

# Das Common Warehouse Metamodel - ein Referenzmodell für Data-Warehouse-Metadaten

Florian Melchert, Alexander Schwinn, Clemens Herrmann

Institut für Wirtschaftsinformatik  
Universität St. Gallen  
Müller-Friedberg-Strasse 8  
CH-9000 St. Gallen

{florian.melchert;alexander.schwinn;clemens.herrmann}@unisg.ch

**Abstract:** Der vorliegende Beitrag beschreibt das Vorgehen zur Realisierung eines integrierten Metadatenmanagements im Data-Warehouse-Projekt einer Schweizer Universalbank. Untersucht wurde insbesondere, inwieweit das Common Warehouse Metamodel der Object Management Group geeignet ist, als Referenzmodell für die Metadatenbasis herangezogen zu werden. Nach einem Überblick über das Referenzmodell werden das Vorgehen zu seiner Anwendung sowie die dabei identifizierten Stärken und Schwächen beschrieben.

## 1 Einsatz eines Referenzmodells zur Vereinheitlichung des Metadatenbestandes im Data Warehousing

Das Fehlen einer einheitlichen unternehmensweiten Sicht auf entscheidungsunterstützende Informationen veranlasste eine große Schweizer Universalbank gegen Ende der 1990er Jahre dazu, ein Data-Warehouse-Projekt zu initiieren und die zuvor genutzten bereichsspezifischen Management Support Systeme bereits zu großen Teilen durch auf einem Data Warehouse (DWH) basierende analytische Informationssysteme zu ersetzen (vgl. [Me00, S. 46ff.]). Zudem gelangte man zu der Einsicht, dass sich das Nutzenpotenzial des Data Warehousing besser ausnutzen ließe, wenn man die bisher größtenteils aufgaben- und/oder personenspezifisch erfolgende Verwaltung der DWH-Metadaten über ein zentrales Metadaten Repository (MDR) personen- und aufgabenübergreifend verfügbar macht. Grundvoraussetzung für die Realisierung eines entsprechenden Systems war die Erstellung eines integrierten Metadatenmodells, welches Syntax und Semantik der vorhandenen Metadaten einheitlich festlegt. Um den Modellierungsaufwand möglichst zu minimieren, gleichzeitig aber eine hohe Qualität und vor allem eine hohe Akzeptanz des MDR bei allen Nutzern zu realisieren, entschied man sich, auf das Common Warehouse Metamodel (CWM) der Object Management Group als standardisiertes Referenzmodell für Data-Warehouse-Metadaten zurückzugreifen. Im Folgenden Kapitel wird das CWM zunächst kurz vorgestellt, bevor anschließend der praktische Einsatz im Rahmen des konkreten Projekts und die dabei gewonnen Erfahrungen näher erläutert werden. Der Beitrag schließt mit einer kritischen Betrachtung des CWM sowie einem Ausblick auf das weitere Vorgehen.

## 2 Das OMG Common Warehouse Metamodel

Das Common Warehouse Metamodel (CWM) ist ein Referenzmodell für Metadaten<sup>1</sup> in der Domäne Data Warehousing, das seit 1998 unter aktiver Beteiligung mehrerer Softwarehersteller entwickelt wird und seit 2000 als von der Object Management Group (OMG) verabschiedeter Standard vorliegt (vgl. [OM03, S. xviii ff.]). Wesentliches Charakteristikum des CWM ist seine Einbettung in die Model Driven Architecture der OMG, (vgl. [OM00, S. 23 ff.]) deren Ziel die Integration der OMG Standards zu einem ganzheitlichen Modellierungsansatz ist (vgl. [Ke02, S. 288]). Dementsprechend existiert eine Reihe von Verknüpfungen zu anderen Standards, die einerseits einen möglichst hohen Grad an Wiederverwendung bestehender Konstrukte bei der CWM-Erstellung garantieren und andererseits die Überführung des plattformunabhängigen Referenzmodells in plattformspezifische Modelle erleichtern sollen (vgl. [Me03, S. 94 f.]).

Das Referenzmodell ist als UML-Klassenmodell verfügbar und besteht aus 204 Klassen, die über etwa 150 Assoziationen miteinander verbunden sind. Zur Erhöhung der Übersichtlichkeit sind die Modellelemente thematisch in 21 Pakete aufgeteilt, wobei jedes Paket wiederum auf einer von fünf thematischen Schichten angeordnet wird (vgl. auch zum Folgenden [Me03, S. 97 ff.]). In der untersten Schicht *Object Model* wird die Modellierungssprache definiert, auf der alle weiteren Schichten aufbauen. Die *Foundation*-Schicht definiert Metamodelle für die Spezifikation von Grundbausteinen wie Datentypen, Ausdrücken, Typcasting- und Indexierungs-Konstrukten ebenso wie Elementen zur Modellierung von Verantwortlichkeiten oder von Softwaresystem-Architekturen. Die Schicht *Resource* enthält Pakete zur Definition der Datenstrukturen von Quell- oder Zielsystemen des DWH. Die *Analysis*-Schicht umfasst Konstrukte zur Modellierung von Datentransformationsprozessen sowie von verschiedenen Formen der multidimensionalen Datenauswertung. Die *Management*-Schicht definiert Elemente, die zur Steuerung und Überwachung der technischen Transformationsprozesse benötigt werden. Jede CWM-Klasse fügt sich in eine Vererbungshierarchie ein, durch die das Referenzmodell einen recht hohen Integrationsgrad erreicht. Neue Konstrukte werden grundsätzlich nur hinzugefügt, wenn entsprechende Elemente bisher nicht vorhanden sind.

Das CWM ist als Referenzmodell vor allen Dingen technischer DWH-Metadaten entwickelt worden, das insbesondere den Metadatenaustausch zwischen verschiedenen Softwarekomponenten der DWH-Architektur erleichtern soll [OM03, S. xvi]. Durch die Konformität zu anderen OMG-Standards können CWM-konforme Metadaten entweder über XML-Dokumente dateibasiert ausgetauscht oder mithilfe von CORBA-Schnittstellen einem funktionsbasierten Zugriff zugänglich gemacht werden (vgl. [Me03, S. 95 ff.]). Eine dritte Variante ergibt sich, wenn das CWM als Datenmodell eines Metadaten Repository genutzt wird, in dem CWM-konforme Metadaten verschiedener Herkunft verwaltet werden können. Mit einem als CWM Metastore bezeichneten DDL-Skript demonstriert das CWM-Entwicklerteam zudem einen viel versprechenden Ansatz, wie sich Repository- und funktionsorientierter Zugriff auf Basis eines relationalen Datenbankmanagementsystems kombinieren lassen (vgl. [Po03, S. 431-554]).

---

<sup>1</sup> Zur Klärung der Begriffe Metadaten, Metadatenmodell und Metamodel vgl. [Me03, S. 92]; [St98]; [Au03].

### 3 Das CWM in der praktischen Umsetzung

Nachdem das Metadatenteam des DWH-Projekts vor allem aufgrund der Herstellerneutralität, der Konformität zu anderen frei verfügbaren Standards und seiner inhaltlichen Fokussierung auf DWH-relevante Metadaten das CWM als Referenzmodell ausgewählt hatte, entschied man sich dafür, die Referenzmodell-anwendung in einem abgegrenzten Anwendungsbereich zu testen. Hierfür wählte man die erste Phase des Datenextraktions- und -transformationsprozesses, in der dateibasierte Daten operativer Legacy-Systeme bereinigt und in relationale Datenbanktabellen überführt werden. Da dieser Prozess bereits weitgehend automatisiert ist und über Metadaten aus verschiedenen Quellen gesteuert wird, ließ sich hieran sehr gut untersuchen, inwieweit das CWM geeignet ist, die existierenden Metadatenmodelle abzubilden und zu einem integrierten Modell zu vereinen. Bei den vorhandenen Metadaten handelt es sich hauptsächlich um Definitionen recordbasierter und relationaler Datenstrukturen sowie um Transformationsbeziehungen. Um die vorliegenden Modelle in eine CWM-konforme Form zu überführen, galt es zunächst, die für den betrachteten Bereich relevanten Referenzmodellbestandteile auszuwählen (vgl. [Sc98, S. 313 f.]). Die nach Metadatenkategorien gebildeten Packages des CWM dienen hierbei als erster Anhaltspunkt für eine Vorselektion, jedoch erforderte die genaue Identifikation der benötigten Konstrukte einen detaillierten Abgleich auf Klassen- bzw. Attributebene des Referenzmodells. Dieser Abgleich offenbarte, dass sich zunächst ein Grossteil der betrachteten Metadatenklassen durch CWM-Klassen abdecken lässt, wobei insbesondere Klassen der folgenden CWM-Pakete zum Einsatz kommen: Das Package *Record* liefert die grundlegenden Klassen zur Modellierung von Records und Recordfeldern, die meisten der gepflegten Eigenschaften können jedoch nicht abgebildet werden. Im Package *Relational* finden sich Konstrukte für die Modellierung von Metadaten relationaler Datenbanken. Metadaten zur Modellierung technischer ETL-Prozesse konnten aus dem Paket *Transformation* bezogen werden. Das Package *Business Information* erlaubt über die Klassen *Description* und *ResponsibleParty* die Modellierung von Beschreibungsattributen sowie von Verantwortlichkeiten. Mithilfe des Packages *SoftwareDeployment* können Metadaten zu eingesetzten Informationssystemen, deren Systemkomponenten sowie zu Datenbankschnittstellen modelliert werden. Das Package *Core* stellt als Kernbestandteil des CWM zudem die Superklassen sowie grundlegende Assoziationen zur Verfügung, die im Rahmen der Wiederverwendung in anderen Packages benötigt werden.

Über die genannten Konstrukte hinaus wurden jedoch auch Bereiche identifiziert, die nicht ausreichend oder gar nicht im CWM berücksichtigt wurden. Besonders schwerwiegend fiel dies im Bereich der Metadatenversionierung auf, der im betrachteten Anwendungsfall einen unverzichtbaren Teil darstellt. Ebenfalls als unzureichend durch das Metamodell abgedeckt wurde der Bereich der Berechtigungsverwaltung eingestuft. Bei der Untersuchung auf Attributebene offenbarte sich zudem, dass bis auf wenige Ausnahmen nur grundlegende Attribute wie der Objektname oder etwaige Beschreibungen bzw. Kommentare in den Konstrukten des CWM vorgesehen sind. Zusätzlich benötigte Attribute wie z. B. das Datum der letzten Änderung oder technische Angaben wie z. B. Datensatztrennzeichen oder Default-Werte sind i. d. R. nicht modelliert.

Die dargelegten Unzulänglichkeiten machten eine Erweiterung des Referenzmodells erforderlich (vgl. [Sc98, S. 314]), für die das CWM mit Stereotypen, Stereotypen in Verbindung mit Tagged Values sowie Erweiterungsklassen drei Mechanismen zur Verfügung stellt, von denen der zweckmäßigste auszuwählen war (vgl. [Po02, S. 207-219]). Im vorliegenden Fall bot sich allein der Mechanismus der Erweiterungsklassen an, da nur dieser die Modellierung zusätzlicher, nicht-textueller Attribute erlaubt. Zudem trägt dieser Mechanismus dem Modularitätsgedanken des CWM und somit der Übersichtlichkeit besser Rechnung als eine Lösung über Stereotypen und Tagged Values. Darüber hinaus besteht die Möglichkeit, die konstruierten Erweiterungsklassen inhaltlich zu Erweiterungs-Packages zusammenzufassen und zu veröffentlichen. Dieses Vorgehen wird auch von der OMG begrüßt, die einige bereits fertig gestellte Erweiterungs-Packages zu einem Package *CWM Extensions* (CWMX) zusammenfasst und in Ergänzung zum CWM-Standard anbietet (vgl. [OM01]).

Zur Komplettierung des Testszenarios wurde schließlich die Modellierung von unternehmensspezifischen Erweiterungsklassen durchgeführt. Um eine möglichst hohe Konformität der neu zu konstruierenden Elemente mit den bestehenden CWM-Bestandteilen sicherzustellen, machte man hierbei intensiven Gebrauch vom Vererbungsprinzip des CWM. So konnten die meisten zusätzlich benötigten Attribute als Bestandteile neu angelegter Subklassen definiert werden, deren Beziehungen zueinander nicht zusätzlich modelliert werden mussten, da sie bereits durch vererbte Assoziation zwischen CWM-Superklassen zur Verfügung standen.

#### **4 Kritische Würdigung des CWM-Einsatzes und Ausblick**

Im Bereich der betrachteten technischen DWH-Metadaten musste das Metadatenteam erkennen, dass sich die unternehmensspezifischen Konstrukte mithilfe des CWM nur teilweise abbilden lassen. Der Grund hierfür liegt in der niedrigen Definitionstiefe des Referenzmetamodells, welches eher breit im Sinne der Unterstützung möglichst vieler verschiedener Metadatenkategorien ausgelegt ist. In den einzelnen Kategorien werden i. d. R. nur die allgemein benötigten Elemente und Beziehungen definiert. Im Bereich der technischen Metadaten ist daher damit zu rechnen, dass Spezifika bestimmter Softwarelösungen – wie sie bspw. im betrachteten Anwendungsfall existieren – nicht Bestandteil des Referenzmodells sind. Dennoch scheint es auch in diesem Bereich aus drei Gründen durchaus sinnvoll zu sein, die Konstruktion eines konzeptionellen Metadatenmodells auf dem CWM basieren zu lassen. Zum Ersten kann auf diese Weise zumindest ein Teil der vorhandenen technischen Metadaten standardkonform abgebildet werden, wodurch sich die Möglichkeiten zum Metadaten austausch verbessern. Zum Zweiten ermöglichen die CWM-Erweiterungsmechanismen bei richtig verstandener Anwendung der CWM-Modellierungsprinzipien die flexible Anpassung des Referenzmodells an unternehmensspezifische Anforderungen bei gleichzeitiger Wahrung einer hohen Standardkonformität durch einen engen Bezug der Erweiterungen zu den Standardelementen (vgl. [Po02, S. 223 ff.]). Zum Dritten bietet sich durch die von der OMG propagierten CWM Extension Packages die Möglichkeit, eigene Modellteile zu verbreiten bzw. Modellteile anderer CWM-Nutzer wieder zu verwenden.

Die nächsten Schritte zur Realisierung eines integrierten Metadaten Repository wären die Erstellung des konzeptionellen Metadatenmodells sowie dessen anschließende Implementierung. Aufgrund der bereits im Einsatz befindlichen Software sowie des im Unternehmen vorhandenen Know-hows erscheint die Realisierung in einer relationalen Datenbank sinnvoll. Eine Ausgangsbasis für die Entwicklung könnte das vom CWM-Entwicklerteam propagierte Datenbankschema CWM Metastore sein, das das CWM-Modell auf eine relationale Datenbank abbildet und mit Datenzugriffsfunktionen anreichert. Mit der Implementierung des integrierten Metadaten Repository würde eine wesentliche Basis für ein umfassendes Metadatenmanagement geschaffen, dass neben der integrierten Verwaltung auch die Pflege sowie die Nutzung der Metadaten umfasst und organisatorisch in das Data Warehousing integriert wird. Hinsichtlich der Weiterentwicklung des CWM wäre eine Erweiterung um bisher nicht enthaltene Aspekte wie Versionierung und Berechtigungsmanagement wünschenswert, die aus Praxissicht durchaus hohe Relevanz für DWH-Projekte haben und dementsprechend auch Bestandteil eines Referenzmodells für DWH-Metadaten sein sollten.

## Literaturverzeichnis

- [Au03] Auth, G.: Prozessorientierte Organisation des Metadatenmanagements für Data-Warehouse-Systeme. Universität St. Gallen, Bamberg 2003.
- [Ke02] Kent, S.: Model Driven Engineering. In: Butler, M; Petre, L.; Sere, K. (Hrsg.): Integrated Formal Methods. Third International Conference, IFM 2002, Turku, Finland, May 15-18, 2002. Proceedings. Lecture Notes in Computer Science 2335, Springer, Berlin et al. 2002, S. 286-299.
- [Me03] Melchert, F.: Das Common Warehouse Metamodel als Standard für Metadaten im Data Warehousing. In: von Maur, E.; Winter, R. (Hrsg.): Data Warehouse Management. Das St. Galler Konzept zur ganzheitlichen Gestaltung der Informationslogistik. Springer, Berlin et al. 2003. S. 89-111.
- [Me00] Meyer, M.: Organisatorische Gestaltung des unternehmensweiten Data Warehousing - Konzeption der Rollen, Verantwortlichkeiten und Prozesse am Beispiel einer Schweizer Universalbank. Bamberg 2000.
- [OM00] OMG Architecture Board ORMSC: Model Driven Architecture (MDA), ormsc/2001-07-01, OMG, o. O., 2000.
- [OM01] Common Warehouse Metamodel (CWM) Specification. Volume 2 - Extensions. Version 1.0. OMG, o. O., 2001.
- [OM03] OMG: Common Warehouse Metamodel (CWM) Specification. Volume 1. Version 1.1. formal-03-03-02, OMG, o. O., 2003.
- [Po02] Poole, J.; Chang, D.; Tolbert, D.; Mellor, D.: Common warehouse metamodel: An Introduction to the Standard for Data Warehouse Integration. John Wiley & Son, Inc., New York, et al. 2002.
- [Po03] Poole, J.; Chang, D.; Tolbert, D.; Mellor, D.: Common Warehouse Metamodel Developer's Guide. John Wiley & Son, Inc., New York, et al. 2003.
- [Sc98] Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Gabler, Wiesbaden 1998.
- [St98] Strahringer, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. Modellierung 98, Proceedings; Bericht Nr. 6/98-I, Angewandte Mathematik und Informatik, Münster, 1998, S. 15-20.