

Metadaten-gestützter Data-Warehouse-Entwurf mit ADAPTed¹ UML

Torsten Priebe, Günther Pernul²

Wirtschaftsinformatik und Informationssysteme

Universität Essen, Universitätsstr. 9, 45141 Essen

Email: {priebe|pernul}@wi-inf.uni-essen.de

Zusammenfassung: Die Bedeutung von Data Warehousing und Online Analytical Processing (OLAP) für entscheidungsunterstützende strategische Informationssysteme eines Unternehmens ist rapide gewachsen. Vor diesem Hintergrund gewinnen Ansätze zum Metadaten- und Repository-gestützten Entwurf von OLAP- und Data-Warehouse-Systemen vermehrt an Bedeutung. Bereits eine Reihe von Forschungsprojekten hat sich in den vergangenen Jahren mit dieser Thematik beschäftigt, meist in Zusammenhang mit einer grafischen Notation zur Modellierung. Die bekannten Ansätze beschränken sich dabei jedoch hauptsächlich auf die konzeptuelle (systemunabhängige) Ebene. In diesem Beitrag wird ein auf ADAPT und UML basierender Modellrahmen und ein Repository-zentrischer Architekturansatz zur Unterstützung des Übergangs vom Modell zur Implementierung vorgestellt.

Schlüsselworte: Data Warehouse, Entwurf, OLAP, UML

1 Einleitung

Eines der Hauptprobleme strategischer Informationssystemen stellt die Heterogenität der operativen Systeme dar. Vor diesem Hintergrund ist das Data-Warehouse-Konzept als Integrationsbasis für eine redundante Speicherung von Daten aus operativen Informationssystemen entstanden. Dieser Beitrag befasst sich mit dem Entwurf solcher Data-Warehouse-Systeme.

¹ ADAPT ist ein Warenzeichen der Symmetry Corporation, San Rafael, CA, U.S.A.

² Dieser Beitrag entstand mit Unterstützung der Europäischen Union im Rahmen INCO-COPERNICUS-Programmes, Projekt-Nr. 977091 (GOAL – Geographic Information Online Analysis).

Warum muss ein Data Warehouse überhaupt entworfen bzw. modelliert werden, wenn die Daten redundant aus den (ja bereits modellierten) operativen Datenbanken übernommen werden? Reichen bekannte Ansätze der Schemaintegration nicht aus, um ein konsolidiertes Warehouse-Datenmodell zu entwickeln? Über die Frage, ob das Modell eines Data Warehouse auf diese Weise tatsächlich (bottom-up) von den existierenden Daten her bestimmt wird [EJNW96], oder ob eine top-down Modellierung aus Sicht der zu analysierenden betriebswirtschaftlichen Kennzahlen sinnvoll und möglich ist [Toto00], herrscht keine Einigkeit. In der Regel werden beide Ansätze miteinander verbunden. Ein wichtiges, für eine saubere Modellierung sprechendes Argument ist die üblicherweise multidimensionale Sicht auf die Daten. Das Datenmodell eines Data Warehouse orientiert sich weniger an den Funktionen operativer Anwendungssysteme, sondern vielmehr an Analysethemen (vgl. Themenbezogenheit in [Inmo96] und Analyseorientierung in [BaGü00]). Bei Bedarf werden zusätzlich externe Daten einbezogen.

Der Beitrag entstand im Rahmen des Forschungsprojektes GOAL, das sich die Integration von geographischen Informationssystemen (GIS) und Data-Warehouse-Techniken zum Ziel gesetzt hat. Zwei Pilotanwendungen (A1 und A2) wurden für das Projekt definiert. Anwendung A1 untersucht Ticket-Verkäufe in Schlössern und anderen Denkmälern der Tschechischen Region Südböhmen. Anwendung A2 beschäftigt sich mit Trinkwasserversorgung und -verbrauch in einer westböhmischen Region nahe der Stadt Sokolov, CZ. In diesem Beitrag wird Anwendung A1 als Fallstudie verwendet.

Der Rest dieses Beitrages gliedert sich wie folgt: Abschnitt 2 stellt relevante Arbeiten im Bereich grafischer Modellierung und Metadaten-Management vor und grenzt die Modellierungsebenen bzw. -phasen voneinander ab. In Abschnitt 3 wird der Kern der Arbeit, die Modellierungssprache ADAPTEd UML, vorgestellt. Abschnitt 4 schließlich schließt den Beitrag mit einem Architekturansatz zum Metadaten-gestützten Data-Warehouse-Entwurf und einem Ausblick ab.

2 Multidimensionale Modellierung

2.1 Relevante Arbeiten

Im Bereich der Modellierung operativer Datenbanksysteme hat sich das Entity-Relationship-Modell von Chen [Chen76] als Standard durchgesetzt. Die Eignung der ER-Notation zur Darstellung multidimensionaler Modelle wird intensiv diskutiert. Das ER-Modell unterstützt keine semantische Unterscheidung zwischen qualifizierenden und quantifizierenden Entities. Schwierigkeiten gibt es z.B. bei der Abbildung von Dimensionshierarchien. Inzwischen mehren sich daher die Stimmen, die die Notwendigkeit einer speziellen multidimensionalen Notation zur

Darstellung von OLAP-Systemen sehen. Neben Erweiterungen des ER-Modells finden sich auch völlig neuartige Notationen (z.B. das Dimensional Fact Modeling [GMR98]) und UML-basierte Ansätze (z.B. [Toto00]). Leider ist noch kein Standard in Sicht.

Das multidimensionale Entity-Relationship-Modell (ME/R) ist innerhalb der Forschungsgruppe Wissensbasen des bayrischen Forschungszentrums für wissensbasierte Systeme (FORWISS) im Rahmen des Projektes „System 42“ entstanden [SBHD98]. Im Teilprojekt „Babel Fish“ wird eine Methodik zum modellgestützten Entwurf und Betrieb von Repository-getriebenen Data-Warehouse-Systemen entwickelt. Für die semantische (konzeptuelle) Modellierung wird hierbei die ER-Notation um drei neue Elemente (Fact Relation, Roll-up Relation und Hierarchieebene) erweitert.

Einen zweiten Ansatz zur multidimensionalen Datenmodellierung stellt das aus der Consulting-Praxis stammende, von Dan Bulos (Symmetry Corp.) entwickelte Application Design for Analytical Processing (ADAPT) dar. ADAPT wurde erstmalig in der Fachzeitschrift Database Programming & Design [Bulo96] publiziert und wird in [BuFo98] genauer beschrieben. ADAPT hat inzwischen aber auch diverse Reaktionen in wissenschaftlichen Veröffentlichungen hervorgerufen.

Auch im Rahmen der Projekte ODAWA (OFFIS Tools for Data Warehousing) des Oldenburger Forschungs- und Entwicklungsinstitutes für Informatik-Werkzeuge und -Systeme und MetaMIS der Universität Münster werden Designmethodiken für Data-Warehouse-Systeme entwickelt. Ziel ist es jeweils (wie auch bei „Babel Fish“), die Modellierung und Implementation zentral zu koppeln und so die Wartung und Weiterentwicklung zu vereinfachen.

2.2 Modellierungsphasen

Die Erstellung eines konzeptuellen Modells ist eine systemunabhängige Aktivität (vgl. z.B. [EINa00]), d.h. für diese Phase ist es vollkommen unerheblich, welche Datenbank- (oder OLAP-) Software zum Einsatz kommt. Das logische Design hingegen besteht aus der Transformation des konzeptuellen Datenmodells in ein logisches Zielmodell, das sogenannte Implementationsmodell. Dieses ist vom Datenmodell der ausgewählten Software beeinflusst.

Wie stellt sich nun diese Trennung in der multidimensionalen OLAP-Welt dar? Die meisten Veröffentlichungen siedeln das multidimensionale Modell nur auf konzeptueller Ebene an (z.B. [Toto00]), die logische Modellierung wird auf die relationale Modellierung für ROLAP-Zielsysteme (Star-/Snowflake-Schemata) beschränkt. In diesem Beitrag wird eine etwas andere Meinung vertreten. Das multidimensionale Modell ist unabhängig von der Art der Datenspeicherung auf allen drei Ebenen anzuwenden.

Da die konzeptuelle Modellierung systemunabhängig erfolgt, steht die Anwendbarkeit des multidimensionalen Modells auf dieser Ebene außer Frage. Bei MOLAP-Systemen ist die Notwendigkeit eines logischen multidimensionalen Modells sicher auch unumstritten, da hier die Fähigkeiten des verwendeten MOLAP-Werkzeuges (z.B. im Hinblick auf komplexe Dimensionen) berücksichtigt werden müssen. Zur Notation logischer Modelle sind daher sicherlich Konstrukte zur expliziten Darstellung von Dimensionen und Hierarchien notwendig. Ein logisches multidimensionales Modell ist nach Meinung der Autoren allerdings auch in ROLAP-Systemen angebracht, wenn auch nur in Form von Metadaten, die dazu dienen, eine virtuelle multidimensionale Sicht für die relational gespeicherten Daten zu erzeugen. Dieses „Implementationsmodell“ kann schließlich ebenfalls vom konzeptuellen (systemunabhängigen) Modell abweichen, da auch hier auf die Fähigkeiten des verwendeten ROLAP-Produktes Rücksicht genommen werden muss. Die relationale Modellierung erfolgt bei ROLAP-Systemen auf logischer und physischer Ebene parallel zum multidimensionalen Modell.

3 ADAPTed UML

Als Sprache (grafische Notation) zur Modellierung komplexer Softwaresysteme hat sich die Unified Modeling Language (UML, vgl. [OMG00b]) als Defacto-Standard durchgesetzt. Sie bietet eine klare Fundierung durch das UML-Metamodell und eine breite Unterstützung durch Modellierungswerkzeuge. In ihrer ursprünglichen Form ist UML zur Modellierung von objektorientierten Softwaresystemen ausgelegt. UML wird aber mehr und mehr auch in anderen Bereichen der Softwareentwicklung, z.B. der Datenmodellierung, eingesetzt.

Betrachtet man allerdings die Relevanz UML-basierter Notationen im Data-Warehouse-Bereich, so muss man eingestehen, dass diese Notationsformen in der Praxis bisher keinerlei Rolle spielen. Wenn eine konzeptuelle Modellierung überhaupt stattfindet, so wird i.d.R. eine ER-ähnliche intuitive Notation gewählt. Die einzige Ausnahme scheint die ADAPT-Notation darzustellen, die zumindest im Oracle-Express-Umfeld Anerkennung findet. Bei ADAPT fehlt allerdings eine formale Fundierung, und die Semantik ist teilweise unklar.

Im Folgenden wird versucht, diese Ungereimtheiten zu eliminieren, indem eine an ADAPT angelehnte UML-basierte Notation namens „ADAPTed UML“ entwickelt wird. Wie alle UML-Erweiterungen, macht ADAPTed UML extensiven Gebrauch von Stereotypen. Um die Verständlichkeit der Diagramme zu vereinfachen, werden statt der (besonders für Endanwender) verwirrenden textuellen Stereotypen die aus ADAPT bekannten Symbole verwendet, woraus sich der Name ergibt. Als Grundlage werden UML-Klassendiagramme verwendet. Die Kombinierbarkeit der Modellelemente wird anhand von Metamodellen eingeschränkt.

3.1 Konzeptuelles Modell

Wie bereits erwähnt, stellt ein konzeptuelles (auch semantisches) Modell ein systemunabhängiges Abbild der Realität dar. Die folgenden Basisanforderungen an ein konzeptuelles Datenmodell werden in der Literatur diskutiert (vgl. z.B. [EINa00]):

- **Ausdruckstärke:** Das Datenmodell muss mächtig genug sein, um die Unterschiede zwischen Daten- und Abhängigkeitstypen aufzuzeigen. Außerdem sollte es möglich sein, Anwendungssemantiken und -constraints zu beschreiben. Da UML als Grundlage verwendet wird, können dort vorhandene Konstrukte, wie Constraints und Bemerkungen (Notes) auch in ADAPTEd-UML-Diagrammen eingesetzt werden.
- **Einfachheit:** Das Modell sollte einfach genug sein, um von einem typischen (End-) Benutzer verwendet und verstanden zu werden. Eine grafische Repräsentation ist wünschenswert. Durch Verwendung der (z.T. in der Praxis bereits gebräuchlichen) ADAPT-Symbole zur Darstellung von Stereotypen wird die Verständlichkeit erhöht.
- **Minimalität:** Das Modell sollte nur eine kleine Anzahl an Basiskonzepten beinhalten. Die Konzepte sollten eindeutig sein, sich in Ihrer Bedeutung also nicht überschneiden. Dies spricht für einen minimalistischen Ansatz. Die Anzahl der verfügbaren Symbole sollte (wie z.B. bei ME/R) möglichst gering gehalten werden, anstatt zu versuchen alle Sonderfälle abzudecken (wie es bei ADAPT der Fall ist). Konstrukte wie explizite Dimensionen und Hierarchien werden wie bereits erwähnt nicht der konzeptuellen Ebene zugeordnet, da sie keine Realweltobjekte, sondern nur vorgedachte Navigationspfade in der zu entwickelnden OLAP-Anwendung darstellen.
- **Formalität:** Die Konzepte des Modells sollten formal definiert und korrekt sein. Daher kann ein konzeptuelles Schema als formale eindeutige Abstraktion der Realität gesehen werden. Die ADAPTEd-UML-Diagramme werden durch ein formales Metamodell beschrieben. Hierdurch sind die Diagramme eindeutig spezifiziert und eine (evtl. automatisierte) Syntaxprüfung wird möglich.

Das entwickelte Metamodell des konzeptuellen ADAPTEd-UML-Modells beinhaltet die Kernelemente Würfel (Cube), Kennzahl (Measure), Hierarchieebene (DimLevel) und Dimensionsattribut (DimAttribute). Ein Würfel besitzt null bis mehrere Kennzahlen, die die zu analysierenden Fakten darstellen. Die Hierarchieebenen sind in Halbordnungen (impliziten Dimensionen) angeordnet, wobei ein Würfel jeweils durch eine Menge von Basishierarchieebenen (Basisgranularität) aufgespannt wird.

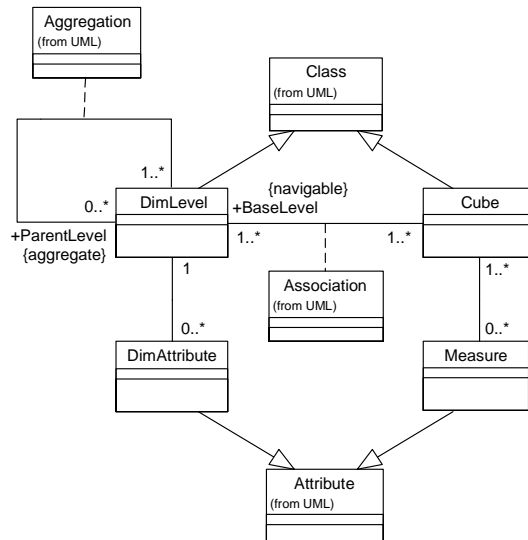


Abbildung 1: Metamodell des konzeptuellen Modells mit Notationselementen

Das in Abbildung 1 dargestellte Metamodell beinhaltet neben den beschriebenen Modellelementen Hinweise auf die grafische Repräsentation in Form von Verweisen auf UML-Kernelemente (durch „from UML“ gekennzeichnet). DimLevel und Cube sind Spezialisierungen von Class und daher als Klassen zu zeichnen. Die konkreten Klassennamen sind als Stereotypen zu verwenden (Abbildung 2 zeigt außerdem die von ADAPT übernommenen Stereotypsymbole).

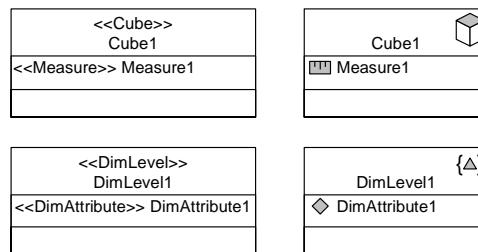


Abbildung 2: Stereotypen zur Darstellung des konzeptuellen Modells

Die Abhängigkeiten zwischen Würfel und Basishierarchieebene ist eine Association mit definierter Navigierbarkeit (Pfeilrichtung). Die Abhängigkeiten der Hierarchieebenen innerhalb einer Halbordnung werden als Aggregation dargestellt. Kennzahlen (Measure) und Dimensionsattribute (DimAttribute) werden als UML-Attribute umgesetzt. Die entsprechenden Stereotypen werden ebenfalls durch ADAPT-Symbole repräsentiert.

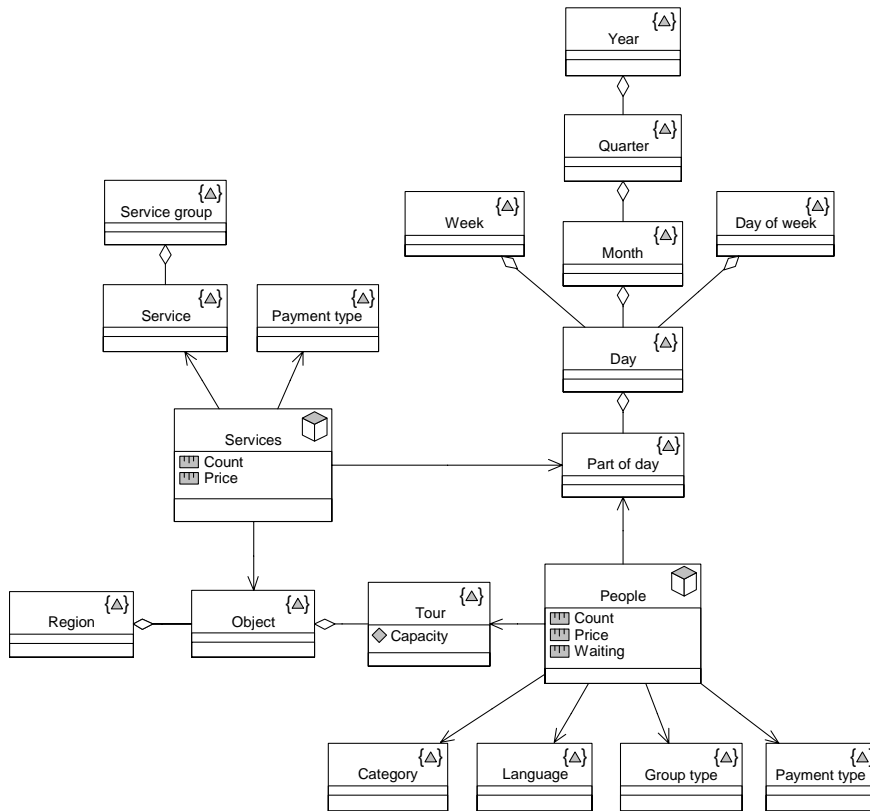


Abbildung 3: Konzeptuelles Modell des Szenarios

Abbildung 3 zeigt das Szenario (GOAL-Anwendung A1) als konzeptuelles ADAPTEd-UML-Diagramm. Es existieren zwei separate Wurfel mit unterschiedlichen Kennzahlen. Der „People“-Wurfel speichert Besucherzahlen und -preise, der „Services“-Wurfel beinhaltet Daten ber von den Besuchern erworbene Dienstleistungen. Dabei existieren die Fakten des „People“-Wurfels in der Granularitat „Tour“, wahrend die Fakten des „Services“-Wurfels weniger detailliert (auf „Objekt“-Ebene) vorliegen.

Kennzahlen werden zwar nicht wie in ADAPT als Kennzahlendimension modelliert, jedoch wird das ADAPT-Symbol fr Kennzahlendimensionen zur Kennzeichnung des Stereotyps „<<Measure>>“ verwendet. Es sei erwahnt, dass berechnete Kennzahlen nicht explizit bercksichtigt wurden, es bietet sich aber eine Darstellung als abgeleitetes Attribut (Kennzeichnung durch ein „/“ vor dem Namen) an, wobei die Berechnungsformeln durch UML-Constraints ausgedrckt werden.

Außerdem existieren keine expliziten Dimensionen und Hierarchien. In ADAPT werden Dimensionen von oben nach unten (d.h. vom niedrigsten bis zum höchsten Detaillevel) dargestellt. Die Kopplung an den Würfel erfolgt in ADAPT über das Dimensions- und Hierarchiesymbol. Im Gegensatz dazu wird hier die jeweils feinste Hierarchieebene an den Würfel gebunden, um dessen Basisgranularität anzugeben. So ist es (wie z.B. auch in ME/R) möglich, dass sich mehrere Würfel eine „Dimension“ teilen, obwohl die zugehörigen Fakten auf unterschiedlicher Granularität vorliegen.

3.2 Logisches Modell

Im Zuge des logischen Designs wird das systemunabhängige konzeptuelle Modell in ein logisches (systemabhängiges) „Implementationsmodell“ überführt. Einschränkungen und Besonderheiten des eingesetzten Systems (sprich DBMS- bzw. OLAP-Produktes) werden berücksichtigt. Im relationalen Fall operativer Datenbanken kann man prinzipiell von dem (einen) logischen Relationenmodell sprechen. Der logische Entwurf besteht in diesem Fall aus der Transformation von Entities und Beziehungen in die entsprechenden Relationen bzw. Tabellen. Im multidimensionalen Fall von OLAP ist dies anders. Die logischen Modelle verfügbarer OLAP-Systeme differieren zum Teil stark.

So gibt es beispielsweise in MicroStrategy 7 keinen Zwang zur expliziten Modellierung von Dimensionen und Hierarchien, und man ist relativ frei bei der Definition von Abhängigkeiten zwischen den Hierarchieebenen (nach MicroStrategy-Terminologie Attribute genannt). Das beschriebene konzeptuelle Modell wird insofern also mehr oder weniger logisch direkt umgesetzt. In anderen Systemen (z.B. Microsoft OLAP/Analysis Services oder Cognos PowerPlay) werden dagegen explizite Dimensionen und Hierarchien mit gewissen daraus resultierenden Einschränkungen modelliert. Die Systeme unterscheiden sich z.B. in der Weise, ob Hierarchieebenen in verschiedenen (überlappenden) Hierarchien oder gar Dimensionen vorkommen können, oder ob mehrere Würfel in einem Modell unterstützt werden. Einige Systeme unterstützen keine Dimensionsattribute. Evtl. notwendige Transformationen (z.B. die Auflösung überlappender Hierarchien durch Duplizieren von Hierarchieebenen) werden auch in [HaSB00] vorgenommen.

Das in diesem Beitrag verwendete logische Metamodell orientiert sich an den Microsoft OLAP/Analysis Services. Zu den bereits im konzeptuellen Fall verwendeten Elementen kommen zusätzlich Dimensionen (Dimension) und Hierarchien (DimHierarchy). Dimensionen können von mehreren Würfeln verwendet werden (sog. „shared dimensions“). Jede Hierarchie gehört jedoch zu genau einer Dimension und jede Hierarchieebene zu genau einer Hierarchie. Überlappungen erlaubt Microsoft in diesem Bereich nicht.

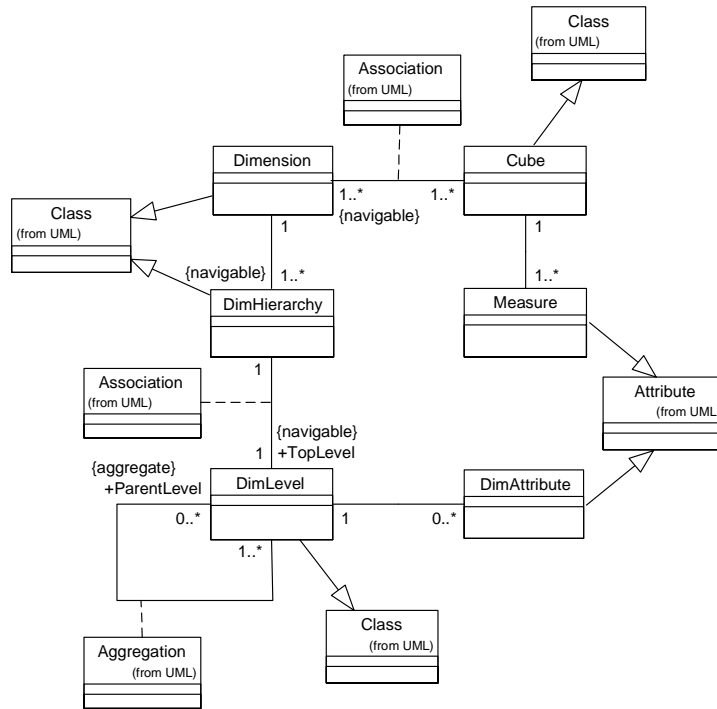


Abbildung 4: Metamodell des logischen Modells mit Notationselementen

Abbildung 4 zeigt das Metamodell des verwendeten logischen Datenmodells. Abgesehen von den Hinweisen auf die grafische Repräsentation ähnelt das Metamodell sehr dem OLP-Teilmodell für OLAP-Schemata im Open Information Model (OIM) der Metadata Coalition (vgl. Abschnitt 4), das in etwa dem physischen multidimensionalen Modell von Microsoft entspricht. Abbildung 5 zeigt die für die Elemente Dimension und DimHierarchy verwendeten Stereotypen.

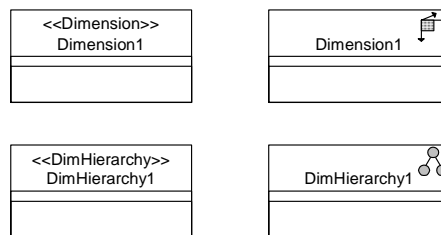


Abbildung 5: Zusätzliche Stereotypen zur Darstellung des logischen Modells

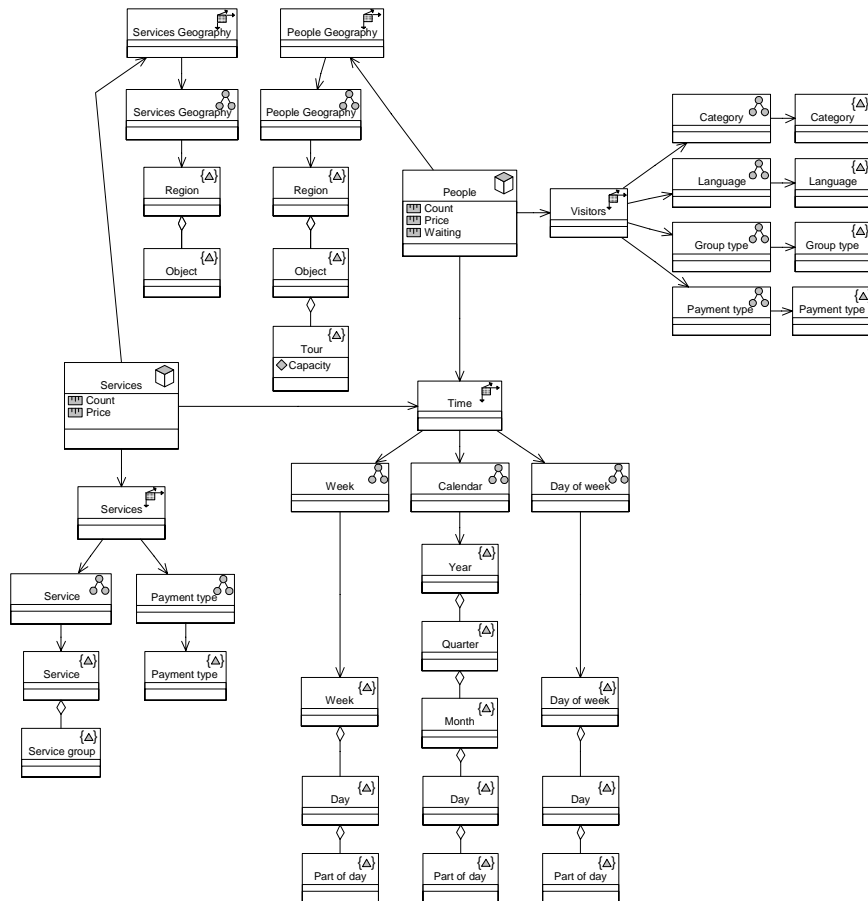


Abbildung 6: Logisches Modell des Szenarios

Abbildung 6 zeigt das logische Modell des Szenarios (d.h. eine mögliche Umsetzung des konzeptuellen Modells mit Hilfe der Microsoft OLAP/Analysis Services). Die überlappenden Dimensionen bzw. Hierarchien wurden durch Duplizierung der Hierarchieebenen aufgelöst. Außerdem wurden Dimensionen und Hierarchien explizit in das Modell aufgenommen und mit einem eindeutigen Namen versehen.

Im Gegensatz zum konzeptuellen Modell erfolgt die Kopplung der Hierarchien an den Würfel über die Dimensions- und Hierarchieklassen an der größten Hierarchieebene der jeweiligen Hierarchie. Dies entspricht der Vorgehensweise bei ADAPT und ist unproblematisch, da beispielsweise unterschiedliche Basisgranularitäten sowieso in Form von mehreren (unabhängigen) Dimensionen bzw. Hierarchien modelliert werden müssen.

3.3 Dimensionsbäume

Das multidimensionale Schema wird auf der jeweiligen Modellierungsebene durch die vorgestellten Diagramme dargestellt. Auch wenn im klassischen Datenbankdesign bewusst nur diese Schemaebene betrachtet wird, so ist es in OLAP-Systemen durchaus angebracht, die Instanzen weitestgehend statischer Dimensionen (sog. „slow changing dimensions“) schon zum Designzeitpunkt in Augenschein zu nehmen, um die Verdichtungs- bzw. Aggregationspfade zu dokumentieren. Diese Notwendigkeit wird beispielsweise auch in ADAPT berücksichtigt, nur dass hier durch fehlende Notationsregeln z.T. eine Vermischung von Schema- und Instanzdaten erfolgt.

Als Erweiterung von ADAPTEd UML bietet sich ein Diagrammtyp zur Darstellung von Dimensionsbäumen an, in dem Hierarchieelemente dargestellt werden, die als Instanzen einer Hierarchieebene anzusehen sind. Dementsprechend sind sie als Objektinstanzen der entsprechenden DimLevel-Klassen notiert und mit einem „<<DimMember>>“ Stereotyp versehen. Da es sich um konkrete Dimensionselemente handelt, besitzen evtl. Dimensionsattribute konkrete Werte, die der UML-Notation entsprechend als Initialisierungswert dargestellt werden können.

Dimensionsbäume geben Verdichtungspfade in den Daten an, die als gerichtete Associations notiert werden. Sie können wie das Datenmodell auf konzeptueller und logischer Ebene betrachtet werden. Auf logischer Ebene macht es Sinn, Dimensionsbäume für explizit modellierte Hierarchien anzugeben. Auf konzeptueller Ebene gibt es keine solchen expliziten Hierarchien; Ausgangspunkt ist hier jeweils eine Hierarchieebene (DimLevel) mit maximaler Granularität, also ein „Blatt“ im konzeptuellen ADAPTEd-UML-Modell. Abbildung 7 zeigt einen Ausschnitt der Geographiehierarchie des Szenarios.

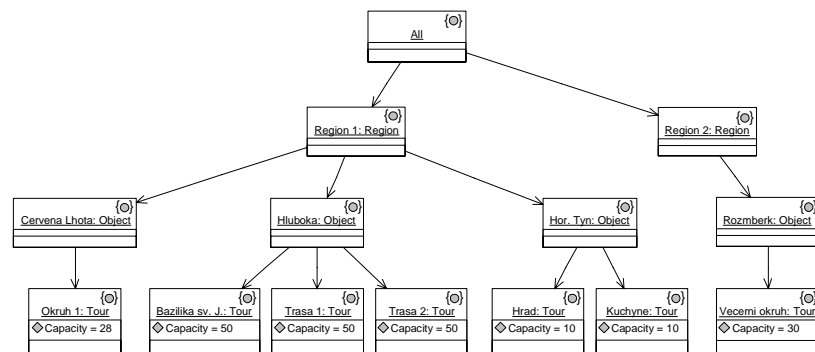


Abbildung 7: Beispiel eines Dimensionsbaums

4 Architekturansatz und Ausblick

Die in Abschnitt 2 beschriebenen Projekte zeigen die Relevanz von Metadaten- und Repository-gestützten Ansätzen zum Data-Warehouse-Entwurf. Es ist sinnvoll die Datenmodelle auf den unterschiedlichen Ebenen (konzeptuell, logisch, physisch) nahe beieinander, also möglichst zentral, zu speichern, um einen reibungslosen Übergang zwischen den Ebenen (eben den Entwurfsprozess) zu unterstützen.

Betrachtet man die Metadaten-Problematik in der Praxis, so besitzen zwei Ansätze Aussicht auf Erfolg: Das Common Warehouse Metamodel (CWM) der Object Management Group (unterstützt u.a. von IBM, Oracle, Unisys) [OMG00a] und ein Ansatz der Metadata Coalition (unter der Federführung von Microsoft), das Open Information Model (OIM) [MDC99]³. Das OIM besteht aus mehreren Teilmodellen. Für den Data-Warehouse-Entwurf ist insbesondere das „OLAP Schema“ (OLP-Teilmodell) von Bedeutung. Das Modell besitzt außerdem insofern gute Chancen sich weiter durchzusetzen, als es auch für „OLE DB for OLAP“ als Grundlage dient. Diese von Microsoft eingeführte Schnittstelle zum Zugriff auf OLAP-Daten wird inzwischen auch von immer mehr Drittanbietern unterstützt.

Die Analysis Services (ehemals OLAP Services) des Microsoft SQL Server 2000 können eine Erweiterung des OLP-Teilmodells zu nutzen, um Metadaten direkt OIM-konform im Microsoft Repository abzulegen. Andere Hersteller werden diesem Beispiel möglicherweise folgen. Das UML-Teilmodell des OIM dient zur Speicherung von UML-Modellen. Wiederum unterstützen einige UML-Modellierungswerkzeuge, wie z.B. Rational Rose, bereits die Speicherung von UML-Modellen im Repository. Es liegt also nahe, das Repository als Medium zu nutzen, um (konzeptuelle bzw. logische) ADAPTEd-UML-Modelle mit dem durch das OLAP-System verwendeten physischen Modell abzugleichen.

Abbildung 8 zeigt einen Gesamtüberblick über einen integrierten Architekturansatz von der Modellierung über die Implementierung/Wartung bis zur Anwendung und Dokumentation. Der gesamte Vorgang kann durch entsprechende Werkzeuge unterstützt werden.

³ Inzwischen haben die Metadata Coalition (MDC) und die OMG verkündet, dass die beiden bisher konkurrierenden Metamodellansätze verschmelzen sollen. Die MDC wird in der OMG aufgehen, Details sind aber noch nicht bekannt.

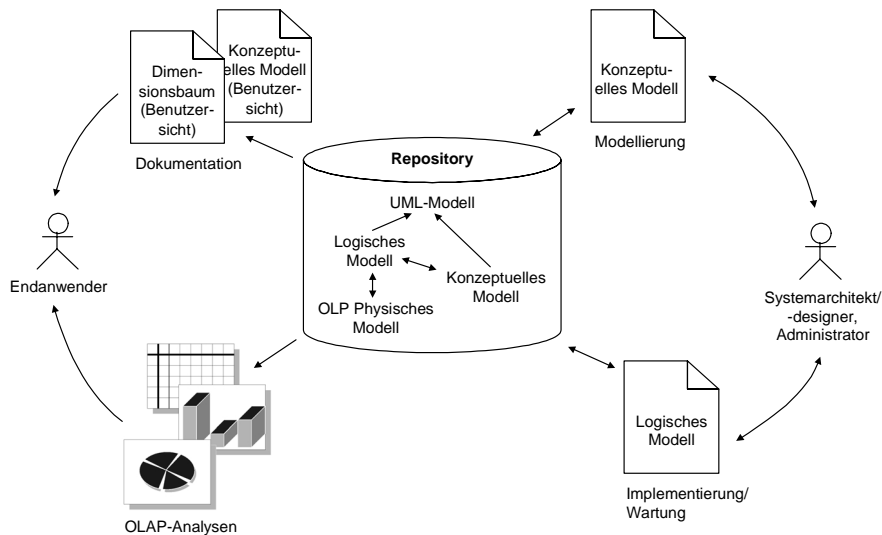


Abbildung 8: Metadatenfluß und -zugriff mit Hilfe von ADAPTEd-UML

Im Zusammenhang mit der in diesem Beitrag präsentierten Arbeit steht auch die Entwicklung einer Methodologie zur Modellierung von Sicherheitssemantiken für OLAP-Systeme (vgl. auch [PrPe00]). So gilt es im Rahmen des GOAL-Projektes, Sicherheitsaspekte zu untersuchen und eine Autorisierungs- und Zugriffskontrollkomponente zu entwickeln. Unterschiedliche Benutzergruppen mit eingeschränktem Zugriff auf das Data Warehouse wurden definiert.

Zur grafischen Notation solcher Zugriffsregeln (z.B. der Manager des Objektes Hluboka hat nur Zugriff auf Daten des dieses Objekt betreffenden Teilwürfels) lassen sich UML-Constraints nutzen. Zur Beschreibung der Policies wird zusätzlich eine entsprechende Constraintsprache, eine „Multidimensional Security Constraint Language“ (MDSCL), benötigt. Außerdem lassen sich (z.B. zu Dokumentationszwecken) unterschiedliche Benutzersichten durch modifizierte (d.h. zum Teil ausgeblendete) ADAPTEd-UML-Diagramme darstellen.

Im Rahmen zukünftiger Arbeiten ist es geplant, diese Vorgehensweise zur Modellierung von OLAP-Sicherheitssemantiken auszubauen und in das Gesamtkonzept zu integrieren.

Literatur

- [BaGü01] Bauer, A., Günzel, H. (Eds.): Data-Warehouse-Systeme – Architektur, Entwicklung und Anwendung. dpunkt Verlag; Heidelberg, 2001.

- [Chen76] Chen, P.P.-S.: The Entity Relationship Model – Towards a Unified View of Data. ACM TODS Vol. 1, No.1, 1976.
- [Bulo96] Bulos, D.: A New Dimension. In Database Programming & Design; 6/1996; nachgedruckt in Chamoni, P, Gluchowski, P. (Hrsg.): Analytische Informationssysteme. Springer; Berlin et al., 1998.
- [BuFo98] Bulos, D., Forsman, S.: Getting Started with ADAPT. Whitepaper, Symmetry Corp.; San Rafael, 1998.
- [EJNW96] Eicker, S., Jung, R., Nietsch, M., Winter, R.: Entwicklung eines Data Warehouse für das Produktionscontrolling: Konzepte und Erfahrungen. In Becker, J., Grob, H.L., Müller-Funk, T., Vossen, G. (Hrsg.): Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 51; Universität Münster, November 1996.
- [ElNa00] Elmasri, R.; Navathe, S. B.: Fundamentals of Database Systems; Addison-Wesley Longman, Inc.; 3 Auflage; 2000.
- [GMR98] Golfarelli, M., Maio, M., Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schemes. In Proceedings of the Hawaii International Conference on System Sciences; Kona, Hawaii, January 6-9, 1998.
- [HaSB00] Hahn, K., Sapia, C., Blaschka, M.: Automatically Generating OLAP Schemata fom Conceptual Graphical Models. Proc. Third ACM International Workshop on Data Warehousing and OLAP (DOLAP 2000), McLean, VA, U.S.A., November 2000.
- [Inmo96] Inmon, W.H.: Building the Data Warehouse. Wiley & Sons; New York et al., 1996.
- [MDC99] Meta Data Coalition: Open Information Model, Version 1.1 (Proposal); August, 1999.
- [OMG00a] Object Management Group: Common Warehouse Metamodel (CWM) Specification. Proposal to the OMG ADTF RFP: Common Warehouse Metadata Interchange (CWMI); February, 2000.
- [OMG00b] Object Management Group: OMG Unified Modeling Language Specification. Version 1.3; March 2000.
- [PrPe00] Priebe, T., Pernul, G.: Towards OLAP Security Design – Survey and Research Issues. Proc. Third ACM International Workshop on Data Warehousing and OLAP (DOLAP 2000), McLean, VA, U.S.A., November 2000.
- [SBHD98] Sapia, C., Blaschka, M., Höfling, G., Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. In Kambayashi, Y. et. al. (Hrsg.), Advances in Database Technologies; LNCS Vol. 1552; Springer, 1999.
- [Toto00] Totok, A.: Modellierung von OLAP- und Data-Warehouse-Systemen. Dt. Univ.-Verl. (Gabler); Wiesbaden, 2000; Zugl.: Diss.; Techn. Univ. Braunschweig, 1999.