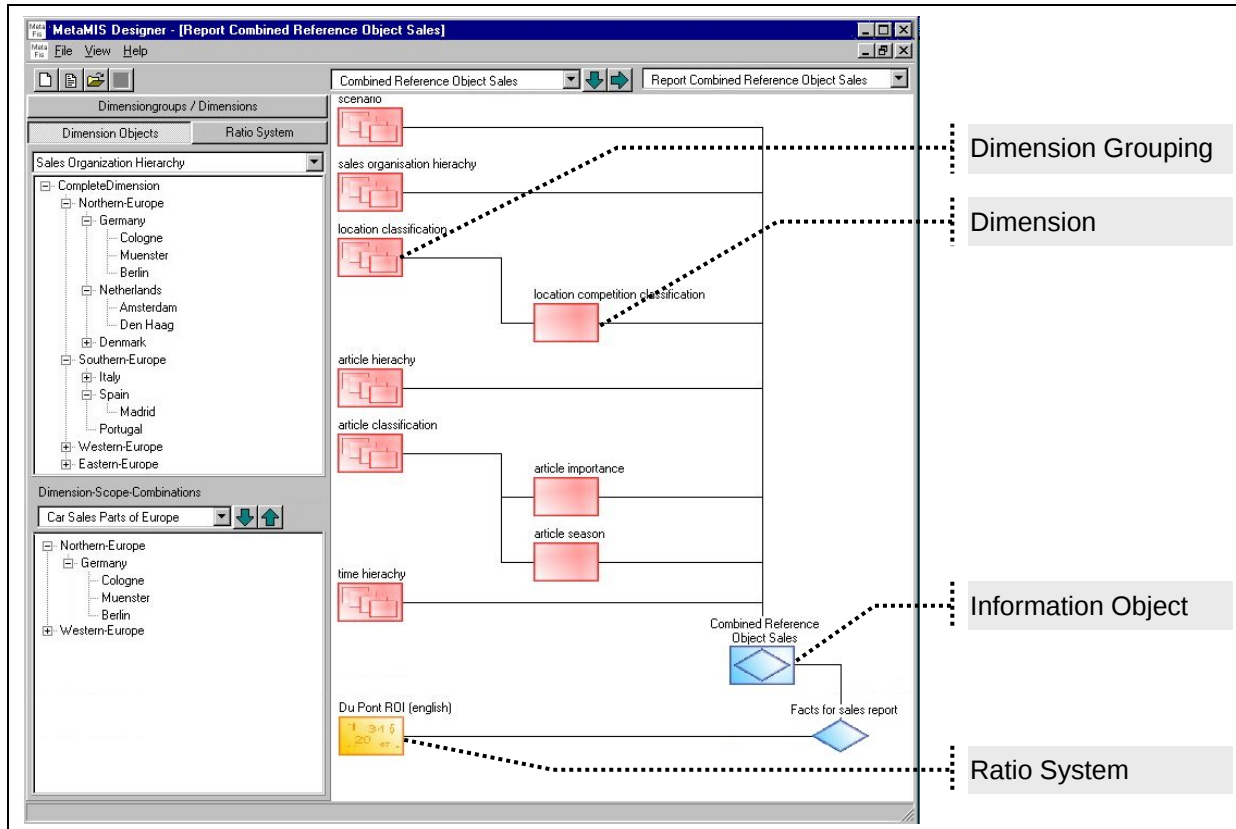


Abb. 2-2: Modellierung des Informationsbedarfs mithilfe des Werkzeugs
MetaMIS Designer [Vgl Holten 2001, S. 21]

Der mit MetaMIS verfolgte Ansatz bietet zudem den Vorteil, dass die Perspektive der Fachbereiche den Ausgangspunkt für die Entwicklung des Führungsinformationssystems bildet und dass die Anforderungen in einer fest vorgegebenen und einheitlichen Technik dokumentiert werden. Eine Limitierung des MetaMIS-Ansatzes kann darin gesehen werden, dass bisher lediglich strukturierte Metadaten durch das System verwaltet werden. Semi- oder unstrukturierte Daten bspw. zur Dokumentation von Informationsprozessen oder Regeln zur Fehlerbehandlung von ETL-Prozessen werden bisher vernachlässigt. Darüber hinaus besteht die Möglichkeit der Konsistenzsicherung zwischen den verschiedenen Entwicklungsebenen bisher nur konzeptionell, da der realisierte Prototyp noch keine Funktionalität für ein Änderungsmanagement bereitstellt.

Insgesamt kann MetaMIS als erfolgversprechendes Konzept einer soliden Methode zur unternehmensweiten Entwicklung DWH-basierter Führungsinformationssysteme gewertet werden, das zudem durch ein modulares Systemkonzept die Möglichkeit bietet, sich neuen technischen Entwicklungen in der DWH-Standardsoftware anzupassen. Für einen praktischen Ein-

satz wäre es allerdings nötig, ein Änderungsmanagement sowie Möglichkeiten zur Verwaltung auch semi- und unstrukturierter Informationen in den Ansatz zu integrieren.

2.4 Nutzenpotentiale einzelner Metadatenkategorien aus Sicht der Praxis

Im Kap. 2.2 wurden verschiedene Einsatzmöglichkeiten für Metadaten im Data Warehousing vorgestellt. Bezüglich der Ausnutzung der entsprechenden Nutzenpotentiale ist im Vorfeld der Implementierung einer integrierten Metadatenverwaltung jedoch zunächst festzulegen, welcher Aufwand für welche Komponente einer Implementierung gerechtfertigt ist bzw. welche Prioritäten im Rahmen der Realisierung zu setzen sind. Da bezüglich solcher Fragestellungen bisher nahezu keine Aussagen existieren, wurde im Rahmen des bereits angesprochenen Workshops des CC DW2 eine Expertengruppe aus Mitarbeitern der DWH-Fachabteilungen verschiedener Grossunternehmen beauftragt, Metadatatypen hinsichtlich dreier Merkmale zu bewerten, um daraus Prioritäten für eine spätere Realisierung ableiten zu können. Als Grundlage für die Bewertung wurde dabei die in Kap. 2.1 vorgestellte pragmatische Aufzählung von Metadatenkategorien [vgl. Tab. 2-1] herangezogen. Jedes Mitglied der Expertengruppe erhielt in der Folge die Aufgabe, zu jeder der Kategorien seine Einschätzung hinsichtlich der drei Faktoren Nutzen, Realisierungsaufwand und Praktikabilität der Umsetzung bewertet werden. Zur Bewertung erhielt jedes Gruppenmitglied eine feste Anzahl von positiven und negativen Bewertungspunkten, die auf die einzelnen Kategorien verteilt werden konnten. Zur Auswertung wurde für jede Metadatenkategorie die Gesamtzahl positiver und negativer Punkte pro Bewertungskriterium ermittelt. Abb. 2-3 zeigt das Endergebnis der Bewertungsrunde, wobei für jedes Bewertungskriterium jeweils nur die Differenz aus positiver und negativer Punktzahl im Verhältnis zu insgesamt ausgegebenen Punktemenge dargestellt ist.

Betrachtet man zunächst das Kriterium Aufwand, so kann festgehalten werden, dass insbesondere die Kategorien Business Nomenclature, Warehouse Content und Business Rules mit vergleichsweise sehr hohem Aufwand verbunden werden. Dies scheint ein Indiz dafür zu sein, dass gerade solche Metadaten, die nur mit einem hohen Anteil manueller Dokumentationsstätigkeiten umsetzbar sind, als aufwendig einzuschätzen sind. Der noch recht hohe Aufwand der Kategorie Warehouse Process dürfte hingegen darauf zurückzuführen sein, dass die Informationen über Data Lineage meist nur mit viel Mühe aus den genutzten ETL-

Werkzeugen extrahiert und anderen Nutzern verfügbar gemacht werden können.² Die Kategorien Datenmodelle, Software Deployment und vor allem Warehouse Operation wurden als weniger aufwendig eingestuft, was vermutlich darin begründet liegt, dass die entsprechenden Metadaten bereits in elektronischer Form und benutzerfreundlicher Repräsentationsform vorliegen (Daten- und Architekturmodelle, Dokumentation zu ETL-Prozessen).

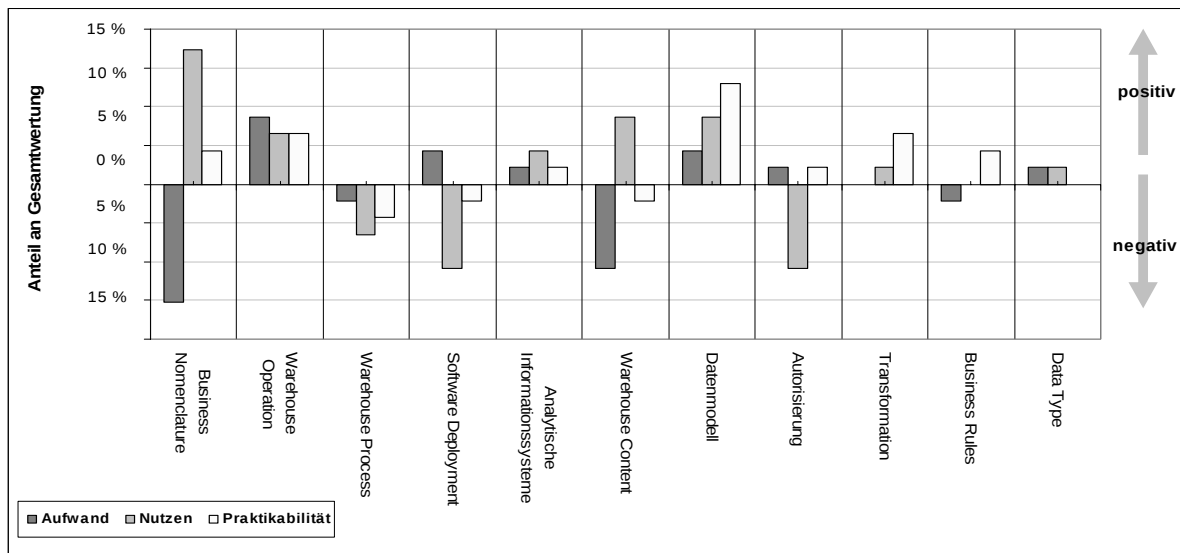


Abb. 2-3: *Einschätzung der CWM-Packages aus Sicht der Praxis*

Im Hinblick auf das Kriterium Nutzen bietet sich ein anderes Bild. Lediglich von den Kategorien Software Deployment, Autorisierung und Warehouse Process wird kein zusätzlicher Nutzen erwartet. Mit Business Nomenclature und Warehouse Content wurden gerade diejenigen Kategorien als besonders nutzenstiftend eingestuft, die auch den höchsten Aufwand zur Realisierung befürchten lassen. Dieser starke Kontrast zwischen Aufwand und Nutzen demonstriert eindrucksvoll die Problematik, der sich das Metadatenmanagement ausgesetzt sieht. Über die beiden bereits genannten hinaus wurden insbesondere die Kategorien Warehouse Operation, Datenmodell sowie Analytische Informationssysteme als nützlich eingestuft.

Die Bewertung der Praktikabilität, d. h. der Umsetzbarkeit im eigenen Unternehmen, geht grösstenteils mit der Nutzeneinschätzung konform, lediglich bei zwei Kategorien gab es eine Divergenz zwischen Nutzen und Praktikabilität. Während die Umsetzung nur in den Kategorien Warehouse Process, Warehouse Content und Software Deployment als eher unprak-

²

Eine Ausnahme dürfte hierbei der Hersteller Informatica darstellen, der diese Informationen in seinen ETL-Werkzeugen sowohl über vordefinierte Repository Views als auch über ein API bereitstellt.

tikabel angesehen wird, schätzen die Experten insbesondere die Kategorien Datenmodell, Warehouse Operation, Transformation, Business Rules und Business Nomenclature als praktikabel realisierbar ein.

Mithilfe der drei Bewertungsgrößen lässt sich abschliessend zumindest eine ungefähre Empfehlung für die Reihenfolge angeben, in der die einzelnen Kategorien in ein Metadatenmanagement integriert werden sollten. Der Praktikabilität messen die Autoren dabei eine erhöhte Bedeutung bei, da nur dann, wenn die Implementierung durchsetzbar erscheint, sich auf absehbare Zeit auch ein Erfolg erreichen lassen wird. Auf Basis dieser Strategie lassen sich *quick wins* erzielen, die dem MDM-Projekt zu mehr Aufmerksamkeit verhelfen und das weitere Vorgehen positiv beeinflussen. Dementsprechend wird für die Bestimmung einer Umsetzungsreihenfolge das Kriterium Praktikabilität höher bewertet als die übrigen Kriterien Aufwand und Nutzen. Tab. 2-3 zeigt das Ergebnis der Priorisierung. Demnach scheint es am ehesten empfehlenswert, bei der Realisierung eines integrierten Metadatenmanagements zunächst mit der Implementierung für die Metadatenkategorie Datenmodell zu beginnen. Im Anschluss sollten vornehmlich ETL-Metadaten übernommen werden, bevor mit eher fachlichen Metadaten fortgefahren wird. Die Kategorie Business Nomenclature mit dem höchsten Einzelnutzen sollte erst dann integriert werden, wenn bereits ein erstes funktionierendes Basissystem für das MDM etabliert ist.

Rang	Metadatenkategorie	Kriterium:	Praktikabilität	Aufwand	Nutzen
		Gewicht:	0,4	0,3	0,3
		Rangwert			
1	Datenmodell	0,0365	5,22%	1,74%	3,48%
2	Warehouse Operation	0,0287	2,61%	3,48%	2,61%
3	Transformation	0,0130	2,61%	0,00%	0,87%
4	Analytische Informationssysteme	0,0113	0,87%	0,87%	1,74%
5	Business Nomenclature	0,0096	1,74%	-6,09%	6,96%
6	Data Type	0,0052	0,00%	0,87%	0,87%
7	Business Rules	0,0043	1,74%	-0,87%	0,00%
8	Warehouse Content	- 0,0061	-0,87%	-4,35%	3,48%
9	Autorisierung	- 0,0070	0,87%	0,87%	-4,35%
10	Software Deployment	- 0,0113	-0,87%	1,74%	-4,35%
11	Warehouse Process	- 0,0174	-1,74%	-0,87%	-2,61%

Tab. 2-3: Priorisierung von Metadaten-Kategorien

3 Technische Metadatenintegration im Data Warehousing

3.1 Metadatenmanagementsysteme

Um das aus Metadaten erwachsende Nutzenpotential für das Data Warehousing vollständig nutzen zu können, bedarf es einer werkzeugübergreifenden Verwaltung all solcher Metadaten, die für mehrere Parteien von Bedeutung sind. Nur auf diese Weise kann sichergestellt werden, dass die bereits verfügbaren Zusatzinformationen hinsichtlich der Beschreibung der DWH-Daten, der sie bearbeitenden Prozesse sowie der involvierten Personen sämtlichen am Data Warehousing beteiligten Parteien zur Verfügung stehen. Im Rahmen eines Metadatenmanagements (MDM) müssen Metadaten daher gezielt generiert, verwaltet und bereitgestellt werden. Metadatenmanagement als die Gesamtheit aller Prozesse, die für die Entwicklung, die Organisation, den Betrieb und die Nutzung des Metadatensystems erforderlich sind, wird unterstützt durch ein Metadatenmanagementsystem (MDMS), welches aus den Applikationen und Datenhaltungssystemen besteht, die zur Erfassung, Übertragung, Speicherung und Bereitstellung der Metadaten benötigt werden. Als primäre Datenquellen stellen dabei die Softwarewerkzeuge des Data Warehouse den Grossteil der Komponenten des MDMS. Eine wesentliche Voraussetzung für die Implementierung des MDMS ist die Entscheidung für eine Architektur, die die einzelnen Software-Komponenten zu einem Gesamtsystem zusammenführt. Im Folgenden werden zunächst die grundlegenden Varianten einer MDMS-Architektur vorgestellt. Anschliessend wird basierend auf den Erfahrungen aus einem realen Showcase die heute am weitesten verbreitete Architekturvariante vorgestellt und hinsichtlich ihrer Vor- und Nachteile beurteilt.

3.2 Architekturvarianten für Metadatenmanagementsysteme

Im Mittelpunkt der Architektur eines Metadatenmanagementsystems steht in der Regel ein Repository, in dem Metadaten aus unterschiedlichen Quellen integriert werden [vgl. z. B. Müller/Stöhr/Rahm 1999, S. 1; Tozer 1999; Marco 2000; Jennings 2001]. Ein Repository ist eine spezielle Datenbankanwendung, die „Informationen über Objekte der Softwareproduktion (z. B. Programme, Datenfelder, Masken, Listen), deren Beschreibungen und Beziehungen untereinander verwaltet, auswertet und bereitstellt“ [Habermann/Leymann 1993, S. 15]. Ein Metadaten-Repository im Sinne dieses Beitrags ist speziell zur Verwaltung und Auswertung von Metadaten für das DWH-System bestimmt. Es ist logisch unabhängig von der Data-Warehouse-Datenbank, kann aber physikalisch dieselbe DBMS-Plattform nutzen [Vadu-

va/Dittrich 2001, S. 4]. Teilweise werden in der Literatur auch Modellierungskomponenten für die Repository-Elemente zu den Bestandteilen eines Repositories gezählt [Ortner 1999, S. 236].

Repositories können eigenständige Applikationen oder Bestandteile von Applikationen sein. Eigenständige Repositories, wie z. B. die erwähnten Entwicklungsrepositories, dienen zur Dokumentation und zur Verwaltung wiederverwendbarer Komponenten. Im Data-Warehousing-Umfeld werden hingegen häufig sog. integrierte Repositories verwendet [vgl. Seiner 1999]. Dabei wird das Repository als Metadatenpeicher eines spezifischen Werkzeugs verwendet und ist nicht oder nur eingeschränkt für die Nutzung durch Entwickler oder Endbenutzer vorgesehen. Den integrierten Repositories fehlen sowohl Modellierungskomponenten als auch Möglichkeiten, unter Umgehung des vorgelagerten Werkzeugs auf die im Repository gespeicherten Daten zuzugreifen.

Weiterhin lassen sich aktive und passive Repositories unterscheiden. Ein aktives Repository steht zur Laufzeit der Anwendungssoftware zu Verfügung, um andere Laufzeit-Komponenten mit Informationen zu versorgen. Zum Beispiel ist ein Metadaten-Repository bezogen auf ein ETL-Tool aktiv, wenn das ETL-Tool zur Laufzeit auf die Metadaten im Repository zugreift und der ETL-Prozess ansonsten nicht ausgeführt werden könnte. Ein passives Repository dient als reines Dokumentationsarchiv zur passiven Bereitstellung von Metadaten an menschliche Nutzer wie z. B. Entwickler oder über geeignete Schnittstellen auch Endanwender.

Do und Rahm unterscheiden drei unterschiedliche Architektur-Ansätze für die Implementierung eines Metadaten-Repositories im Rahmen eines integrierten Metadatenmanagementsystems [vgl. Do/Rahm 2000, S. 8 ff.], die im Folgenden kurz beschrieben werden.

3.2.1 Zentrale Architektur

In einer zentralen Architektur werden Metadaten ausschliesslich in einem einzigen zentralen Repository verwaltet. Das zentrale Repository speichert sowohl Metadaten, die für einzelne Komponenten spezifisch sind und von der übrigen DWH-Umgebung nicht benötigt werden (z. B. spezifische Datenstrukturen eines Reporting-Tools) als auch Metadaten, die von mehreren Komponenten gemeinsam genutzt werden (z. B. fachliche Begriffsdefinitionen). Die DWH-Komponenten verwalten keinerlei Metadaten lokal, sondern greifen direkt auf das globale Metadaten-Repository zu.

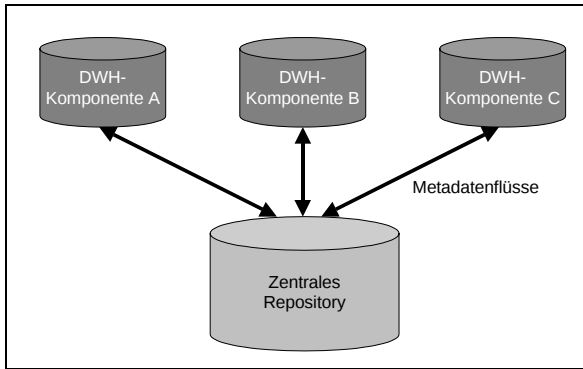


Abb. 3-1: Zentrale Architektur für ein Metadaten-Repository
[vgl. Do/Rahm 2000, S. 8]

Wichtigster Vorteil einer solchen Lösung ist die konsistente Verwaltung der Metadaten an einem Ort. Komplexe und fehleranfällige Replizierungsmechanismen für den Abgleich mehrerer Repositories sind nicht erforderlich. Zudem benötigt jede angeschlossene DWH-Komponente lediglich Zugriff auf die zentrale Datenbank, um an sämtliche im DWH verwalteten Metadaten zu gelangen. Dies reduziert die Anzahl benötigter Schnittstellen und minimiert den Kommunikationsbedarf für Metadatenaustausch.

In der Praxis wird beim Aufbau von Data-Warehouse-Systemen i. d. R. auf kommerziell verfügbare Softwareprodukte zurückgegriffen, wodurch im Vergleich zu einer Eigenentwicklung eine deutliche Kostenersparnis erzielt wird. Obwohl eine Fülle von Herstellern Produkte für den Aufbau von DWH-Systemen anbieten, können nur wenige dieser Hersteller sämtliche Bereiche der DWH-Architektur mit ihrer Produktpalette abdecken [Buytendijk/Friedman 2001]. Aus diesem Grund und um die oft beträchtlichen Investitionen in ein DWH-System besser zu schützen (keine Abhängigkeit von einem einzigen Hersteller), bauen die meisten Unternehmen ihre spezifischen Data-Warehouse-Lösungen aus ausgewählten Produkten unterschiedlicher Hersteller auf (sog. Best-of-breed-Ansatz). In einer solchen, heterogenen DWH-Architektur gestaltet sich die Implementierung eines zentralen Metadaten-Repositories in Reinform jedoch zum momentanen Zeitpunkt als nahezu unmöglich. Insbesondere die vollständige Auslagerung von produktspezifischen Metadaten in ein zentrales Repository lässt sich nicht realisieren. Hierfür gibt es mehrere Gründe, die in engem Zusammenhang stehen. Zum einen verfügen die Repositories heute am Markt erhältlicher DWH-Softwareprodukte meist über proprietäre Datenmodelle und -strukturen, die mit denen anderer Repositories nicht kompatibel sind. In vielen Fällen sind darüber hinaus Dokumentationen dieser Datenmodelle und -strukturen nicht öffentlich zugänglich, sodass eine Schemaintegration mehrerer Repositories nahezu unmöglich scheint. Zum anderen sind bei den meisten kommerziel-

len DWH-Werkzeugen Daten- und Funktionsteile nicht hinreichend entkoppelt, sodass es daher i. d. R. nicht möglich ist, die Metadaten in einem externen Repository zu verwalten. Selbst falls diese Hindernisse nicht vorliegen, so scheitert ein rein zentraler Ansatz meist daran, dass die von den betreffenden DWH-Werkzeugen zur Verfügung gestellten Metadaten-schnittstellen nur einige wenige proprietäre Formate unterstützen, die eine wirklich freie Wahl der zu integrierenden Komponenten unmöglich macht. Universelle und herstellerunabhängige Formate für den Metadatenaustausch sind mittlerweile zwar verfügbar, werden allerdings bis dato nur in Ausnahmefällen auch von kommerziellen Tools genutzt und erlauben immer nur den Austausch einer Teilmenge des tatsächlich benötigten Metadatenbestandes. Ein Beispiel für die optionale Nutzung einer solchen Standardschnittstelle zeigt der Hersteller Oracle in der Version 9i des ETL-Werkzeugs Warehouse Builder, der den Im- und Export gewisser Metadaten über XML-Dateien ermöglicht, welche auf dem Common Warehouse Metamodel (CWM) Standard der OMG (s. auch Kap. 4) basieren.

Neben rein technischen Problemen warnt Blechar [Blechar 2001] zudem auch vor politischen und kulturellen Problemen, z. B. hinsichtlich der Festlegung der Verantwortlichkeit für die Pflege der Metadaten (Metadata ownership) sowie der Verteilung der Kosten für Entwicklung und Betrieb.

3.2.2 Dezentrale Architektur

Do und Rahm sprechen von einer dezentralen Architektur, wenn jede DWH-Komponente seine eigenen Metadaten in einem lokalen Repository verwaltet und über bilaterale Schnittstellen Metadaten anderer DWH-Komponenten direkt von diesen bezieht [vgl. Do/Rahm 2000, S. 8].

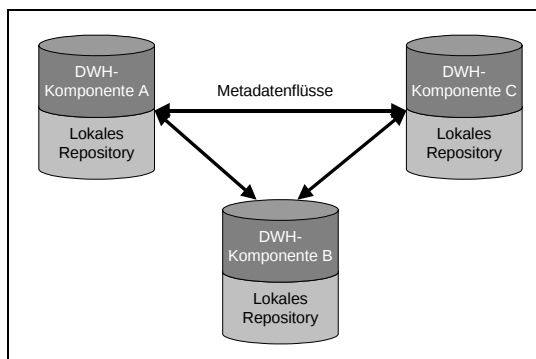


Abb. 3-2: *Dezentrale Architektur für ein Metadaten-Repository*
[vgl. Do/Rahm 2000, S. 8]

Bei dieser Architekturvariante entfallen einige Nachteile der rein zentralen Architektur, da erstens kein zentrales Repository installiert werden muss und zweitens auch solche Softwarewerkzeuge einsetzbar sind, die nur mit einem lokalen Repository problemlos betrieben werden können. Noch schwerer als in der zentralen Variante wirkt in einer solchen Architektur die Schnittstellenproblematik, da hier nunmehr jede DWH-Komponente Kommunikationsbeziehungen mit jeder anderen Komponente unterhalten muss, mit der ein Metadaten austausch stattfinden soll. Zwar verfügen i. d. R. auch Werkzeuge mit nur lokal installierbarem Repository über zusätzliche Schnittstellen für den Austausch von Metadaten, jedoch funktioniert dieser Austausch meist nur zwischen einigen wenigen Werkzeugen und z. T. auch nur unidirektional. Eine der wenigen Ausnahmen bietet hier der Hersteller Informatica, der für seine ETL-Produkte PowerCenter und PowerMart neben speziellen Metadaten-Bridges für den Metadaten austausch mit Produkten einiger anderer Hersteller auch ein API anbietet, mit dem auf das Informatica-Repository sowohl lesend als auch schreibend zugegriffen werden kann. Hierüber lässt sich ein kundenspezifischer Metadaten-Austausch realisieren, falls auch die Metadatenbestände der übrigen eingesetzten DWH-Komponenten in ähnlicher Weise zugänglich sind.

Über die Austauschproblematik hinaus birgt diese Architekturvariante den Nachteil in sich, dass keine zentrale Instanz die Abwicklung des Metadaten austausches übernimmt. Vielmehr erlaubt der Grossteil der aktuellen DWH-Werkzeuge lediglich einen simplen dateibasierten Metadaten-Export, wobei dieser häufig nur manuell und nur aus dem jeweiligen Werkzeug selbst heraus angestoßen werden kann. Die Realisierung eines funktionierenden, regelmässigen Metadaten austausches in dieser dezentralen Variante lässt sich daher ohne organisatorische Regelungen kaum bewerkstelligen. Obwohl sich ein funktionierendes Metadatenmanagement unter solchen Voraussetzungen eher selten etablieren dürfte, ist die dezentrale Variante heutzutage sicher am weitesten verbreitet, da sie auf bereits vorhandener Funktionalität basiert und mit geringem Zusatzaufwand umsetzbar ist.

3.2.3 Hybride Architektur – Metadata Warehouse

Die in der momentanen Situation aussichtsreichste Variante zur technischen Realisierung einer integrierten Metadatenverwaltung besteht in einer hybriden Architektur, bei der die Vorteile der beiden Extreme einer rein zentralen und einer rein dezentralen Metadatenhaltung kombiniert werden können. Jede DWH-Komponente verwaltet dabei ihre eigenen Metadaten zunächst in einem lokalen Repository. Zusätzlich überstellt es eine Kopie der eigenen Metadaten in ein zentrales Repository, welches im Gegensatz zum Repository des rein zentralen

Ansatzes lediglich als passive Informationsbasis genutzt wird, über die sich jede Komponente mit den Metadaten anderer Werkzeuge versorgen kann.

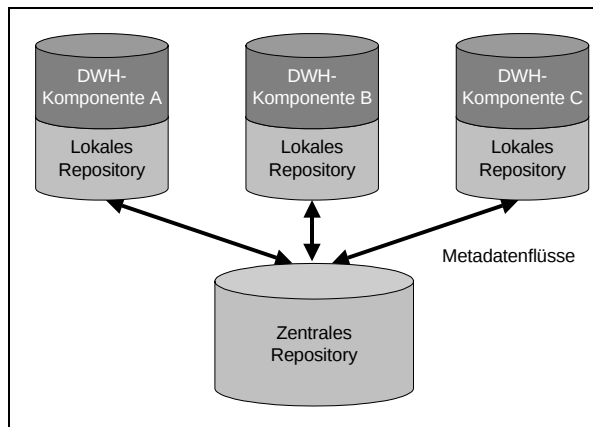


Abb. 3-3: *Hybride Architektur für ein Metadaten-Repository*
[vgl. Do/Rahm 2000, S. 8]

Auf diese Weise können alle DWH-Komponenten ihre lokalen Repositories unabhängig verwalten und trotzdem von den Vorteilen einer zentralen Metadatenquelle profitieren. Allerdings ist auch diese Lösung nicht frei von Hindernissen, die es zu überwinden gilt. Wie auch bei allen anderen Varianten kann der Metadatenaustausch nur über existierende Schnittstellen der einzelnen Werkzeuge funktionieren. Darüber hinaus bedarf es eines Replikationsmechanismus, mit dessen Hilfe die zentralen Kopien der lokalen Metadaten regelmässig aktualisiert werden. Bei der Implementierung eines solchen Mechanismus bestehen allerdings mehrere Freiheitsgrade. So kann die Bereitstellung der Metadaten im zentralen Repository einerseits durch Lieferung seitens der einzelnen Komponenten oder aber andererseits durch Abholung seitens eines Repositorydienstes erfolgen. Bezüglich der zentralen Speicherung der Metadaten besteht die Möglichkeit, ein gemeinsames Format zu etablieren, in das die lokalen Metadaten überführt werden oder aber die lokalen Daten einfach nicht integriert abzulegen und lediglich eine integrierte Sicht auf das gemeinsame Repository zu etbalieren. Je nach Ausgestaltung des MDM-Systems kann der Replikationsmechanismus zu einem tatsächlichen Extraktions-, Transformations- und Ladeprozess ausgebaut werden, wie er auch im Data Warehouse selbst verwendet wird. Diese umfassendste Architekturvariante ist in Abb. 3-4 am Beispiel einer idealtypischen DWH-Architektur detailliert dargestellt. Die auf den einzelnen Schichten angeordneten DWH-Komponenten (Werkzeuge, Datenbanken, Schnittstellen etc.) produzieren und konsumieren während Entwicklung und Betrieb fortwährend Metadaten (z. B. Datenstrukturen, ETL Mappings oder Felddesreibungen). Typischerweise werden diese Metadaten von den zugehörigen Komponenten lokal verwaltet, ohne dass andere Kom-

ponenten direkten Zugriff auf diese Metadaten haben (z. B. DBMS Catalog mit Schemadefinitionen der DWH-Datenbank). In der konzeptionellen Architektur sind diese lokalen Metadatenbestände in der Schicht „Local Metadata“ zusammengefasst.

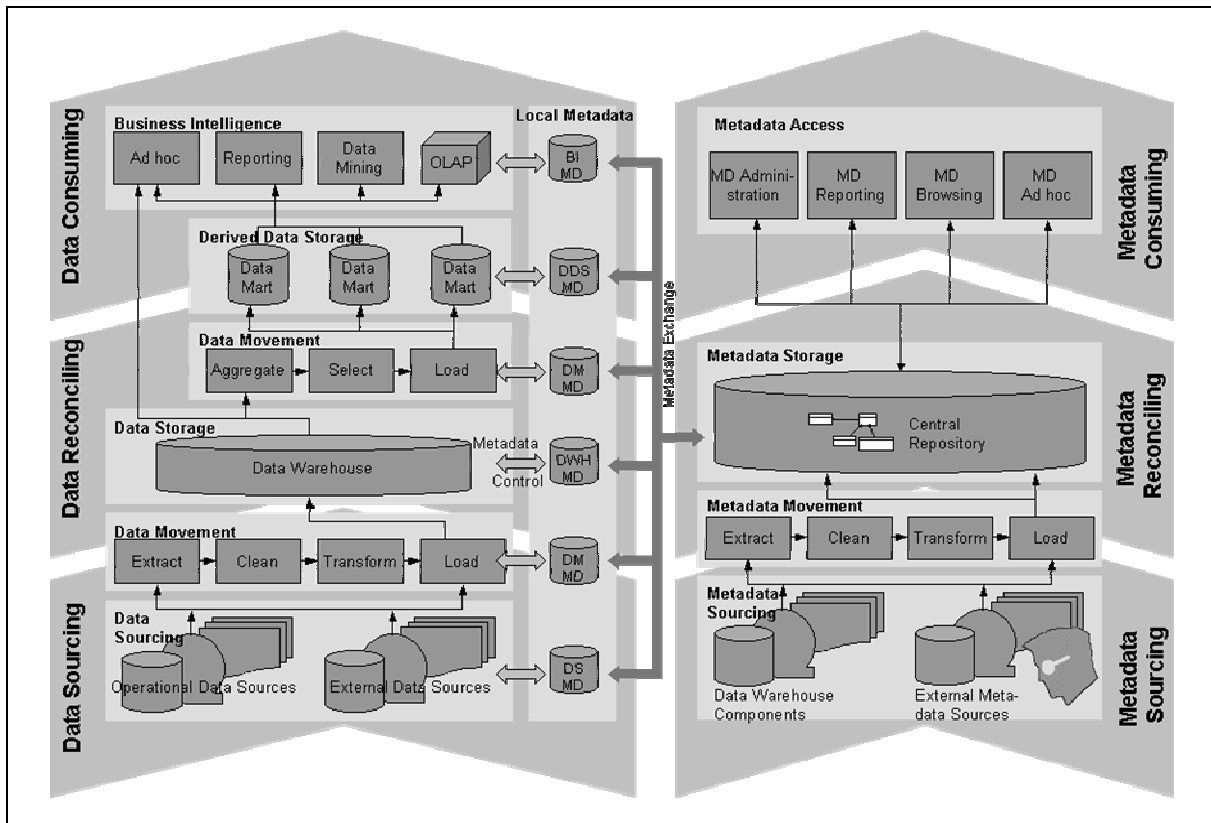


Abb. 3-4: Architektur eines DWH mit integriertem Metadata Warehouse

In Anlehnung an die Zielsetzung des Data Warehousings kann als Ziel des Metadatenmanagements die Errichtung einer Informationslogistik für die Versorgung des Data Warehousing-Prozesses mit Metadaten formuliert werden. Neben den lokalen Metadaten des DWH-Systems existieren eine Fülle von Metadaten, die z. B. in Form unstrukturierter Dokumente (z. B. Benutzerhandbücher) oder gar nur als implizites Wissen der am Data Warehousing beteiligten Personen [Devlin 1997, S. 57] vorliegen und für eine Verarbeitung im MDMS zunächst strukturiert und codiert werden müssen. Die Integration von externen Metadaten mit den Metadaten der „Data Warehouse Components“ wird über das zentrale Metadaten-Repository realisiert, welches darüber hinaus eine werkzeugunabhängige Versionierung und Konfigurationsbildung der Metadaten ermöglicht. In seiner wichtigsten Funktion dient das Repository den DWH-Komponenten als Drehscheibe zum Austausch von lokalen Metadaten, welche hierzu über einen standardisierten Austauschmechanismus in das zentrale Repository geladen und in einem speziellen, auf einem gemeinsamen Metamodell basierenden

Format abgelegt werden. DWH-Komponenten, die diesen Mechanismus nutzen wollen, müssen einen sog. Adapter implementieren, der die interne Repräsentation der Metadaten auf das Standardformat abbildet und damit externalisiert.

Die Architekturkomponenten zur Nutzung der Metadaten orientieren sich an den grundlegenden Nutzungsaspekten *Unterstützung von Entwicklung und Betrieb des DWH-Systems* sowie *Unterstützung der Nutzung des DWH-Systems*. Entwicklung und Betrieb profitieren primär durch aktive Metadaten-Nutzung der DWH-Komponenten (vgl. Abschnitt 2.2). Für die Endbenutzer werden analytische Applikationen bereitgestellt, die eine Auswertung des Metadaten-Repositories erlauben. Devlin [Devlin 1997, S. 275] schlägt bspw. eine Applikation *Business Information Guide (BIG)* vor, welche den Benutzer beim Auffinden relevanter Objektdaten im DWH-System und ihrer Interpretation unterstützt. Der Zweck des BIG ist die Einordnung von Objektdaten in ihren Business Kontext. Die Applikation *Data Warehouse Catalog (DWC)* wendet sich in erster Linie an technische Benutzer und stellt ihnen eine integrierte Dokumentation des DWH-Systems aus technischer Sicht zur Verfügung. Andere denkbare Metadaten-Applikationen unterstützen das Management von Begriffssystemen oder die Abfrage von Qualitätsinformationen.

3.3 Technische Metadatenintegration am Beispiel einer dezentralen Architektur

Während Metadatenmanagement auf Basis eines Shared Repository ein sehr großes Nutzenpotential bietet [Do, Rahm 2000, S. 10] verursacht es gleichzeitig einen nicht zu unterschätzenden Entwicklungsaufwand – sowohl unter Zeit- als auch unter Kostenaspekten. Eine wesentlich günstigere und schneller zu realisierende Alternative besteht in der Nutzung bilateraler Metadaten-Bridges, die in den meisten DWH-Standardsoftwareprodukten bereits enthalten sind. Zur realistischen Einschätzung der Stärken und Schwächen dieser Metadatenmanagement-Variante wurde im Rahmen eines Workshops des CC DW2 ein DWH-Showcase-System mit entsprechenden Metadaten austauschverbindungen installiert und von mehreren Praktikern auf seine Tauglichkeit hin untersucht.

Abb. 3-5 zeigt die Architektur des Showcase-Systems, das auf einem Oracle8i DBMS basiert. Das Datenmodellierungswerkzeug Erwin 4 wird eingesetzt, um die konzeptionellen und logischen Datenmodelle der DWH-DB zu erstellen und die entsprechenden DB-Tabellen daraus zu generieren. Mithilfe der ETL-Software Informatica PowerCenter 5.1 werden Daten aus OLTP-Datenbanktabellen extrahiert, transformiert und in die DWH-DB geladen. Die Auswertung der DWH-Daten erfolgt durch die Business-Intelligence-Suite BrioQuery 6.5. Jede der einge-

setzten Software-Komponenten verwaltet eigene Metadaten, wobei teilweise relationale Tabellen und teilweise Flatfiles proprietären Formats als Datenspeicher dienen. Auf Basis bilateraler Schnittstellen gelingt es in dieser Umgebung, Metadaten der Information-Modelling- und der Data-Movement-Schicht in der Business-Intelligence-Schicht der Data-Warehousing-Architektur bereitzustellen. Hierzu wird zunächst das Metadaten-Bridge Informatica PowerPlug 5.1 eingesetzt, um Datenmodellierungs-Metadaten in das Repository von Informatica PowerCenter zu überführen. Hier zeigt sich bereits eine Beschränkung, da die transportierbaren Metadaten durch das Programm nahezu fest vorgegeben werden und sich i. W. auf Namen und Beschreibungen von Fachbegriffen zu Relationen der logischen Ebene beschränken. Darüber hinaus kann der Transport der Metadaten nur manuell ausgelöst werden, da das Programm weder über Schedulingfunktionalität verfügt, noch ein für den Transport verantwortlicher Systemdienst installiert werden kann.

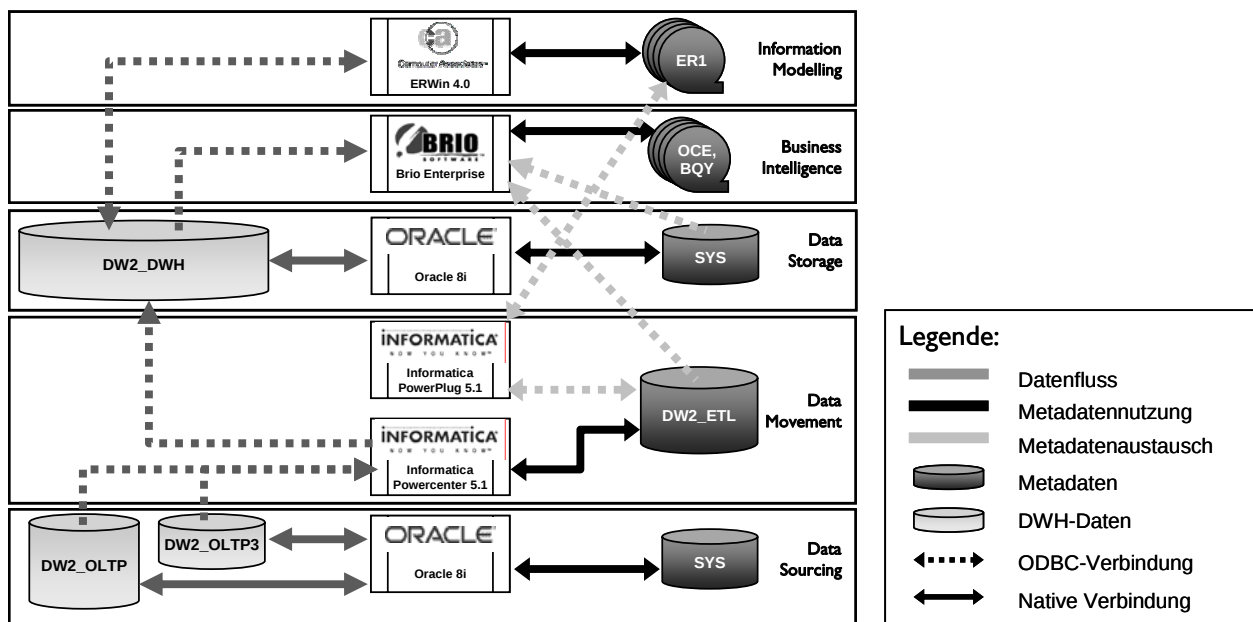


Abb. 3-5: Architektur des Showcase-Systems für Metadaten austausch

Wesentlich flexibler gestaltet sich die Schnittstelle zwischen den Schichten Data Movement und Business Intelligence. Die eingesetzte Auswertungssoftware BrioQuery 6.5 erlaubt neben dem Zugriff auf das DWH auch eine Verbindung zu einer beliebigen relationalen Metadatenquelle. Hierdurch wird es möglich, nahezu sämtliche Metadaten des ETL-Repositories in der Business-Intelligence-Schicht verfügbar zu machen. Neben bereits vorgefertigten Metadatenberichten können mittels SQL eigene Abfragen definiert werden. Nachteil dieser Lösung ist die fehlende Möglichkeit, auch Metadaten der BI-Schicht in die Repositories der übrigen Schichten zu transportieren. Darüber hinaus stehen die Metadaten lediglich bei der Entwick-

lung von Auswertungen zur Verfügung, die Anzeige von Metadaten zu einem konkreten Auswertungsfakt sieht die Software nicht vor.

Der Showcase konnte aufzeigen, dass eine grundlegende Form des Metadaten austausches zwischen Standardsoftwarekomponenten zwar möglich ist, in den Punkten Praktikabilität und Umfang der integrierbaren Metadaten jedoch z. T. erhebliche Restriktionen bestehen. Zum ersten beschränkt sich auch dieser Ansatz auf den Austausch strukturierter Metadaten, wohingegen keine Möglichkeiten bestehen, auch unstrukturierte Metadaten mithilfe der eingesetzten Software integriert zu verwalten. Zum zweiten wurde die Flexibilität der Austauschbeziehungen allgemein als zu niedrig eingeschätzt, was sich sowohl auf die austauschbaren Metadatenkategorien als auch auf die Steuerung des Austauschprozesses bezieht.

Eine Möglichkeit der Verbesserung wurde in der Standardisierung der Metadatenchnittstellen gesehen, da hierdurch der Austausch einer grösseren Menge von Metadaten realisiert werden kann.

Im folgenden Kapitel werden zunächst die Vorteile der Standardisierung noch einmal näher beleuchtet. Anschliessend wird das Common Warehouse Metamodel der Object Management Group vorgestellt, das als weltweiter Standard für den Austausch von DWH-Metadaten etabliert werden soll.

4 Standardisierung als Basis für die Metadatenintegration

4.1 Der Nutzen von Standards für die Integration von Metadaten

Um das Nutzenpotenzial von Metadaten im Data Warehousing möglichst weitgehend ausschöpfen zu können, ist eine werkzeugübergreifende Verwendung der Metadaten unumgänglich. Dabei kommt es zum einen darauf an, den Metadatenaustausch technisch zu realisieren, zum anderen muss jedoch ebenfalls dafür gesorgt werden, dass Metadaten verschiedener Herkunft inhaltlich kompatibel sind. Während sich der technische Metadatenaustausch zwischen zwei Software-Komponenten meist über APIs oder Datenbankverbindungen relativ leicht und umfassend umsetzen lässt, stellt sich die inhaltliche Integration meist als eine ungleich schwierigere Aufgabe dar. Der Grund hierfür liegt in der Tatsache, dass die Metadatenmodelle einzelner Werkzeuge meist auf die möglichst optimale Unterstützung spezifischer Aufgaben hin konzipiert wurden. In Verbindung mit der Verwendung unterschiedlicher Modellierungstechniken hat dies zu einer Vielzahl sehr heterogener, toolspezifischer Metamodelle geführt. Möchte man die Metadaten zweier Tools inhaltlich korrekt austauschen, so bedarf es zunächst einer Zuordnung der entsprechenden Modellelemente auf Metamodellebene. Auf diese Weise wären zur Integration der Metadaten von n Werkzeugen $n(n-1)/2$ Metamodellzuordnungen nötig.

Eine Möglichkeit, die Komplexität der Integrationsaufgabe zu verringern, besteht in der Definition einer Standardschnittstelle für Metadatenmodelle. Auf diese Weise müsste jedes Metamodell einmal an das Standard-Metamodell angepasst werden und könnte dann mit jedem anderen an den Standard angepassten Werkzeug Metadaten austauschen. Bei n an das Standard-Metamodell angepassten, werkzeugspezifischen Metamodellen reduziert sich somit der Zuordnungsaufwand auf n Zuordnungsoperationen. Für den Metadatenaustausch müsste die Metamodell-Zuordnung dann in Form einer Software-Schnittstelle implementiert werden, die sowohl den Export von Metadaten in das dem Standard-Metamodell entsprechende Standardformat als auch den Import von Standard-konformen Metadaten in das werkzeugspezifische Format erlaubt.³ Die Nutzung des Standard-Formats würde also bei n zu integrierenden Metadaten Repositories genau $2n$ Schnittstellen erfordern, während ohne das Standard-Format im

³

Die Verwendung des Standard-Metamodells als Modell für die Metadaten Repositories der einzelnen Werkzeuge würde den Aufwand zwar reduzieren, scheint jedoch aufgrund der Eigenheiten der meisten Tools unrealistisch.

schlechtesten Fall $n(n-1)$ Schnittstellen erforderlich wären. In Abb. 4-1 wird der geschilderte Sachverhalt grafisch verdeutlicht.

Für die technische Realisierung ist die Vorteilhaftigkeit der standardbasierten Lösung zunächst unabhängig davon, ob der Metadaten austausch bilateral oder unter Nutzung eines gemeinsamen Metadaten Repository erfolgt.

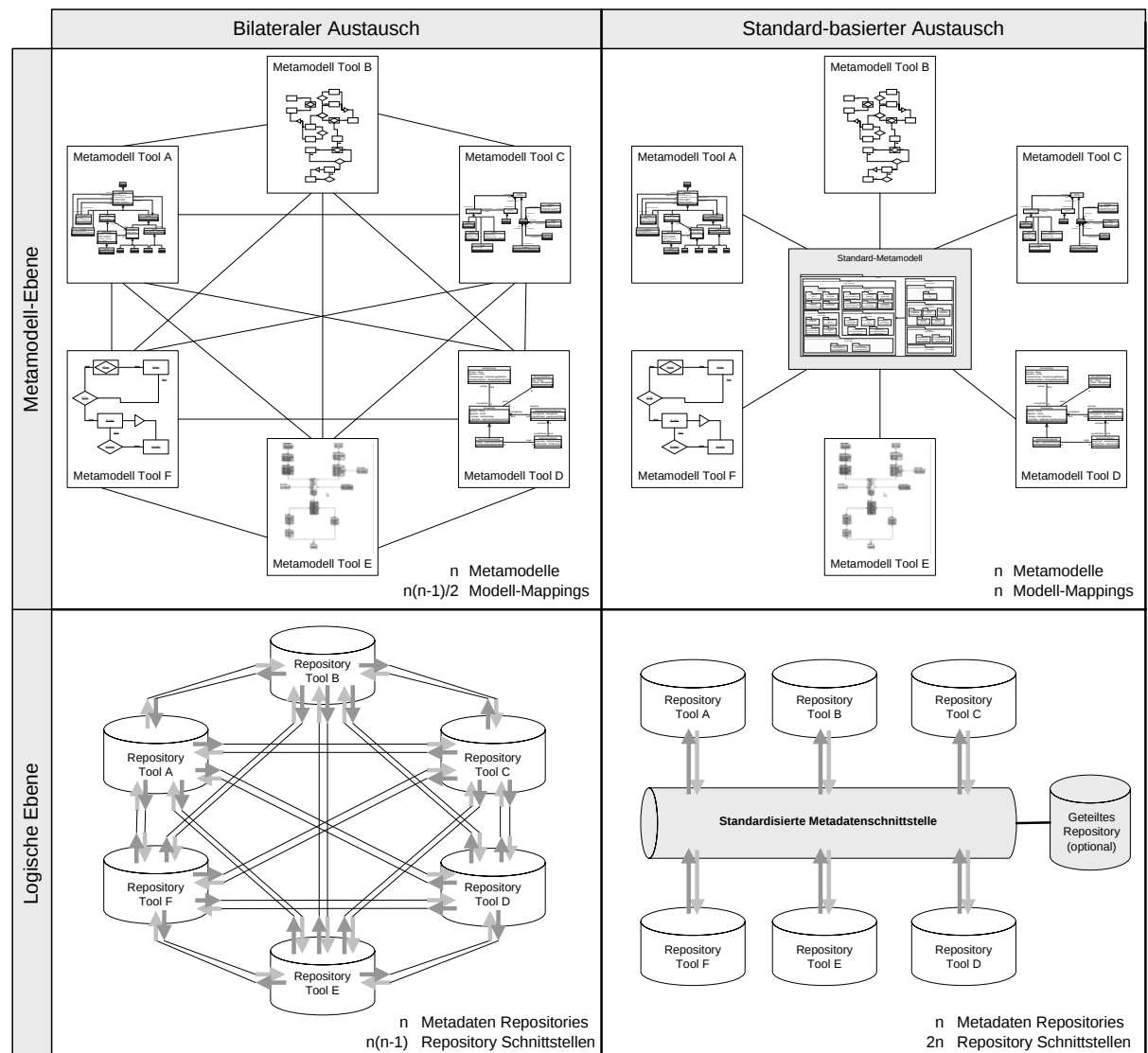


Abb. 4-1: Vergleich der Schnittstellenanzahl zwischen bilateralem und standardbasiertem Metadaten austausch

4.2 Das Common Warehouse Metamodel als Standard für DWH-Metadaten

4.2.1 Einführung in das CWM-Konzept

Um die Kommunikation verschiedener Softwarekomponenten innerhalb einer Data-Warehouse-Architektur zu verbessern, wurde von einer Arbeitsgruppe der Object Management Group das sog. Common Warehouse Metamodel (CWM) entwickelt, welches einen Standard für die Metadaten im Data-Warehousing-Umfeld definiert. Die CWM-Entwicklergruppe hat sich dabei die folgenden Ziele gesetzt:

- Das CWM soll allen Bereichen des Data Warehousing eine gemeinsame Sprache zur Verfügung stellen, mit deren Hilfe Metadaten syntaktisch und semantisch korrekt ausgetauscht werden können.
- Um der häufig zu beobachtenden Heterogenität von Data-Warehouse-Systemen gerecht werden zu können, soll der zu entwickelnde Standard möglichst architektur- und plattformunabhängig anwendbar sein.
- Da das CWM als Standard immer nur einen kleinsten gemeinsamen Nenner der darüber miteinander verbundenen Einzelmodelle darstellen kann, soll es zumindest so gestaltet sein, dass es einfach erweitert und mit anderen Standards in Einklang gebracht werden kann.
- Um die Akzeptanz des Standards sicherzustellen, soll er einerseits frei verfügbar und andererseits offen sein, d. h. jeder sollte das Recht erhalten können, den Standard mitzugestalten.

Auf Basis dieser Maxime wurde 1999 unter aktiver Mitwirkung der Unternehmen IBM, Unisys, NCR, Hyperion Solutions, Oracle, UBS, Genesis Development und Dimension EDI eine erste Version des Common Warehouse Model erarbeitet, das mittlerweile in einer überarbeiteten Version 1.1 verfügbar ist. Ein Schwerpunkt der Entwicklung lag auf der Verwendung etablierter und standardisierter OMG-Basistechnologien, um auf diese Weise die Plattformunabhängigkeit zu gewährleisten sowie die Erweiterbarkeit des Metamodells und die Vereinbarkeit mit anderen Standards zu fördern. Als Modellierungssprache wird ein Subset der Unified Modeling Language (UML) verwendet, welches i. W. die Erstellung von Klassendiagrammen⁴ und deren Zusammenfassung in Packages erlaubt. Grundlage für die Erstellung

⁴ Die Definition der CWM-Klassendiagramme enthält dabei einige kleinere Vereinfachungen gegenüber der originalen UML-Definition. Vgl. [Tolbert (2001), S. 15f].

von Modellen bildet die sog. Meta Object Facility (MOF), ein vierstufiges Metamodell, welches gewährleisten soll, dass das CWM mit anderen auf der MOF basierenden Metamodellen kompatibel ist. Das CWM stellt eine Instanz der MOF dar und kann über in der MOF definierte Schnittstellen mit anderen MOF-konformen Metamodellen wie beispielsweise dem Metamodell der UML verknüpft werden. Für den Austausch der den CWM-Konventionen entsprechenden Metadaten wird zusätzlich auf den OMG-Standard XML Metadata Interchange (XMI) als Basistechnologie zurückgegriffen. XMI stellt eine Anwendung der Beschreibungssprache XML dar und wurde speziell für den Austausch von MOF-konformen Metadaten konzipiert. Hierzu stellt XML vor allem zwei Arten von sog. Production Rules zur Verfügung. Während die eine Art dieser Rules dazu dient, aus MOF-konformen Metamodellen wie dem CWM entsprechende XML Document Type Definitions abzuleiten, die die Syntaxdefinition für Metadaten enthaltende XML-Dokumente darstellen, können mithilfe der Production Rules der anderen Art Metadaten in XML-Dokumente kodiert bzw. aus XMI-Dateien die enthaltenen Metadaten extrahiert werden [Vgl. OMG (2002), S. 4-43; Jeckle 1999].

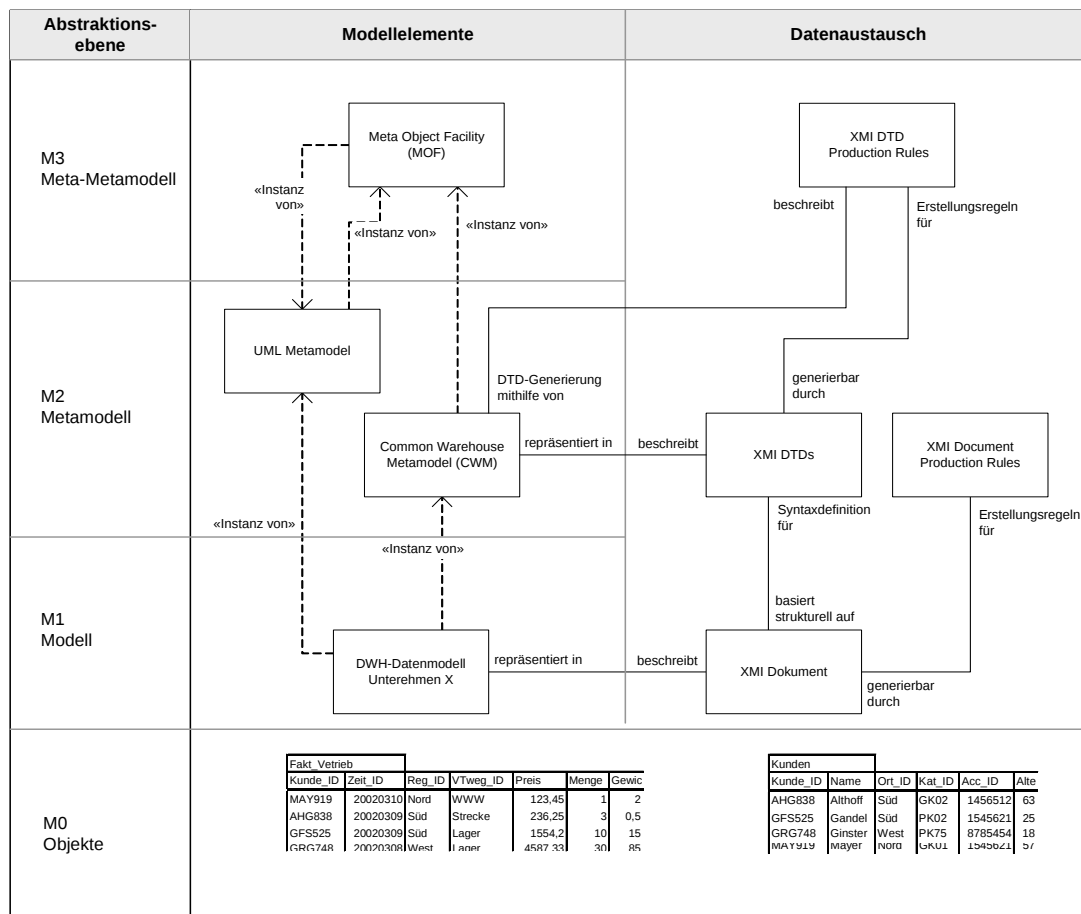


Abb. 4-2: Ebenen der CWM-Modellierung

Zusammenfassend bietet das CWM-Konzept somit einen Standard für die Modellierung und den Austausch von Metadaten und Metadatenmodellen im Data Warehousing. Die Zusammenhänge zwischen den einzelnen Bestandteilen des Konzeptes ist in Abb. 4-2 überblickartig dargestellt. Im Folgenden soll kurz auf Umfang und Struktur des eigentlichen Metamodells, in das Konzept integrierte Erweiterungsmechanismen sowie auf eine Einschätzung der Nutzenpotentiale der einzelnen Standardpakete aus Sicht der Praxis eingegangen werden.

4.2.2 Vorstellung der CWM-Packages

Bei der Entwicklung des CWM wurden neben der Festlegung auf die Modellierungssprache UML sowie die Klassendiagramm-Notation i. W. zwei Ziele verfolgt. Aus technischer Sicht sollte das CWM der typischen Werkzeugvielfalt und Heterogenität von DWH-Systemen gerecht werden. Dem Anspruch, einen plattformunabhängigen und trotzdem möglichst umfassenden Standard zu definieren, konnte das OMG-Entwicklerteam daher nur durch die Definition eines umfangreichen und entsprechend komplexen Metamodells gerecht werden, das insgesamt über 200 Klassen beinhaltet, die über mehr als 150 Assoziationen miteinander in Beziehung stehen. Mit Blick auf die Anwendbarkeit des CWM war jedoch das zweite erklärte Ziel, ein möglichst verständliches und leicht zu implementierendes Metamodell zu konstruieren. Um dementsprechend die Modellkomplexität beherrschbar zu machen, wurde das CWM in sog. Packages aufgeteilt, wobei jedes Package nur die Klassen und Assoziationen umfasst, die zur Definition einer einzelnen Metadatenart erforderlich sind. Die Beziehungen zwischen Packages werden unter extensiver Nutzung des UML-Vererbungskonzepts hergestellt. Jede CWM-Klasse geht aus einer ein- oder mehrstufigen Spezialisierung der Klasse *Element* hervor und fügt sich so in eine CWM-Vererbungshierarchie ein, wie sie in Abb. 4-3 überblicksartig dargestellt ist. Ein Package hängt somit über die in ihm enthaltenen Klassen von all denjenigen Packages ab, die die korrespondierenden Superklassen enthalten.

Um die Verflechtungen zwischen den Packages in überschaubaren Grenzen und die Gesamtzahl der Modellelemente möglichst gering zu halten, werden Assoziationen so häufig wie möglich wiederverwendet. Eine Assoziation zwischen zwei Klassen wird dem Metamodell nur dann neu hinzugefügt, wenn nicht schon auf einer höheren Ebene der Vererbungshierarchie die jeweiligen Superklassen miteinander assoziiert sind. Diese Art der Mehrzweck- oder Wiederverwendung von Assoziationen ist dabei nur ein Hilfsmittel, da UML in der verwendeten Version 1.3 keine Vererbung binärer Assoziationen erlaubt. Abb. 4-4 zeigt das Prinzip der Wiederverwendung an einem Beispiel.

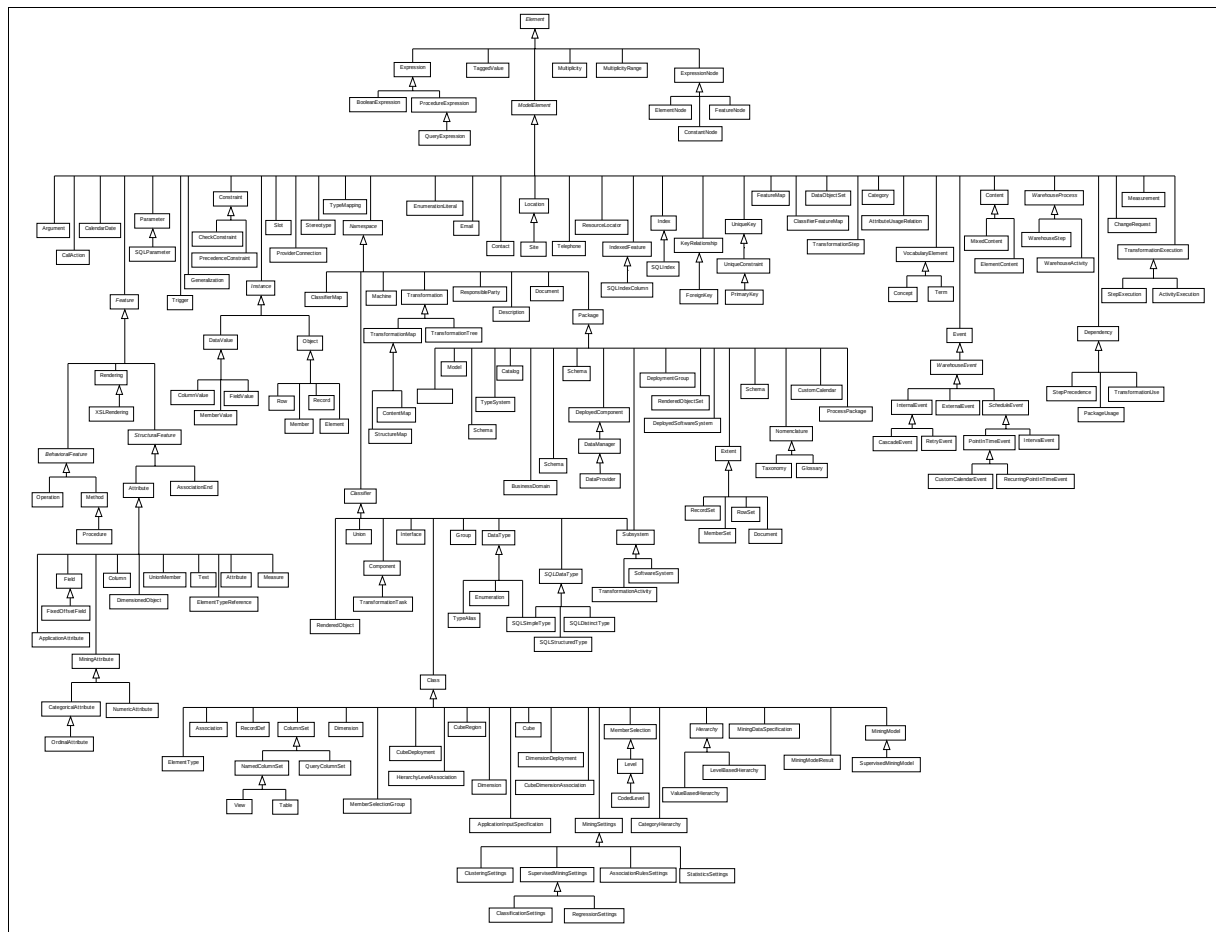


Abb. 4-3: Vererbungshierarchie aller Klassen des CWM

Sowohl im Package Relational als auch im Package Record wird eine Ganzes-Teile-Beziehung zwischen einem Datencontainer wie der Tabelle (Relational-Klasse *Table*) oder dem Record (Record-Klasse *RecordDef*) und den jeweils darin enthaltenen Einzelvariablen wie dem Tabellenfeld (Relational-Klasse *Column*) oder dem Recordfeld (Record-Klasse *Field*) benötigt. Anstatt die entsprechenden Assoziationen in jedem einzelnen Package zu modellieren, bedient sich das CWM der Vererbung. Die erwähnten Datencontainer erben beide von derselben Klasse *Class* des Core-Pakets, die wiederum eine Spezialisierung des Konstrukts *Classifier* darstellt. Ebenso erben die jeweiligen Einzelvariablen von derselben Klasse *Attribute*, die ihrerseits von der Core-Klasse *Feature* erbt. Durch die im Core-Paket enthaltene Komposition zwischen *Classifier* und *Feature* ist es nun möglich, in den Packages Relational und Record auf gleichbedeutende Kompositionen zwischen *Table* und *Column* bzw. zwischen *RecordDef* und *Field* zu verzichten und stattdessen auf die „vererbte“ Assoziation des Core-Pakets zurückzugreifen. Der gesamte dunkelgrau markierte Bereich in Abb. 4-4 ist nicht explizit im CWM modelliert sondern stellt lediglich die Semantik dar, die das CWM-

Entwicklerteam zum Ausdruck bringen wollte [vgl. Tolbert 2001, S. 12]. Die abgebildeten Vererbungsbeziehungen zwischen den Kompositionen sind dabei auch nicht UML-konform.

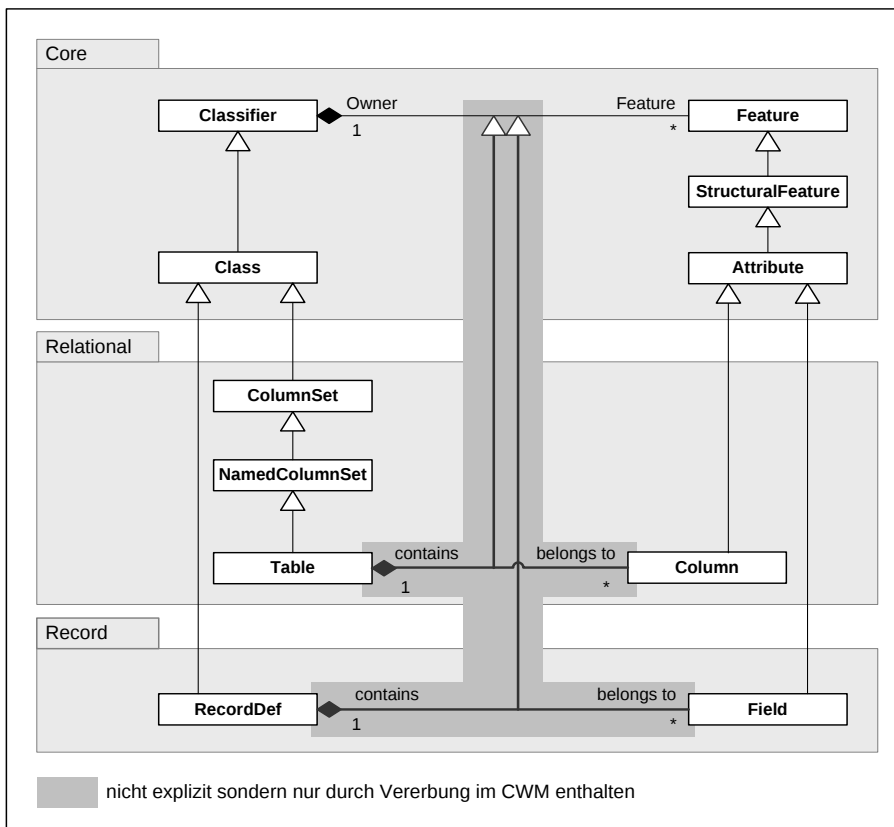


Abb. 4-4: Beispiel für die Mehrfach-Verwendung von Assoziationen im CWM
[vgl. Tolbert 2001, S. 12]

Durch seine modulare Struktur verfügt das CWM über insgesamt 22 Packages, die wie in Abb. 4-5 dargestellt zur besseren Übersichtlichkeit auf fünf Schichten angeordnet werden. Jede Schicht repräsentiert dabei die Modellbestandteile zu Metadaten einer Entwicklungsstufe des Data Warehousing. In der untersten Schicht wird zunächst die Modellierungssprache definiert, auf der alle weiteren Schichten aufbauen. Die zweite Schicht gruppiert Metamodelle für die Spezifikation von Grundbausteinen wie Datentypen, Ausdrücken oder Architekturmodelle, während auf der dritten Schicht verschiedene Datenstrukturtypen beschrieben werden. Die vierte Schicht enthält Modelle zur Beschreibung von transformations- und analysebezogenen Aktivitäten und die oberste fünfte Schicht definiert Metamodelle für Management- und Betriebsprozesse des Data Warehouse. Im Bezug auf die Abhängigkeiten zwischen den Packages kann festgehalten werden, dass Pakete einer Schicht nur von anderen Paketen derselben oder solchen darunter liegender Schichten abhängen können. Im Folgenden wird kurz auf

die einzelnen Schichten sowie die jeweils enthaltenen Packages eingegangen [vgl. insb. Poole u. a. 2002, S. 82-131].

Management	Warehouse Process			Warehouse Operation		
Analysis	Transformation	OLAP	Data Mining	Information Visualization	Business Nomenclature	
Ressource	Object Model (UML)	Relational	Record	Multidimensional	XML	
Foundation	Business Information	Data Types	Expressions	Keys and Indexes	Type Mapping	Software Deployment
Object Model	Core	Behavioral	Relationships	Instance		

Abb. 4-5: Die Packages des CWM

Der Object-Model-Layer

Das CWM hat als vollständiges Metamodell den Anspruch, auch die benutzte Modellierungssprache zu definieren. Hierzu wurde ein Subset des UML-Metamodells in das CWM übernommen und in vier Packages beschrieben, die zusammen die unterste Schicht, den Layer Object Model, bilden. Das Package Core bildet dabei die Basis des gesamten CWM, indem es die Ursprungsklasse Model Element in grundlegende statische Objektmodellbestandteile wie Klassen, Packages, Constraints u. a. spezialisiert. Als Basispackage weist Core als einziges keine Abhängigkeiten von anderen Metamodellteilen auf. Das Behavioral Package modelliert verhaltensbezogene Aspekte von Modellelementen wie Operationen, Ereignisse und Schnittstellen. Unterschiedliche Arten von Beziehungstypen werden im Package Relationship spezifiziert. Mithilfe des Instance-Packages können Instanzen der übrigen Modellkonstrukte abgebildet werden, wodurch die Verknüpfung von Daten und Metadaten möglich wird.

Der Foundation Layer

Während die Object Model Schicht lediglich allgemeine Modellkonstrukte definiert, stellt der Foundation Layer verschiedene CWM-spezifische Grundbausteine für die darüberliegenden Schichten bereit. Im Business Information Package werden verschiedene Zusatzinformationen zu Modellelementen wie etwa Beschreibungen, Dokumente, und Ansprechpartner samt Kontaktdaten konstruiert. Das Paket DataTypes liefert Konstrukte zur Definition sowohl elementarer als auch komplexer Datentypen. Den Austausch von Ausdrücken verschiedener künstlicher Sprachen ermöglicht das Paket Expression. Neben der undifferenzierten Behandlung von Ausdrücken in unbekannter Notation bietet das Paket auch die Möglichkeit, die einzelnen Bestandteile von Ausdrücken in komplexen Strukturen abzubilden und so die komplet-

te Semantik austauschbar zu machen. Mithilfe des Packages Keys and Indexes werden die Strukturen für die Abbildung von Indizes und Schlüsseln spezifiziert. In Ergänzung zum Package DataTypes ermöglichen es die Konstrukte des Pakets TypeMapping, verschiedene einfache Datentypen aufeinander abzubilden, sofern dies ohne Transformation der entsprechenden Werte möglich ist. Das Paket Software Deployment ermöglicht die Modellierung von DWH-Systemarchitekturen, wobei die Soft- und Hardwarekomponenten und –instanzen definiert als auch die zwischen den einzelnen Architekturkomponenten bestehenden Informationsmittel samt Verbindungsinformationen abgebildet werden können.

Der Resource Layer

Die Pakete der Resource Schicht dienen der Definition von Datenstrukturformaten. Jedes Format kann dabei sowohl von Quell- als auch von Zielsystemen des Datentransformationsprozesses implementiert sein. Für die Definition einer objektorientierten Datenstruktur werden die Konstrukte des Object Model Layer wiederverwendet. Das Package Relational dient der logischen Modellierung relationaler Datenbanken und greift dazu insbesondere auf die Metamodellkonstrukte der Pakete DataTypes und KeysIndexes zurück. Das Metamodell erlaubt die Definition von relationalen und objekt-relationalen Datenstrukturen, die dem Standard SQL99 entsprechen. Mithilfe des Packages Record können die meist in prozeduralen Programmiersprachen verwendeten strukturierten Datentypen modelliert werden, während das Paket Multidimensional der Modellierung von multidimensionalen Datenstrukturen dient, wie sie häufig in Data Marts implementiert werden. Aufgrund der uneinheitlichen Formate kommerziell verfügbarer multidimensionaler Datenbanken definiert das Multidimensional-Paket lediglich die generischen Elemente wie Dimensionen und Dimensionsobjekte. Das XML-Paket definiert die Basiselemente der Extensible Markup Language entsprechend dem vom WWW Consortium festgelegten Standard XML 1.0. Hierbei werden insbesondere die Strukturbestandteile von XML-Dokumenten in Form von XML Document Type Definitions spezifiziert.

Der Analysis Layer

Die Analyseschicht beinhaltet alle Metamodellbestandteile zur Beschreibung der typischen Kernaufgaben des Data Warehousing wie Datentransformation und Datenanalyse, wobei die enthaltenen Pakete auf den im Resource Layer definierten Datenstrukturtypen aufsetzen. Das Transformation Package spezifiziert die Metamodellkomponenten, die für die Definition von Transformationsprozessen benötigt werden. Hierbei unterscheidet die CWM-Terminologie

zwischen Mappings und tatsächlichen Transformationen. Während von Transformationen immer dann gesprochen wird, wenn Elemente zwischen verschiedenen Modellen der gleichen Transformationsstufe zueinander in Beziehung gesetzt werden sollen, beschreibt der Begriff Mapping die Zuordnung von Modellelementen unterschiedlicher Abstraktionsstufen. Die Überführung eines konzeptionellen Datenmodells in eine logische Tabellenstruktur einer relationalen Datenbank wird bspw. als Mapping, die Beschreibung der Umwandlungsbeziehungen zwischen den Feldern zweier relationaler Tabellen als Transformation bezeichnet [Vgl. Poole u. a. 2002, S. 121]. Die Granularität der darstellbaren Transformationsbeziehungen ist dabei frei wählbar, d. h. es können sowohl grobgranulare Transformationen zwischen ganzen Systemen, feingranulare Zuordnungen z. B. auf Feldebene relationaler Datenbanktabellen oder auch Beziehungen zwischen Informationsträgern mit unterschiedlichen Granularitätsstufen modelliert werden. Das OLAP Package definiert Metamodellelemente zur Beschreibung von multidimensionalen OLAP-Analysestrukturen und zwar unabhängig von den Speicherstrukturen, in denen die zu analysierenden Daten abgelegt sind. Hierzu können Analyse-Dimensionen definiert, mit Hierarchien versehen und ggf. mit einzelnen Bezugsobjekten verknüpft werden. Als Alternative zur expliziten Modellierung aller Bezugsobjekte einer Dimension können auch Verweise auf die Dimensionsbezugsobjekte enthaltende Quellen angegeben werden. Die Verbindung zwischen den Analysestrukturen und der physischen Realisierung wird mithilfe der Mapping-Konstrukte des Transformation-Packages hergestellt. Das Paket Data Mining spezifiziert Metamodellteile, mit deren Hilfe mathematische Modelle für das Data Mining beschrieben werden können. Neben Metaklassen zur Definition der Modelle enthält das Package auch Klassen zur Erfassung von Modellkonfigurationsparametern und mathematischen Skalentypen. Metadaten, die zur Beschreibung von Visualisierungsformen für Data-Warehouse-Daten benötigt werden, sind im Paket Information Visualization zusammengefasst. Aufgrund der sehr grossen Heterogenität existierender Darstellungsformen und der hohen Innovationsrate im Bereich der Datenvisualisierung enthält das CWM in diesem Bereich lediglich sehr generische Metamodellkonstrukte, die als Container für konkrete Konstrukte auf Modellebene fungieren sollen. Das Package Business Nomenclature definiert schliesslich Metaklassen, mit deren Hilfe Fachbegriffsmodelle und Glossare erstellt werden können und stellt so eine Möglichkeit bereit, fachliche Metadaten in den Standard zu integrieren.

Der Management Layer

In der Management-Schicht werden Metadaten definiert, die den technischen Betrieb des Data Warehouse unterstützen. Dabei dient das Warehouse Process Package insbesondere der Definition von Metadaten zum Scheduling und zur physischen Ausführung von Transformationen und Mappings. Hierzu werden Konstrukte zur Erstellung von Ereignissen und Aufgaben geliefert, mit denen sich ausführbare Prozesse abbilden lassen. Die Konstrukte des Packages Warehouse Operation dienen der Definition von Metadaten zur Dokumentation von Prozessauführungen, zur Festlegung von Messgrößen für Modellelemente sowie zur Erfassung von Änderungsanfragen zu Modellelementen.

4.3 Möglichkeiten der Erweiterung des CWM

Ein Standard soll vornehmlich die Etablierung von Kommunikationsbeziehungen zwischen mehreren Parteien vereinfachen. Hierzu bedarf es zunächst einer festen gemeinsamen Verständigungsbasis, die im Rahmen des Metadatenmanagements durch die 22 Packages des CWM geliefert wird. Verständigen sich jedoch zwei kommunizierende Parteien über die gemeinsame Verwendung einer semantisch reicheren Verständigungsbasis, so würde der Standard an sich nicht mehr ausreichen und dementsprechend für diese Austauschbeziehung nicht genutzt. Um dies zu verhindern, ist ein Anliegen des CWM-Standard, offen für Erweiterungen zu sein, sodass er zumindest immer als Basis für Metadaten austauschbeziehungen genutzt wird. Im Folgenden werden zwei verschiedene Mechanismen vorgestellt, mit deren Hilfe das CWM erweitert werden kann.

Erstellung eigener Subklassen

Durch die extensive Nutzung des Vererbungskonzepts in allen CWM-Packages wird erreicht, dass sich sämtliche Modellelemente auf eine gemeinsame Basis stützen und lediglich Erweiterungen erneut modelliert werden müssen. Mithilfe dieses Prinzips lassen sich auch nicht im CWM enthaltene Erweiterungen modellieren, bei denen zu einem gewissen Mass die Kompatibilität zum Standard gewährleistet bleibt. Problematisch ist dieser Erweiterungsmechanismus allein dadurch, dass er zusätzlich zu implementierende Metamodellelemente definiert. Wird im Rahmen eines CWM-Releasewechsels beispielsweise eine vormals definierte Klasse entfernt und gegen eine neue Klasse ersetzt, so verlieren alle von der ursprünglichen Klasse ererbenden Erweiterungen ihre Konformität zum Standard.

Tagged Values und Stereotypen

Um auch Erweiterungen umsetzen zu können, ohne dafür dem Metamodell neue Elemente hinzuzufügen, bietet das CWM die Klasse TaggedValue sowie das UML-Konstrukt der Stereotypen an. Durch mehrfache Instanziierung der TaggedValue-Klasse lassen sich zu jedem auf dem CWM basierenden Modellelement beliebig viele einwertige Attribute der Form *name* = *value* hinzufügen. Ein CWM-Releasewechsel – sofern er nicht gerade zu einer Veränderung der Klasse TaggedValue führt – würde solche Erweiterungen unberührt lassen. Ebenso wie bei der Nutzung nicht CWM-zugehöriger Subklassen müssen sich auch bei diesem Erweiterungsmechanismus die Kommunikationspartner über die Verwendung der spezifischen TaggedValues verständigt haben. Um diese Verständigung zu erleichtern, kann das UML-Konstrukt der Stereotypen verwendet werden. Hierbei kann jede Klasse als Stereotyp gekennzeichnet werden, indem man ihrem Namen die Bezeichnung *stbez* eines Stereotyps in der Form <<*stbez*>> voranstellt. Auf diese Weise wird kenntlich gemacht, dass der entsprechenden Klasse besondere Merkmale wie z. B. spezielle TaggedValues zugeordnet sind. Ein Problem dieser Erweiterungsvariante ist ihre limitierte Ausdrucksform. TaggedValues sind so modelliert, dass der Wert einer TaggedValue-Instanz immer vom Typ String sein muss. Der beschriebene Erweiterungsmechanismus eignet sich somit eher für kleinere Erweiterungen, für die die Modellierung einer eigenen Subklasse und deren Implementierung zu aufwendig wären.

Die CWM-Extension-Packages

Die 22 zuvor beschriebenen Packages des CWM bilden den allgemeinen Standard für Metadatenaustausch im Data Warehousing, über den alle Werkzeuge die grundlegenden Metadaten untereinander austauschen können. Die Erweiterungsmechanismen bieten darüber hinaus die Möglichkeit, werkzeugspezifische Ergänzungen zu modellieren. Einige dieser Erweiterungen werden bereits von der OMG als Vorschläge bereitgestellt, wobei wiederum die zu einem abgeschlossenen Bereich gehörenden Konstrukte in Packages zusammengefasst werden, die sich in eine der fünf Ebenen des CWM einordnen lassen. In der Foundation-Schicht existiert das Package ER, das die Modellelemente der Entity-Relationship-Modellierungssprache definiert. Auf der Ressource-Schicht werden gleich fünf weitere Pakete bereitgestellt, die plattformspezifische Speicherformen für die Programmiersprache COBOL, sowie für die Systeme IMS, DMSII, Hyperion Essbase und Oracle Express spezifizieren. Auf der Analysis-Schicht werden zusätzlich die Pakete InformationSet und InformationReporting bereitgestellt, die die CWM-Pakete OLAP bzw. Information Visualization um Metadaten für die Repräsentation

von externen Informationsquellen bzw. Berichten erweitern. Auf der Management-Schicht wird in der neuesten Version der CWM Extensions noch das Package DB2 Warehouse Manager definiert, das die für das gleichnamige DWH-Werkzeug benötigten Metadaten definiert und dabei eine Erweiterung der CWM-Pakete Core, SoftwareDeployment, Relational, Transformation, Business Nomenclature and WarehouseProcess darstellt. Während das CWM selbst als fester, nur von der OMG veränderbarer Standard definiert ist, stellen die CWM Extensions lediglich Vorschläge für mögliche systemspezifische Erweiterungsklassen dar, die jederzeit geändert oder erweitert werden dürfen, ohne dass dies Einfluss auf den CWM-Standard haben würde. Darüber hinaus versteht die OMG die Extension-Pakete als Beispiele, die die Anwender ermutigen und anleiten soll, eigene Erweiterungen zu modellieren.

Erweiterungsbedarf aus Sicht der Praxis

Der Umfang der CWM-Spezifikation erweckt leicht den Eindruck, es handle sich um einen allumfassenden Standard für DWH-Metadaten. Ob dieser Eindruck den Tatsachen entspricht, konnte aufgrund fehlender praktischer Anwendungsbeispiele bisher noch nicht festgestellt werden. Im Rahmen eines Workshops des CC DW2 hat sich daher einer Arbeitsgruppe mit der Frage beschäftigt, ob sich alle in den DWH-Projekten der teilnehmenden Unternehmen benötigten Metadaten tatsächlich mithilfe des CWM modellieren lassen. Während einer intensiven Diskussion hat sich herausgestellt, dass das CWM vor allem im Bereich der technischen Metadaten eine gute Abdeckung der praktischen Bedarfe ermöglicht. Allerdings konnten die beteiligten Unternehmensvertreter insbesondere im Bereich der fachlichen Metadaten auch Aufgabenstellungen identifizieren, für die das CWM keine adäquate Unterstützung bietet.

- Das CWM bietet keine Möglichkeit, Metadaten über Rollen und Zugriffsrechte abzubilden. Das CWM beinhaltet zwar Konstrukte zur Modellierung diverser Datenstrukturtypen, allerdings wurde komplett auf die Aufnahme der Zugriffsrechte-Bestandteile entsprechender Systeme verzichtet. Vor dem Hintergrund der Tatsache, dass die meisten DWH-Werkzeuge eigene Zugriffsrechte verwalten, könnte der Austausch der Autorisierungs-Metadaten die Erreichung von Interoperabilität erleichtern sowie zu einer besseren Transparenz der vergebenen Rollen und Zugriffsrechte führen.
- Die Berücksichtigung von Datenqualitätsmetadaten wurde allgemein als unzureichend eingeschätzt. Die einzige Möglichkeit, datenqualitätsbezogene Metadaten auf Basis des CWM zu modellieren, bietet die Klasse Measurement des Packages WarehouseOperation, mit deren Hilfe sich zu jedem Modellelement Kennzahlen definieren lassen. Für ein

effektives Datenqualitätsmanagement werden darüber hinaus aber auch Akzeptanzbereiche zur Einschätzung der Datenqualitätsmesswerte sowie Regeln sowohl zur Überprüfung von Kennzahlwerten, als auch zur Benachrichtigung verantwortlicher Personen bei Über- oder Unterschreiten des Akzeptanzbereiches benötigt. Entsprechende Konstrukte für diese Art von Metadaten sind bisher nicht im CWM enthalten.

- Im Bereich der analysebezogenen Metadaten definiert das CWM zwar Packages für OLAP und Data Mining, für andere Typen analytischer Informationssysteme wie bspw. Decision Support Systems o. ä. werden allerdings keine expliziten Metadatenkonstrukte bereitgestellt.
- Die im CWM spezifizierten OLAP-Metadaten dienen hauptsächlich der Darstellung von Analysedimensionen und darin enthaltener Hierarchien. Kennzahlen können zwar modelliert, allerdings nicht mit einem Berechnungsausdruck versehen werden. Darüber hinaus sieht das CWM nicht vor, dass die OLAP-Kennzahlen als Bestandteile der über das Business Nomenclature Package definierbaren Taxonomien geführt werden können. Eine wirkliche Integration von Metadaten verschiedener Quellen wird im Bereich der Kennzahl- und Fachbegriffsdefinitionen somit nicht erreicht. Darüber hinaus bietet das CWM zu wenig Möglichkeiten zur Verwaltung eines Berichtswesens. Berichtsdefinitionen sind entweder nur auf sehr generischer Ebene mithilfe des Information Visualization Packages oder durch Rückgriff auf das nicht standardisierte Extension Package InformationReporting modellierbar.

4.4 Fazit

Das CWM bietet eine gute Basis für den Austausch technischer Metadaten und deckt durch seine herstellerübergreifende Ausrichtung einen recht grossen Teil austauschwürdiger Metadaten ab. Bei der Abdeckung fachlicher Metadaten wie beispielsweise Taxonomien, Informationsbedarfen, Datenqualitätsregeln offenbaren sich allerdings einige Unzulänglichkeiten und ungeeignete Modellkonstrukte, durch die der Nutzen eines integrierten Metadatenmanagement insbesondere für den Endanwender in der Fachabteilung geringer ausfällt als möglich. Insbesondere für Entwicklung und Betrieb des Data Warehouse bietet das CWM aber durchaus enorme Vorteile, da es hilft, Konsistenzbrüche zu vermeiden, die durch die Verwendung unterschiedlicher Werkzeuge entstehen.

5 Zusammenfassung und Ausblick

Zielsetzung des vorliegenden Arbeitsberichts ist es, die Thematik des Metadatenmanagements im Data Warehousing näher zu beleuchten. Zu Beginn wird herausgestellt, dass die Formulierung einer abschliessenden Definition des Metadatenbegriffes für unrealistisch gehalten wird und stattdessen lediglich gewisse Merkmale identifiziert werden können, mit denen sich Metadaten von den durch sie beschriebenen Daten differenzieren lassen. Die Untersuchung der Einsatzmöglichkeiten und Nutzenpotenziale ergab, dass sich mithilfe eines integrierten Metadatenmanagements einerseits die Nutzung der DWH-Daten effektiver gestalten lässt, andererseits aber auch die Entwicklung eines DWH-Systems effizienter und nachhaltiger gestaltet werden kann. Als Beispiel hierfür wurde der Prototyp MetaMIS vorgestellt, der mithilfe eines zentralen, integrierten Metadaten Repository die Generierung von DWH-basierten Führungsinformationssystemen wesentlich erleichtert und teilautomatisiert. Auf Basis der Auswertung einer Praktikerbefragung zur Abschätzung der Grössen Aufwand, Nutzen und Praktikabilität einzelner Metadatenkategorien konnte eine grobe Empfehlung für die Reihenfolge erarbeitet werden, in der einzelne Metadatenkategorien in das Metadatenmanagement integriert werden sollten. Hiernach stellen Datenmodelle die praktikabelste Kategorie dar und sollten als erstes realisiert werden, um einen Quick Win für das Metadatenmanagement-Projekt nachweisen zu können.

Im Anschluss an die Nutzendiskussion wurden verschiedene Varianten für die Architektur eines Metadatenmanagementsystems vorgestellt. Während rein zentrale und rein dezentrale Metadatenhaltung als unpraktikabel eingestuft werden können, stellt das hybride Konzept eines zentralen, aber lediglich Kopien der lokalen Metadaten enthaltendes geteiltes Repository die unter den momentanen Rahmenbedingungen aussichtsreichste Architekturvariante für ein effektives Metadatenmanagementsystem dar. Eingebettet in eine DWH-Systemarchitektur wurde die Variante zu einem Metadata Warehouse erweitert. Schliesslich wurde die für die Realisierung der Metadatenintegration bedeutsame Frage der Standardisierung behandelt, wobei insbesondere das Common Warehouse Metamodel als entsprechende Lösung der OMG vorgestellt wurde. Die Diskussion der einzelnen Modellbestandteile ergab, dass das CWM zwar eine gute Basis für den Austausch vieler, vornehmlich technischer Metadaten darstellt, jedoch gerade im Bereich fachlicher Metadaten noch erheblicher Standardisierungsbedarf besteht.

Die bisherige Arbeit konzentriert sich auf die technischen Aspekte des Metadatenmanagements. Um die Metadatenverwaltung als festen Bestandteil des Data Warehousing dauerhaft zu etablieren, müssen aber insbesondere auch organisatorische Aspekte berücksichtigt werden. Die Bereitstellung, Verwaltung und Nutzung der Metadaten umfasst eine Vielzahl einzelner Aufgaben, die nur dann zu dem gewünschten Ergebnis führen, wenn sie aufeinander abgestimmt sind. Um dies zu gewährleisten muss im Rahmen der Entwicklung auch eine Ablauforganisation definiert und in die Organisation des Unternehmens eingebunden werden.

Kurzfristiges Ziel des CC DW2 ist es daher, die Prozesse des Metadatenmanagements näher zu untersuchen und ggf. ein Prozessreferenzmodell zu erarbeiten. Neben entsprechenden bilateralen Projekten wird hierzu ein weiterer Workshop des Kompetenzzentrums veranstaltet.

Literatur

[Agosta 2001]

Agosta, L.: Reports of the demise of meta data are premature. DM Review 3 (2001), URL: <http://www.dmreview.com/master.cfm?NavID=198&EdID=3080>, 12.10.2001.

[Barkin/Edelstein 1997]

Barquin R., Edelstein H., Planning and designing the data warehouse, New Jersey, Prentice Hall, 1997.

[Blechar 2001]

Blechar, M.: Four alternatives to data warehousing metadata integration. Gartner Research Note. O. O. 2001. URL: <http://www.gartner.com>, 2002-04-11.

[Buytendijk/Friedman 2001]

Buytendijk, F.; Friedman, T.: The packaged data warehouse myth. Gartner Research Note. O. O. 2001. URL: <http://www.gartner.com>, 2002-04-11.

[ComputerWire 1996]

Ohne Autor: What is metadata? In: ComputerWire, März 1996, URL: http://www.computerwire.com/bulletinsuk/212e_1a6.htm, 26.11.2001.

[Devlin 1997]

Devlin, B: Data Warehouse - from Architecture to Implementation. Reading et al. 1997.

[Do/Rahm 2000]

Do, H. H.; Rahm, E.: On metadata interoperability in data warehouses. Technischer Report 1-2000, Institut für Informatik, Universität Leipzig, 2000.

[Ferstl/Sinz 2001]

Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. Band1. 4., überarb. und erw. Aufl. Oldenbourg, München, Wien, 2001.

[Habermann/Leymann 1993]

Habermann, H.-J.; Leymann, F.: Repository. Eine Einführung. Oldenbourg, München, Wien, 1993.

[Holten 2001]

Holten, R.: Konstruktion domänenspezifischer Modellierungstechniken für die Modellierung von Fachkonzepten. Arbeitsbericht Nr. 78, Institut für Wirtschaftsinformatik der Universität Münster, Münster 2001.

[Hurwitz 1998]

Hurwitz Group: Enterprise metadata management. Hurwitz Balanced View Report. O. O. 1998. URL: <http://www.hurwitz.com/>, 2002-04-11.

[Inmon 1996]

Inmon, W. H.: Building the Data Warehouse. 2nd Edition. Wiley, New York et al., 1996.

[Inmon 2001]

Inmon, W. H.: An Illustrated Taxonomy of Metadata. White Paper. O. O. 2001. URL: <http://www.billinmon.com>, 2002-04-11.

[Jeckle 1999]

Jeckle, M.: Modellaustausch mit dem XML Metadata Interchange Format. In: Objekt-Spektrum 5, 1999. URL: <http://www.jeckle.de/modellaustausch/index.html>, 05.12.2001.

[Jennings 2001]

Jennings, M. F.: The generic meta data repository. DM Direct 1 (2001). URL: <http://www.dmreview.com/master.cfm?NavID=198&EdID=2954>, 27.11.2001.

[Kimball 1998]

Kimball R., Reeves L., Ross M., Thornthwite W., The data warehouse lifecycle toolkit, New York, Wiley, 1998.

[Khoshafian/Abnous 1995]

Khoshafian, S.; Abnous, R.: Object orientation – concepts, analysis and design, languages, databases, graphical user interfaces, standards. 2. Auflage. Wiley, New York et al., 1995.

[Marco 2000]

Marco, D.: Building and managing the meta data repository. A full lifecycle guide. Wiley, New York, 2000.

[Mucksch/Behme 2000]

Mucksch, H.; Behme, W.: Das Data Warehouse-Konzept als Basis einer unternehmensweiten Informationslogistik. In: Mucksch, H.; Behme, W. (Hrsg.): Das Data Warehouse-Konzept. Architektur – Datenmodelle – Anwendungen. 4., vollst. überarb. und erw. Auflage. Gabler, Wiesbaden, 2000.

[Müller/Stöhr/Rahm 1999]

Müller, R.; Stöhr, T.; Rahm, E.: An integrative and uniform model for metadata management in data warehousing environments. In: Proc. of International Workshop on Design and Management of Data Warehouses (DMDW '99 at CaiSE '99), Heidelberg, 1999, S. 12/1 – 12/16, URL: <http://dol.uni-leipzig.de/pub/1999-22>, 27.11.2001.

[Narayan 1988]

Narayan, R.: Data dictionary: implementation, use and maintenance. Prentice-Hall, Englewood Cliffs, NJ, 1988.

[OMG 2001]

OMG: Meta Object Facility (MOF) Specification. Version 1.3.1. URL: <http://www.omg.org/cgi-bin/doc?formal/2001-11-02>, 08.04.2002.

[OMG 2002]

OMG: Common Warehouse Metamodel (CWM) Specification. Version 1.1. URL: <http://cgi.omg.org/cgi-bin/doc?ptc/02-01-07>, 08.04.2002

[Ortner 1998]

Ortner, E.: Normsprachliche Entwicklung von Informationssystemen. In: Proc. des GI-Workshops Modellierung 98, Münster, 1998, URL: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-9/Ortner.ps>, 26.11.2001.

[Ortner 1999]

Ortner, E.: Repository Systems. Teil 1: Mehrstufigkeit und Entwicklungsumgebung. In: Informatik Spektrum 4 (1999), S. 235 – 251.

[Poole et al. 2002]

Poole, J.; Chang, D.; Tolbert, D.; Mellor, D.: Common warehouse metamodel: an introduction to the standard for data warehouse integration. New York et al. 2002.

[Seiner 1999]

Seiner, R. S.: Metadata is the answer – now what is the question? In: The Data Administration Newsletter 4, 1999, URL: <http://www.tdan.com>.

[Schwarz 2000]

Schwarz, S.: Metadaten – Integrationskomponente des Data Warehouse. In : [Jung/Winter 2000], S. 101 – 116.

[Staudt/Vaduva/Vetterli 1999]

Staudt, M.; Vaduva, A.; Vetterli, T.: The role of metadata for data warehousing. Technical Report 99.06. Institut für Informatik, Universität Zürich, 1999, URL: <ftp://ftp.ifi.unizh.ch/pub/techreports/TR-99/ifi-99.06.ps.gz>, 28.11.2001.

[Strahringer 1998]

Strahringer, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In: Proc. des GI-Workshops Modellierung 98, Münster, 1998, URL: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-9/Päch.ps>, 26.11.2001.

[Strahringer 1999]

Strahringer, S.: Probleme und Gefahren im Umgang mit „Meta“-Begriffen: Ein Plädoyer für eine sorgfältige Begriffsbildung. In: Proc. der KnowTech '99, Potsdam, 1999,
URL: <http://www.bwl.tu-darmstadt.de/bwl5/forsch/veroeff/strahKTF99.pdf>,
26.11.2001.

[Tolbert 2001]

Tolbert, D.: The CWM Experience Implementing a UML-Based Data Warehouse Metamodel, http://www.cwmforum.org/CWM_UML_Experience.ppt, 04.08.2002

[Tozer 1999]

Tozer, G.: Metadata management for information control and business success. Artech House, Boston, 1999.

[Vaduva/Vetterli 2001]

Vaduva, A.; Vetterli, T.: Metadata management for data warehousing: an overview. In: Cooperative Information Systems Vol. 10, No. 3 (2001), S. 273 – 298.

[Vaduva/Dittrich 2001]

Vaduva, A.; Dittrich, K. R.: Metadata management for data warehousing: between vision and reality. In: Proc. of the Int. Database Engineering & Applications Symposium (IDEAS '01), 2001.

[Volck 2000]

Volck, R.: Data Warehouse-Technologie im Umbruch. Metadatenzentrierte Lösungen erobern den Markt. In: Computerwoche 12/2000, 24.03.2000, S. 17.

[Widom 1995]

Widom, J.: Research problems in data warehousing. In: Proc. of the 4th Int. Conference on Information and Knowledge Management (CIKM), 1995.

[Wiener 2000]

Wiener, J: Metadata in context. In: DM Direct 2 (2000),

URL: <http://www.dmreview.com/master.cfm?NavID=198&EdID=1926>, 26.11.2001.

[Wu/Miller/Nilakanta 2000]

Wu, L.; Miller, L.; Nilakanta, S.: Design of data warehouse using metadata. In: Information and Software Technology 43 (2001), S. 109 – 119.