Grundlagen des Data Warehousing

Modellierung und Architektur

M. Böhnlein, A. Ulbrich-vom Ende
Lehrstuhl für Wirtschaftsinformatik, Universität Bamberg
Feldkirchenstr. 21, D-96045 Bamberg
E-Mail: {achim.ulbrich | michael.boehnlein}@sowi.uni-bamberg.de

Abstract

In den letzten Jahren sind Data Warehouse-Systeme zu einem essentiellen Bestandteil moderner Entscheidungsunterstützungssysteme geworden. Laut einer Analyse der Meta Group wurden 1996 4.5 Milliarden US-Dollar für Data Warehouse-Projekte ausgegeben und für das Jahr 2000 wird ein Umsatz von 15 Milliarden US-Dollar prognostiziert. Dieses Wachstum ist auf die grundlegende Funktionalität dieser Systeme zurückzuführen. Data Warehouse-Systeme ermöglichen einen effizienten Zugriff auf integrierte und historische Informationen aus unterschiedlichen, teilweise heterogenen und autonomen Informationsquellen. Der Beitrag soll einen Überblick über die Grundkonzepte bei der Modellierung und Architektur von Data Warehouse-Systemen vermitteln. Auf weiterführende und vertiefende Literatur wird an der jeweiligen Stelle explizit verwiesen. Empirische Grundlage unserer Forschungstätigkeit ist das Projekt CEUS-HB (*C*omputerbasiertes Entscheidungs*U*nterstützungs*S*ystem für die Hochschulen in Bayern). Ziel dieses Projektes ist die Entwicklung eines hierarchisch verteilten Data Warehouse-Systems für die bayerischen Hochschulen. Es wird unterstützt durch das Bayerische Ministerium für Wissenschaft, Forschung und Kunst. Das interdisziplinäre Forschungsteam besteht aus Mitgliedern des Bayerischen Staatsinstituts für Hochschulforschung (Prof. Dr. H. U. Küpper) und Mitarbeitern der Universität Bamberg (Prof. Dr. E. J. Sinz).

1 Einleitung

Der Bereich der statistischen Auswertung empirisch erhobener sozio-ökonomischer Daten, wie beispielsweise Bevölkerungs-, Wirtschafts- und Umweltstatistiken, reflektiert historisch gesehen wohl das älteste Anwendungsgebiet in der elektronischen Datenverarbeitung. Aufgrund der unterschiedlichen Anforderungen an die Datenhaltung und -verarbeitung im Bereich der statistischen Analyse von Massendaten wurde Anfang der 80er Jahre die Notwendigkeit erkannt, daß eine spezielle Datenbankunterstützung benötigt wird [Shos82]. Aus diesen Forschungsaktivitäten entwickelte sich der Forschungsbereich der *statistischen und wissenschaftlichen Datenbanksysteme* ('*Statistical and Scientific Database Management Systems*'; SSDBMS).

Anfang der 90er Jahre kam die Nachfrage nach Werkzeugen für Entscheidungsträger in Unternehmen auf, um gezielt auf relevante Informationen aus dem stetig anwachsenden Volumen maschinell verarbeiteter Daten in Unternehmen zuzugreifen. Diese Systeme sollen Führungspersonen und Entscheidungsträgern bei Planungs- und Entscheidungsprozessen unterstützen [GPQ+97]. Eine Vielzahl von Akronymen, wie z.B DSS (*Decision Support System*) und EIS (*Executive Information System*) haben sich für diese Systeme im Laufe der Zeit herausgebildet.[1] Seit einiger Zeit treten in diesem Bereich verstärkt die Begriffe *Data Warehouse* im Sinne eines

Datenpool zur Bereitstellung von konsolidierten, historischen und konsistenten Informationen und *Online Analytical Processing* (OLAP) als Bezeichnung für das multidimensionale Analysekonzept in den Vordergrund. Bei genauerer Betrachtung dieser Konzepte fällt eine starke Übereinstimmung der Eigenschaften zwischen den Data Warehouse-Konzepten und den statistischen und wissenschaftlichen Datenbanksystemen auf [Shos97]. Im weiteren Verlauf werden jedoch ausschließlich die Themen OLAP und Data Warehousing behandelt.

Nach der Abgrenzung der Verarbeitungskonzepte OLTP und OLAP in Abschnitt 2 folgt eine Einführung in Struktur- und Operationsteil multidimensionaler Datenstrukturen (Abschnitt 3). In Abschnitt 4 werden die vorherrschenden Modellierungstechniken für multidimensionale Datenstrukturen in konzeptuelle, logische und physische Entwurfsebenen des Datenbankdesigns eingeordnet. Anschließend wird in die konzeptuelle multidimensionale Modellierung am Beispiel von ADAPT und in logische multidimensionale Modellierung am Beispiel des SAP BW eingeführt. Anhand der Beschreibung fachlicher und technischer Probleme beim Einsatz multidimensionaler Analysewerkzeuge auf der Basis von Rohdaten der operativen Systemen wird in Abschnitt 5 die Evolution von Data Warehouse-Systemen herausgearbeitet. In Abschnitt 6 werden die einzelnen Bestandteile der Data Warehouse-System Architektur erläutert. Drängende Probleme und aktueller Forschungsbedarf werden in Abschnitt 7 aufgezeigt.

2 Abgrenzung der Verarbeitungskonzepte OLTP und OLAP

Betriebliche Anwendungssysteme sind als Aufgabenträger für den automatisierten Teil des betrieblichen Informationssystems konzipiert [FeSi98]. Zu ihnen zählen neben Planungs- und Kontrollsystemen, insbesondere Administrations- und Dispositionssysteme [Mert95], die den täglichen Geschäftsablauf von Unternehmen unterstützen, wie z.B. Auftragserfassungs-, Lagerverwaltungs- und Buchführungssysteme. Sie werden häufig auch als operative Systeme bezeichnet und unterliegen dem Verarbeitungskonzept *On-line Transaction Processing* (OLTP). Dabei werden aktuelle Daten zu laufenden Geschäftsvorfällen anwendungsbezogen hinterlegt (vgl. Abbildung 1). Anwender sind vor allem Sachbearbeiter, die interaktiv in vorhersehbaren und repetitiven Zeitintervallen sowohl lesende als auch schreibende Zugriffe auf betriebliche Einzeldaten transaktionsgeschützt vornehmen. Häufige Transaktionen von kurzer Dauer herrschen während des Tagesgeschäfts vor. Als zentrale Datenstruktur im OLTP-Bereich gilt eine zweidimensionale Darstellung in Form von Tabellen.

Hingegen lassen sich Entscheidungsunterstützungssysteme durch ihr inhärentes Verarbeitungskonzept *On-line Analytical Processing* (OLAP) klar von operativen Systemen abgrenzen. Das OLAP-Konzept wurde von Codd [CoCS93], dem Vater relationaler Datenbanksysteme, 1993 vorgestellt und durch 12 bzw. später 18 Regeln präzisiert. Da die OLAP-Regeln wegen ihrer unterschiedlichen Auslegbarkeit höchst umstritten sind [JaGK96], wird an dieser Stelle nicht auf sie zurückgegriffen.

1. Eine Einordnung, sowie eine Abgrenzung der verschiedene Akronyme wird u.a. in [Mert95] und [GlGC97] vorgenommen.

Merkmal	**OLTP**	**OLAP**
Anwendungsbereich	Operative Systeme (Administrations- und Dispositionssysteme)	Entscheidungsunterstützungs- bzw. Data Warehouse-Systeme
Nutzer	Sachbearbeiter	Entscheidungs- und Führungskräfte
Datenstruktur	zweidimensional, anwendungsbezogen	multidimensional, subjektbezogen
Dateninhalt	detaillierte, nicht verdichtete Einzeldaten	verdichtete und abgeleitete Daten
Datenverwaltungsziele	transaktionale Konsistenzerhaltung	zeitbasierte Versionierung
Datenaktualität	aktuelle Geschäftsdaten	historische Verlaufsdaten
Datenaktualisierung	durch laufende Geschäftsvorfälle	periodische Datenaktualisierung (Snapshot)
Zugriffsform	lesen/schreiben/löschen	lesen/verdichten
Zugriffsmuster	vorhersehbar, repetitiv	ad hoc, heuristisch
Zugriffshäufigkeit	hoch	mittel bis niedrig
Antwortzeit	kurz (Sekundenbruchteile)	mittel bis lang (Sekunden bis Minuten)
Transaktionsart und Dauer	kurze Lese und Schreiboperationen	lange Lesetransaktionen

Abb. 1: OLTP vs. OLAP

Vielmehr wird das Akronym FASMI (*Fast Analysis of Shared Multidimensional Information*) nach Pendse und Creeth zur Beschreibung des OLAP-Konzeptes herangezogen [PeCr95]. Entscheidungs- und Führungskräften soll schneller (*Fast*) analytischer (*Analysis*) Zugriff im Mehrbenutzerbetrieb (*Shared*) auf kontextrelevante multidimensionale betriebliche Informationen (*Multidimensional Information*) ermöglicht werden. Die Daten werden subjektbezogen und verdichtet unter spezieller Berücksichtigung historischer Verlaufsdaten vorgehalten. Datenaktualisierungen finden nur periodisch durch Abzüge (*Snaphots*) operativer Systeme statt. Analyserelevante Zugriffe auf OLAP-Daten durch Führungskräfte geschehen ad hoc und eher selten. Dabei wird meist lesend auf die verfügbaren Daten zugegriffen, wobei aufgrund der Historisierung mit einer mittleren bis langen Antwortzeit gerechnet werden muß. Ziel der Datenverwaltung in OLAP-Systemen ist die Versionierung von Daten über die Zeit hinweg. Zentrales Unterscheidungsmerkmal zwischen OLTP- und OLAP-Systemen ist hierbei die Verwendung multidimensionaler Datenstrukturen, die der natürlichen Denkweise menschlicher Entscheidungsträger näher kommt als flache tabellarische Strukturen. Sie werden im folgenden Abschnitt genauer vorgestellt.

3 Einführung in die Modellierung multidimensionaler Datenstrukturen

In OLAP-Systemen bilden multidimensionale Datenstrukturen die Grundlage der Modellierung. Sie sind durch Strukturbeschreibungen und generische Operatoren charakterisierbar. Im Rahmen des Strukturteils multidimensionaler Datenstrukturen werden im folgenden wesentliche Beschreibungselemente, deren Beziehungen und Semantik eingehend erläutert. Auf Navigationsmöglichkeiten in mehrdimensionalen Datenräumen geht der Operationsteil ein. Während Abschnitt 3 die Begriffswelt multidimensionaler Datenstrukturen anhand eines Anwendungsbeispiels aus dem universitären Umfeld unabhängig von konkreten Modellierungsansätzen einführt, werden in Abschnitt 4 verfügbare Modellierungsmethoden für Data Warehouse-Systeme systematisiert.

3.1 Strukturteil

Als Grundidee multidimensionaler Datenstrukturen fungiert die Unterscheidung in qualitative und quantitative Daten [Shos82]. Quantitative Daten, z.B. die Anzahl der Studierenden einer Universität, sollen unter verschiedenen Blickwinkel (Aspekten), wie z.B. Studienausrichtung und Studienabschnitt, betrachtet werden. Die resultierende Datenstruktur bildet einen mehrdimensionalen Datenraum, einen sog. Hyperwürfel (*Hypercube*) (vgl. Abbildung 2) [Pilot98]. Im Inneren des Datenwürfels stehen die quantitativen Daten, die auch als Maßzahlen (*Measures*), Variablen oder Kennzahlen bezeichnet werden. Dabei kann es sich um Basisgrößen (atomare Werte) oder abgeleitete Zahlen (berechnete Werte) handeln.

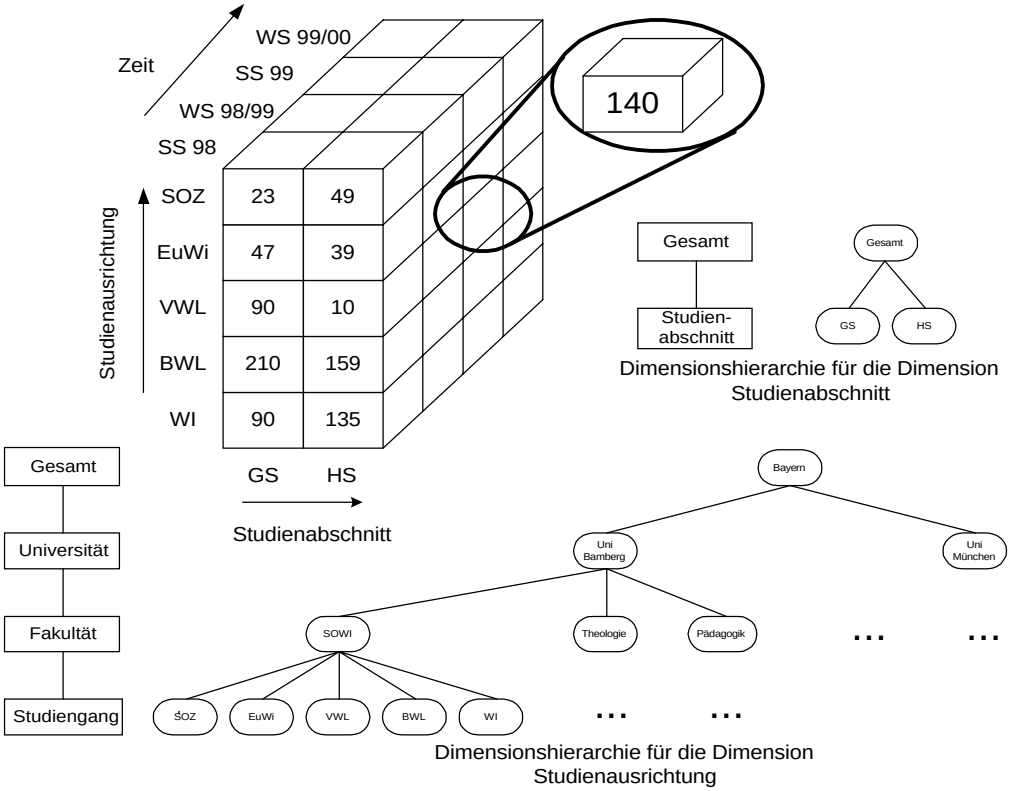

Abb. 2: Strukturteil multidimensionaler Datenstrukturen

Die Dimensionen (Achsen) des Würfels bilden die unterschiedlichen qualitative Gesichtspunkte ab. Eine Visualisierung erfolgt im zweidimensionalen Fall als Tabelle (Matrix), bei drei Dimensionen als Würfel und vierdimensional als sog. Tesseract. Dimensionen enthalten jeweils eine Anzahl von Dimensionselementen, z.B. umfaßt die Dimension Studienausrichtung u.a. die Elemente SOZ (Soziologie), EuWi (Europäische Wirtschaft), VWL (Volkswirtschaftslehre), BWL (Betriebswirtschaftslehre) und WI (Wirtschaftsinformatik) und die Dimension Studienabschnitt die Elemente Grundstudium (GS) und Hauptstudium (HS). Durch das kartesische Produkt der Dimensionselemente aller an einem Würfel beteiligten Dimensionen entsteht die Gesamtzahl der Zellen des Würfels mit jeweils einem konkreten Datenwert. Weiterhin können Dimensionselemente gegebenfalls mehrstufig verdichtet werden und bilden somit Dimensionshierarchien. Innerhalb einer Dimension sind beispielsweise mehrere derartige parallele Konsolidierungspfade möglich. Die Studierendenzahlen in einzelnen Studiengängen können auf

Fakultäts- bzw. Universitätsebene aggregiert werden. Jeder Datenwürfel unterliegt spezifischen Integritätsbedingungen. Es gelten individuelle Konsolidierungsvorschriften entlang der Knoten in Dimensionshierarchien, z.B. werden Studierendenzahlen der einzelnen Studiengänge zu Zahlen auf Fakultätsebene addiert. Es können dabei beliebig komplexe Berechnungsregeln hinterlegt sein.

3.2 Operationen auf multidimensionalen Datenstrukturen

Lediglich im Hinblick auf Datenmanipulationsoperatoren herrscht trotz enormer Begriffsvielfalt weitgehender inhaltlicher Konsens im Data Warehouse-Bereich. Dabei spielen insbesondere Anfrageoperationen, als Teilbereich der Datenmanipulationsoperatoren, im Data Warehouse-Umfeld zur Unterstützung des OLAP-Konzeptes eine bedeutende Rolle. Im folgenden werden die grundlegenden multidimensionalen Operatoren *Drill Down*, *Roll Up*, *Selection*, *Slice*, *Dice*, *Rotate*, *OLAP Join* und *Nest* am Beispiel aus Abbildung 3 kurz vorgestellt ([OLA95][Holt97][GaGl97b][Kena95]).

Die Operationen *Roll Up* und *Drill Down* erlauben das Durchlaufen von Verdichtungsebenen innerhalb einer Dimensionshierarchie. Beim *Drill Down* steigt man von einem bestimmten Aggregationsniveau auf die jeweils nächsttiefere und detailliertere Verdichtungsstufe. Ausgehend von den Studierendenzahlen in den Sozial- und Wirtschaftswissenschaften (SOWI) im Sommersemester 1998 in allen Studienabschnitten wird im Beispiel a) aus Abbildung 3 gleichzeitig eine *Drill Down*-Operation auf die Dimensionen Studienausrichtung und Studienabschnitt durchgeführt. Auf der darunterliegenden Verdichtungsebene werden Daten zu den einzelnen Studiengängen und Studienabschnitten ersichtlich. Die *Roll Up*-Operation stellt die Komplementäroperation zum *Drill Down* dar, d.h. sie wechselt zur jeweils höheren Verdichtungsebene innerhalb einer Dimensionshierarchie (Beispiel b)).

Der Operator *Selection* ermöglicht die Auswahl einzelner Würfeldaten, d.h. sie erfüllt eine Filterfunktionalität, z.B. finde die drei Studiengänge mit den höchsten Studierendenzahlen im Grundstudium während des Sommersemesters 1998 (BWL, VWL und WI). Dagegen stellen die Operatoren *Slice* und *Dice* Spezialfälle der *Selection* dar. Bei einem dreidimensionalen Hypercube entspricht der *Slice*-Operator dem Herausschneiden einer Scheibe aus dem Würfel. Das Ergebnis liegt in Form einer zweidimensionalen Matrix vor. In Beispiel c) wird durch Beschränkung auf das Sommersemester 98 eine zweidimensionale Matrix aller Studienabschnitte und Studiengänge abgegrenzt. Durch Beschränkung auf einzelne Dimensionselemente verschiedener Dimensionen kann die S*lice*-Operation aber auch auf beliebig dimensionierte Hypercubes verallgemeinert werden. Beispiel d) verdeutlicht die Operation *Dice*, bei der ein Teilwürfel des gesamten Hypercubes gebildet wird. Die Auswahl der Studiengänge EuWi, BWL und VWL in den Semestern WS 98/99 und SS 99 führt zu dem Unterwürfel in Beispiel d).

Die Drehung des Hypercubes um eine seiner Achsen entspricht bildlich dem *Rotate*-Operator. Somit erhält der Benutzer durch die Rotation unterschiedliche Sichten auf den betrachteten Datenwürfel. Ausgehend vom Hypercube aus Beispiel e) fokussiert sich der Blick des Benut-

zers durch Drehen um die Achse Studienausrichtung auf die Sicht Studiengänge und Semester im Grundstudium.

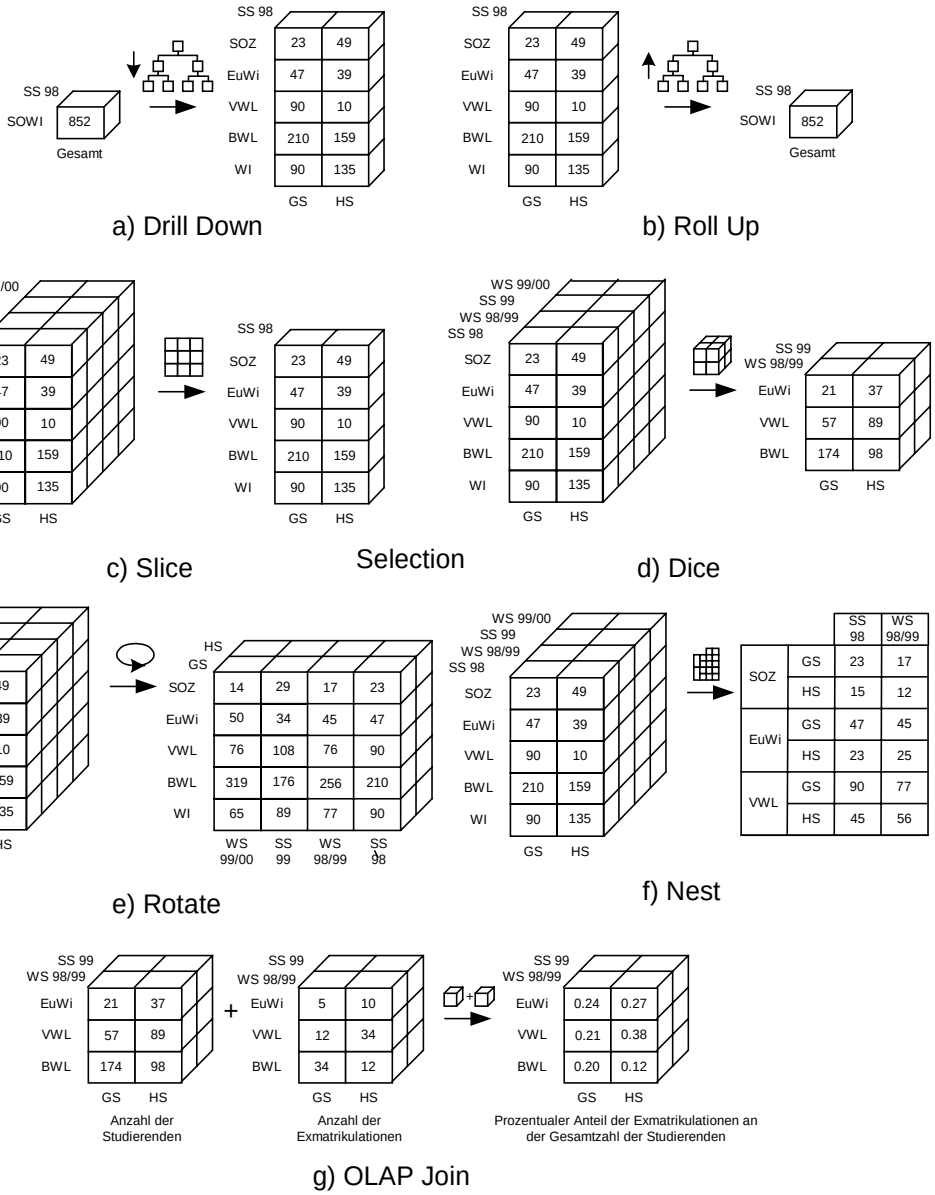

Abb. 3: Operationsteil multidimensionaler Datenstrukturen

Der *Nest*-Operator konzentriert sich auf die Darstellung einer zweidimensionalen Matrix (Kreuztabelle), bei der verschiedene Hierarchiestufen einer oder mehrerer Dimensionen auf einer Achse (Spalte oder Zeile) geschachtelt präsentiert werden. In Beispiel f) werden die Hierarchiestufen Studiengang der Dimension Studienausrichtung und Studienabschnitt der gleichnamigen Dimension in den Zeilen einer Kreuztabelle visualisiert.

Der *OLAP Join* versucht analog zur Verbund-Operation bei der Verknüpfung von Tabellen in relationalen Datenbanksystemen eine Verbindung zwischen mehreren Hypercubes herzustellen. Voraussetzung hierfür sind gemeinsame Dimensionen zwischen den zu verknüpfenden Datenwürfeln. In Beispiel f) werden die Maßzahlen *Anzahl der Studierenden* und *Anzahl der Exmatrikulationen* zweier Hypercubes zu einer neuen Maßzahl *prozentualer Anteil der Exmatrikula-*

tionen an der Gesamtzahl der Studierenden kombiniert, da die Ausgangsdatenwürfel drei gemeinsame Dimensionen besitzen.

4 Modellierungstechniken für multidimensionale Datenstrukturen

4.1 Entwurfsebenen der multidimensionalen Modellierung

Für die Modellierung multidimensionaler Datenstrukturen existiert bereits eine Vielzahl unterschiedlicher Modellierungsansätze, die überwiegend in den letzten fünf Jahren vorgeschlagen wurden. Beim Datenbankentwurf klassischer operativer OLTP-Systeme hat sich die Unterscheidung in die Entwurfsebenen des konzeptuellen, logischen und physischen Entwurfs mit den korrespondierenden Entwurfsergebnissen konzeptuelles, logisches und physisches Schema durchgesetzt ([MaDL87][Voss99]). Diese Trennung wird im folgenden auf den OLAP-Bereich übertragen und dient hier zur Systematisierung und Einordnung der vorhandenen Modellierungsansätze(vgl. Abbildung 4). Interessant erscheint dabei, daß eine relativ saubere Zuordnung der verschiedenen existierenden Modellierungsansätze zu den einzelnen Ebenen möglich ist. Ein umfassender Ansatz, der alle Ebenen durchgängig abdeckt, ist zur Zeit nicht vorhanden.

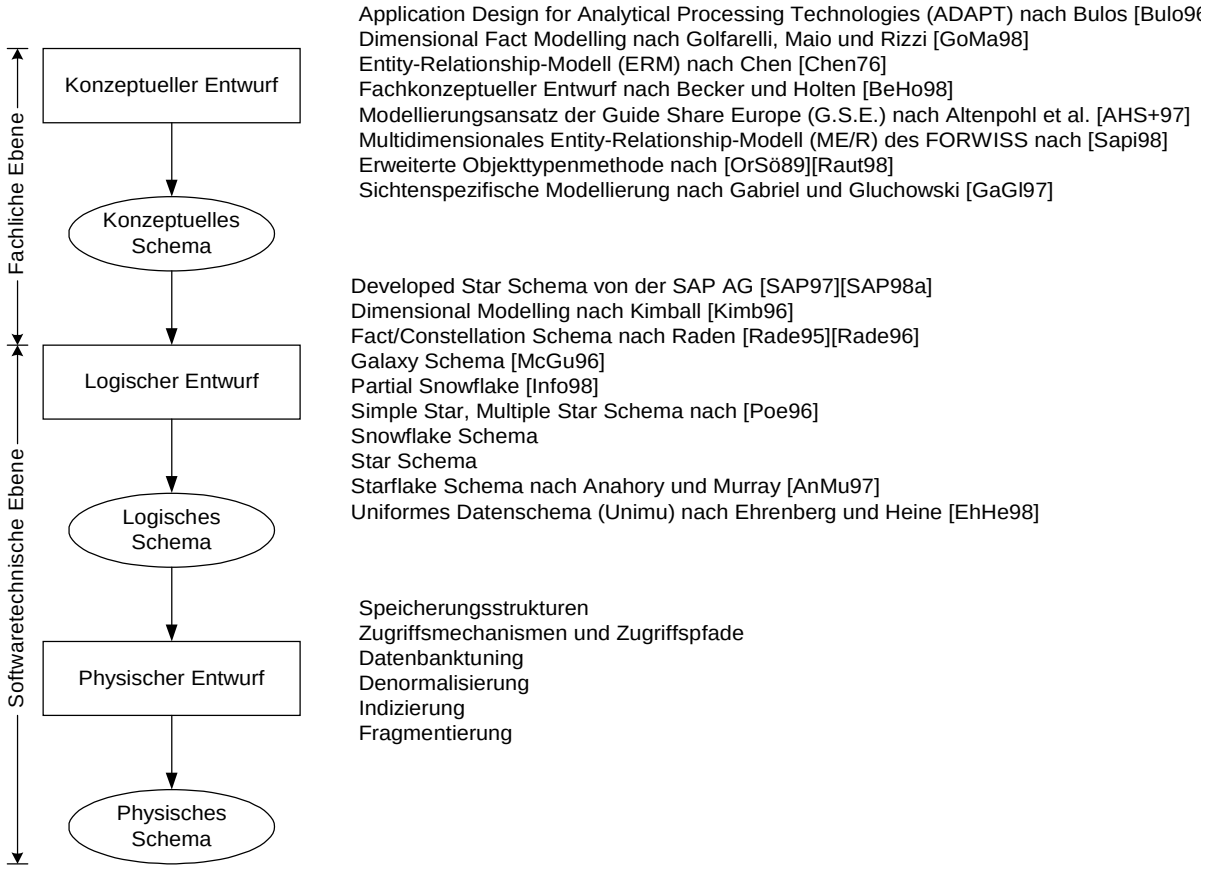

Abb. 4: Übersicht über Modellierungsmethoden für multidimensionale Datenstrukturen

Die konzeptuelle Entwurfsebene resultiert in einem konzeptuellen Schema und ist unabhängig von den speziellen Eigenschaften des einzusetzenden Zieldatenbanksystems [Voss99]. Vielmehr ist eine Ausrichtung auf die Fachtermini multidimensionaler Modellierung (siehe Abschnitt 3) zu fordern, die als Kommunikationsgrundlage zur Diskussion und Validierung des Modells gegenüber Fachvertretern dienen kann. Für die konzeptuelle Entwurfsebene hat sich bis heute noch kein dominierender Modellierungsansatz durchgesetzt.

Es lassen sich aber zumindest drei verschiedene Strömungen bei der Entwicklung von Modellierungsansätzen im konzeptuellen Bereich ausmachen.

- Es wird beispielsweise versucht, traditionelle semantische Datenmodellierungsmethoden für operative Systeme auf den Data Warehouse-Bereich zu übertragen. Hierzu zählen u.a. das klassische *Entity-Relationship-Modell* (ERM) nach Chen [Chen76] und der *Fachkonzeptuelle Entwurf* nach Becker und Holten [BeHo98].
- Weiterhin existieren Vorschläge für die Erweiterung klassischer Datenmodellierungsmethoden um multidimensionale Konstrukte. Von Rautenstrauch [Raut98] wird eine Modifikation der *Objekttypenmethode* (OTM) nach Ortner und Söllner [OrSö89] vorgeschlagen. Im Rahmen des Babel Fish Projekts des FORWISS entstand eine Erweiterung des Entity-Relationship-Modells durch das *Multidimensionale Entity-Relationship-Modell* (ME/R-Modell) [Sapi98]. Auch die Guide Share Europe hat sich um eine Anpassung des klassischen ER-Modells nach Chen bemüht [AHS+97].
- Da das Entity-Relationship-Modell und seine Varianten zwar grundsätzlich datenbankunabhängig, aber mit starken Bezug auf relationale Datenbanksysteme enwickelt wurden, beschäftigen sich weitere Forschungsaktivitäten mit Modellierungsvorschlägen, die ausschließlich auf die multidimensionale Modellierung ausgerichtet sind und keine Ursprünge in klassischen Datenmodellierungsmethoden besitzen. Dazu zählen die *sichtenspezifische Modellierung* nach Gabriel und Gluchowski [GaGl97b], *Dimensional Fact Modeling* nach Golfarelli, Maio, Rizzi [GoMR98] und *Application Design for Analytical Processing Technologies* (ADAPT) nach [Bulo96]. Auf ADAPT wird in Abschnitt 4.2 als Beispiel für einen neuartigen Datenmodellierungsansatz für Data Warehouse-Systeme noch näher eingegangen.

Während der konzeptuelle Entwurf der fachlichen Modellierung zuzurechnen ist, beziehen sich logischer und physischer Entwurf auf die softwaretechnische Modellierung. Folglich wird hier die Unabhängigkeit vom zugrundeliegenden Datenbanksystem aufgegeben. Ein konkretes Datenbankmanagementsystem erlaubt nicht den Umgang mit beliebigen Informationsstrukturen, sondern ist auf ein spezifisches Datenbankmodell und damit auf einen bestimmten Grundvorrat an Beschreibungsmitteln beschränkt. Durch die Überführung des konzeptuellen Schemas in das Datenbankmodell des einzusetzenden Datenbanksystems werden aktuelle Strukturparameter, wie z.B. Normalisierungsgrad, für den logischen Entwurf gewonnen. Einzelheiten der physischen Repräsentation der Daten spielen beim logischen Entwurf noch keine Rolle. Mit dem *Star*- bzw. *Snowflake Schema* ([Rade95][Rade96][McGu96]) und seinen Varianten wurden eine Vielzahl von logischen Modellierungsmethoden für relationale Datenbanksysteme vorgeschlagen. Diese unterscheiden sich vor allem im Grad der Normalisierung, der Berücksichti-

gung von Aggregationen, künstlichen Schlüsselattributen und Anzahl der berücksichtigten Hypercubes. Beispiele für Varianten des *Star*- bzw. *Snowflake Schemas* sind *Dimension Modelling* nach Kimball [Kimb96a], *Fact/Constellation Schema* nach [Rade95], *Galaxy Schema, Partial Snowflake* [Info98], *Simple Star* und *Multiple Star Schema* nach Poe [Poe96], *Starflake Schema* nach Anahory und Murray [AnMu97] und *Uniformisches Datenschema* (UNIMU) nach Ehrenberg und Heine [EhHe98]. Als Beispiel für eine logische Modellierungsmethode soll in Abschnitt 4.3 das *Developed Star Schema* vorgestellt werden, das im Business Information Warehouse (BW) der SAP AG eingesetzt wird.

Der physische Entwurf letztlich beschäftigt sich mit der Definiton des internen Datenbankschemas und der damit zusammenhängenden Systemparameter. Hierbei sind vor allem Leistungsgesichtspunkte bei vorgegebenen Betriebsmitteln wie Gerätekonfiguration oder Betriebssystem zu beachten. Auf der Ebene des physischen Entwurfs existieren keine konkreten Modellierungsansätze. Hier sind besondere Gestaltungsaspekte, wie z.B. Speicherungsstrukturen, Zugriffspfade, Zugriffsmechanismen, Datenbanktuning, Denormalisierung, spezielle Indizierungstechniken (Bitmap Index und Star-Index) und Fragmentierung (horizontal und vertikal) einzuordnen.

Eine Werkzeugunterstützung auf den einzelnen Entwurfsebenen vor allem im kommerziellen Bereich beschränkt sich im wesentlichen auf die logische Modellierungsebene. Zu fordern ist eine stärkere konzeptionelle Unterstützung des Modellierers. Vorgehensmodelle bei der Modellierung sind nur selten zu finden, meist recht unpräzise und beschränken sich häufig auf einen Modellierungsansatz einer Entwurfsebene. Eine ganzheitliche, geschäftsprozeßorientierte Vorgehensweise wird zwar vorgeschlagen [Kimb97], jedoch von keinem der heute existierenden Ansätze unterstützt.

4.2 Konzeptuelle multidimensionale Modellierung am Beispiel von ADAPT

Application Design for Analytical Processing Technologies (ADAPT) ist eine semantische Modellierungsmethode für multidimensionale Datenstrukturen, die der konzeptuellen Entwurfsebene zuzuordnen ist und sich nicht auf herkömmliche Datenmodellierungsmethoden für operative Systeme zurückführen läßt. Entwickelt wurde ADAPT vom Unternehmensberater Dan Bulos, Begründer der Symmetry Corporation, mit dem Ziel, adäquate Beschreibungsmittel für das OLAP-Verarbeitungskonzept zur Verfügung zu stellen. "My intent with ADAPT is to enable developers to build models that understand OLAP's heterogeneous quality; the basic objects of OLAP applications; and how these objects are interrelated." [Bulo96]

ADAPT besteht aus einer Vielzahl von Bausteinen zur Abbildung multidimensionaler Datenstrukturen [BuFo98], wobei die Symbole für *Hypercube*, *Dimension*, *Hierarchy*, *Model* und *Data Source* die Kernelemente darstellen (vgl. Abbildung 5). Ein multidimensionaler Würfel (*Hypercube*) wird durch Zuordnung von Dimensionen (*Dimension*) definiert. Im Universitätsbeispiel wird der *Hypercube* "Anzahl der Studierenden" durch die Dimensionen Studienabschnitt, Zeit und Studienausrichtung näher bestimmt. Durch die Entkopplung der Bausteine für Dimensionen (*Dimension*) und Dimensionshierarchien (*Hierarchy*) kann eine Dimension mit

mehreren parallelen Hierarchien verknüpft werden. Die Verbindung zwischen Dimensionen und korrespondierenden Hierarchien wird durch gerichtete Kanten symbolisiert. Den Dimensionen Studienabschnitt und Studienausrichtung ist jeweils eine Hierarchie zugeordnet. Die Zeitdimension hingegen besitzt keine eigene Hierarchisierung. Die beiden letzten Kernelemente schließlich sind das *Model*, das zur Abbildung von Berechnungsvorschriften dient, und die *Data Source*, zur Visualisierung der Verbindung zu den operativen Datenquellen.

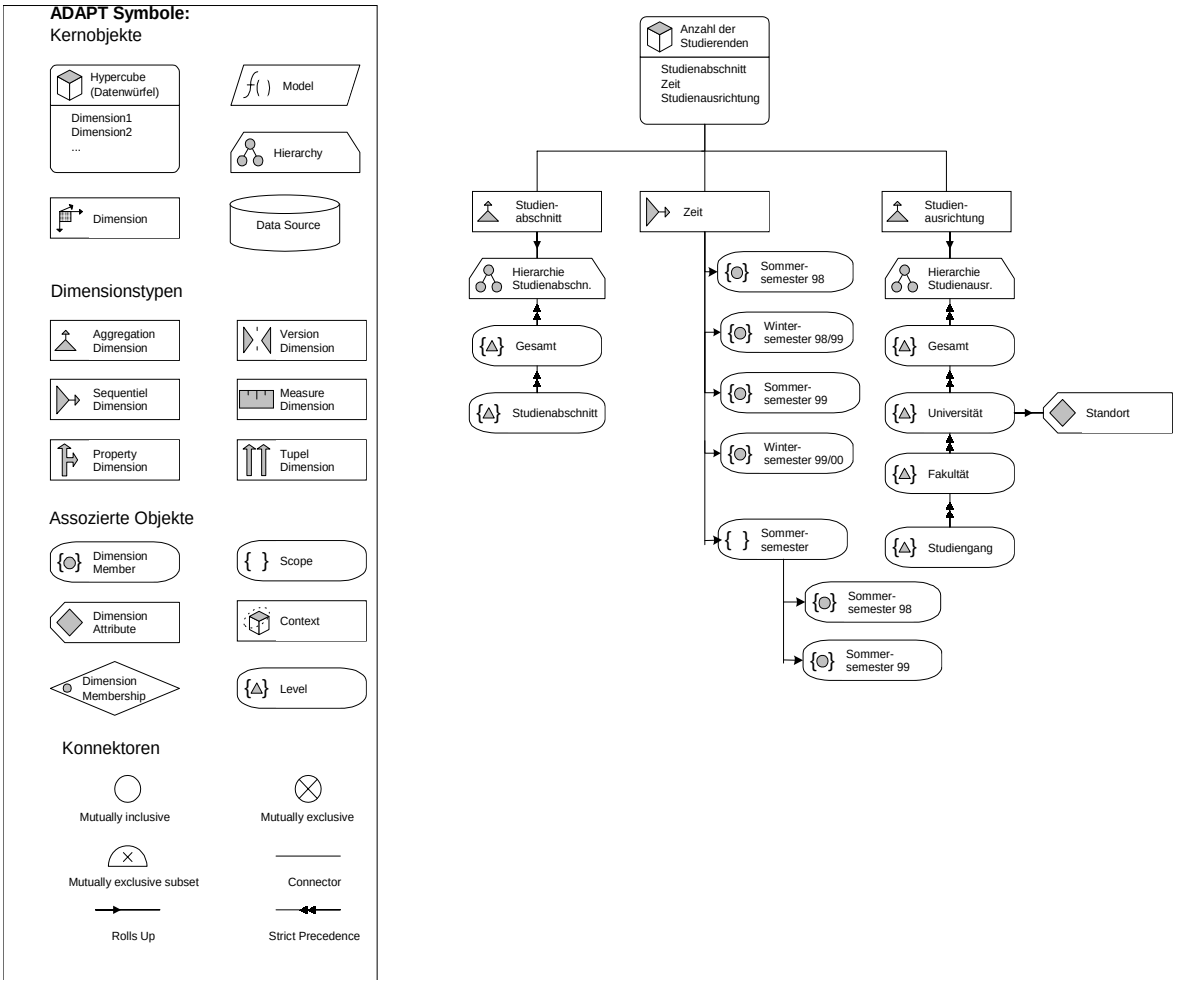

Abb. 5: Universitätsbeispiel mit ADAPT

Einzelne Elemente einer Dimension sind durch sog. *Dimension Members* angebbar. Der Zeitdimension sind die vier *Dimension Members* Sommersemester 98, Wintersemester 98/99, Sommersemester 99 und Wintersemester 99/00 zugeordnet. Anstelle von konkreten Ausprägungen werden bei Hierarchien die jeweiligen Verdichtungsstufen in Form von *Levels* zugeordnet. Die Richtung der Aggregation wird durch die *Rolls Up* bzw. *Strict Precedence* Kante mit doppelter Pfeilspitze gezeigt. Die Hierarchie der Dimension Studienausrichtung umfaßt beispielsweise die Aggregationsebenen (*Levels*) Studiengang, Fakultät, Universität und Gesamt. Weiterhin besteht die Möglichkeit *Dimensions* bzw. *Levels* zusätzliche Eigenschaften durch *Dimension Attributs* zuzuweisen. Der *Level* Universität könnte zum Beispiel über die Eigenschaft Standort näher beschrieben werden. Ein aus der Entity-Relationship-Modellierung entlehntes Konstrukt stellt die *Dimension Membership* dar. Es entspricht einem Relationship-Typ des ERM und dient der Verknüpfung von zwei nicht orthogonal zueinander stehenden Dimensionen. Beziehungs-

angaben werden dabei in der (1,M,N)-Notation beschrieben. *Scopes* und *Contexts* stellen Filtermöglichkeiten auf multidimensionale Datenstrukturen zur Verfügung. Während *Scopes* Teilmengen von Elementen einer Dimension visualisieren, erlaubt ein *Context* die Darstellung eines Teilwürfels. Ein *Scope* Sommersemester innerhalb der Zeitdimension könnte zum Beispiel nur die Dimensionselemente umfassen, die ein bestimmtes Sommersemester repräsentieren. Verschiedene *Scopes* werden über spezifische Konnektoren (*Mutually inclusive, Mutually exclusive, Mutually exclusive subset*) miteinander verknüpft und können damit komplexere Teilmengenbeziehungen abbilden.

In ADAPT werden sechs verschiedene Dimensionstypen unterschieden, die die jeweilige Dimension genauer charakterisieren. Die *Aggregation Dimension* ist der gebräuchlichste Dimensionstyp. Sie unterstellt mehrere Verdichtungsebenen mit spezifischen Aggregationsregeln innerhalb einer Dimension. Die Dimensionen Studienabschnitt und Studienausrichtung enthalten Konsolidierungspfade und sind daher *Aggregation Dimensions*. Dagegen setzt eine *Sequential Dimension* Dimensionselemente ordinaler Ausprägung voraus. Die Elemente der Zeitdimension stehen in sequentieller Reihenfolge zueinander. Eine *Property Dimension* stellt genaugenommen keine eigenständige Dimension dar. Sie dient vielmehr dazu Dimensionselementen weitere Eigenschaften zuzuordnen. *Version Dimensions* enthalten Dimensionselemente die verschiedene Szenarien beschrieben, wie z.B. Ist- und Plandaten. Dagegen enthalten *Measure Dimensions* unterschiedliche Maßzahlen. *Tuple Dimensions* schließlich stellen eine Kombination anderer Dimensionen in Form einer geordneten Menge von Tupeln dar.

Mit der Anwendung von ADAPT sind allerdings auch eine Reihe von Problemen verbunden. Durch die große Zahl an Beschreibungselementen ist ADAPT kaum intuitiv verständlich und erfordert eine intensive Einarbeitung ([GaGl97b][ToJa98]). Es lassen sich leicht sehr komplexe und unübersichtliche Beziehungsgeflechte aufbauen. Weiterhin ist die Semantik der Beschreibungsobjekte nur unzureichend definiert. Es existieren lediglich zwei Quellen ([Bulo96][BuFo98]), die die erlaubten Bausteine und Beziehungen anhand von wenigen Beispielen demonstrieren. Problematisch ist außerdem das Symbol *Data Source*, das auf konzeptueller Ebene eine Verknüpfung zur Datenherkunft in operativen Systemen visualisiert. Die Kennzahlendimension (*Measure Dimension*) erlaubt die Darstellung von Maßzahlen als Dimensionselemente einer Dimension und durchbricht damit das Prinzip der strikten Trennung von quantitativen Daten als Kern und qualitativen Daten als Dimensionen eines Datenwürfels. Das Symbol *Dimension Membership* ermöglicht die direkte Verknüpfung unterschiedlicher Dimensionen und verletzt somit die Forderung nach der Orthogonalität der einzelnen Dimensionen. Die *Property Dimension* ist genaugenommen kein eigenständiger Dimensionstyp, sondern regelt lediglich die Zuordnung von Attributen innerhalb einer Dimension. Außerdem verletzt das Symbol *Model* die Trennung von Daten- und Funktionssicht, da hier Verarbeitungslogik abgebildet werden kann.

4.3 Logische multidimensionale Modellierung am Beispiel von SAP BW

Das Data Warehouse-Produkt der SAP AG *Business Information Warehouse* (BW) verwendet zur logischen Modellierung eine modifizierte Form eines Star Schemas, das sog. *Developed Star Schema* ([SAP97][SAP98a][SAP98b]). Zunächst wird auf die Eigenschaften des klassischen *Star*- und *Snowflake Schemas* eingegangen. Anschließend werden die besonderen Merkmale des *Developed Star Schemas* der SAP AG erläutert.

4.3.1 Das klassische Star Schema

Das klassische *Star Schema* ist für den Einsatz relationaler Datenbanksysteme auf der physischen Entwurfsebene vorgesehen. Es unterscheidet zwischen zwei verschiedenen Arten von Tabellen: Fakt- und Dimensionstabellen, die beide durch Rechtecke symbolisiert werden. Während die Fakttabelle zur Speicherung des Würfelkerns, der Basiszahlen oder abgeleiteten Größen dient, beinhalten die Dimensionstabellen die qualitativen Daten zur Visualisierung von Dimensionen und Dimensionshierarchien des Würfels ([Rade95][Rade96]). Die Darstellung eines *Star Schemas* ähnelt bildlich einem Stern (Abbildung 6). Die einzelnen Zeilen einer Dimensionstabelle werden durch eine minimale Attributkombination, dem Primärschlüssel, identifiziert. Zur Herstellung der Beziehung zwischen den Dimensionstabellen und der zugehörigen Fakttabellen werden die Primärschlüssel der Dimensionstabellen in die Fakttabelle als Fremdschlüssel aufgenommen und bilden dort wiederum zusammen den Primärschlüssel der Fakttabelle. Zur Darstellung dient eine Kante mit oder ohne Pfeilspitze zwischen den jeweiligen Fakt- und Dimensionstabellen. Um auf Datenbankebene eine leichtere Verbindung zwischen Fakt- und Dimensionstabellen zu ermöglichen und die Anzahl der benötigten Verbund-Operationen einzuschränken, sind alle zu einer Dimension gehörigen Daten, einschließlich der Dimensionshierarchien, in einer einzigen nicht normalisierten Tabelle, der Dimensionstabelle, gespeichert. Diese Denormalisierungen beinhalten eine Vielzahl von Redundanzen, die zu Einfüge-, Änderungs- und Löschanomalien führen können und daher besonderer Beachtung bedürfen.

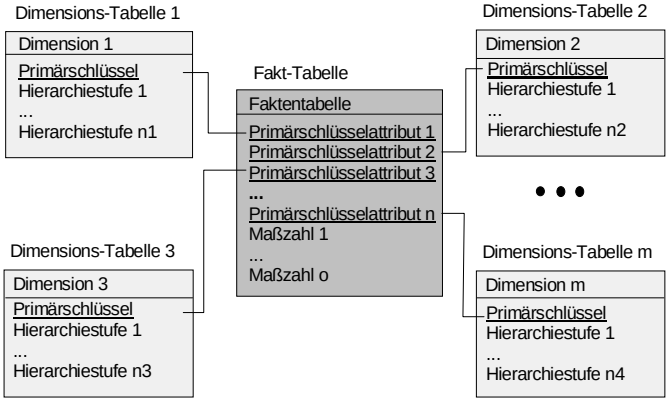

Abb. 6: Klassisches Star Schema

4.3.2 Das klassische Snowflake Schema

Beim klassischen Snowflake Schema (Abbildung 7) hingegen werden die Daten in den Dimensionstabellen in dritter Normalform gespeichert. Durch die Normalisierung entsteht für jede Hierarchiestufe einer Dimension eine eigene Tabelle. Die Beziehungen zwischen den Tabellen werden wie beim Star Schema durch Fremdschlüssel hergestellt. Eine große Zahl von Verbundoperationen bei Anfragen über mehrere Dimensionen oder Dimensionsstufen wiegen häufig den Vorteil des geringeren Speicherplatzverbrauchs und der Kontrolle der Redundanzen auf.

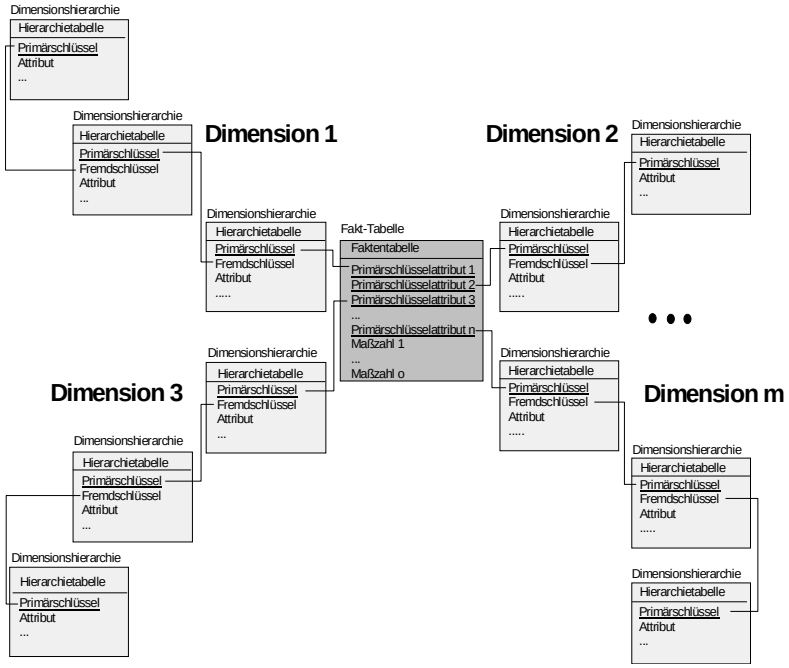

Abb. 7: Klassisches Snowflake Schema

4.3.3 Das Developed Star Schema

Das *Developed Star Schema* der SAP AG stellt eine Erweiterung des klassischen *Star Schemas* dar und besteht aus zwei Grundbestandteilen, zum einen aus dem (1) *InfoCube*[2] *Star Schema*, Fakt- und Dimensionstabellen, entsprechend dem klassischen *Star Schema*, zum anderen aus einem (2) *InfoCube* unabhängigen Teil, den sog. *Master Data-*, *Text-* und *External Hierarchy-*Tabellen (vgl. Abbildung 8).

Ad (1): Im Gegensatz zum klassischen *Star Schema* werden die Primärschlüssel der Dimensionstabellen künstlich vergeben, sog. Surrogatschlüssel, und sind nicht an Sachverhalte der Realität angelehnt, wie z.B. eine Produktnummer in einer Produktdimension. Als Surrogatschlüssel dient das Attribut *Dim ID*, das in Form eines vier Byte Ganzzahlwertes geführt wird. Der Primärschlüssel der Fakttabelle ergibt sich durch Aufnahme aller Surrogatschlüssel der einzelnen Dimensionstabellen. Die Dimensionshierarchien werden in Form von Attributen, sog. *Characteristics*, in den Dimensionstabellen abgelegt. Der Einsatz künstlicher Schlüssel erlaubt die Vergabe von Nullwerten bei einem Teil der Primärschlüsselatribute und ermöglicht dadurch unaus-

2. Ein *InfoCube* entspricht einem multidimensionalen Datenwürfel.

geglichene Hierarchien (*Unbalanced Hierarchies*) und N:M-Beziehungen innerhalb einer Dimension. Neben benutzerdefinierten Dimensionen erfordert das SAP BW die drei vorgegebenen Dimensionen *Time*, *Unit* und *Packet*.

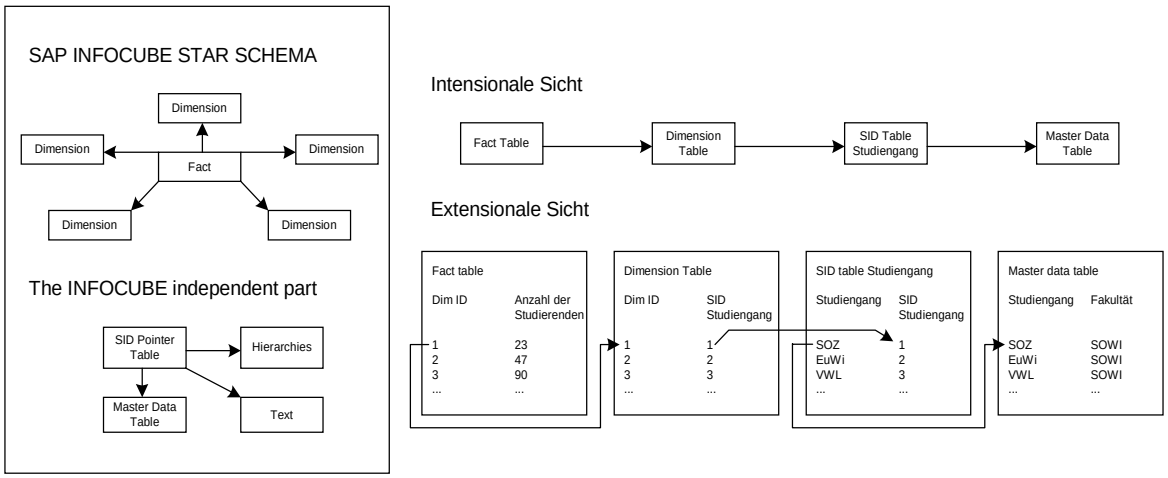

Abb. 8: Universitätsbeispiel mit SAP BW

Ad (2): Bei den Tabellen des *InfoCube* unabhängigen Teils handelt es sich um Daten, die einmal definiert, beliebig oft mit verschieden *InfoCubes* verknüpft werden können. Eine indirekte Verbindung zwischen den Dimensionstabellen und dem *InfoCube* unabhängigen Teil geschieht durch künstlich systemvergegebene Attribute, sog. *Set IDs* (SIDs), und erfordert eine eigene SID-Tabelle. Daten über Dimensionshierarchien müssen nicht innerhalb der Dimensionstabelle gehalten werden, sondern können mit Hilfe einer SID in eine *Hierarchy Table* ausgelagert werden. *Hierarchy Tables* erlauben die Definition mehrerer externer Hierarchien für eine *Characteristic*. Dabei werden externe Hierarchien i.d.R. aus den SAP R/3 OLTP Modulen importiert, wie z.B. die Standardhierarchie für *Cost Center* aus dem SAP Controlling-Modul. Die Attribute einer *Master Data Table* können von einem existierenden SAP OLTP System stammen oder benutzerdefiniert sein. Nicht für jede *Characteristic* einer Dimensionstabelle muß eine korrespondierende *Master Data Table* existieren. Es lassen sich in *Master Data Tables* zwei verschiedene Arten von Attributen unterscheiden: *Navigational Attributes*, mit denen Navigationsoperationen, wie z.B. *Drill Down* oder *Roll Up*, in den InfoCube-Datenbeständen möglich sind und *Reporting Attributes*, die in Berichten Verwendung finden, aber keine Navigationsmöglichkeiten bieten. Textuelle Beschreibungen von *Characteristics* werden in einer eigenen *Text Table* gespeichert, um eine sprachabhängige Beschreibung von Attributwerten und damit die Unterstützung verschiedener Landessprachen zu ermöglichen.

Beim *Developed Star Schema* der SAP AG handelt es sich um eine vielversprechende Variante des klassischen Star Schemas, die besonders für die Einbindung des SAP R/3 Systems, als zentrales operatives OLTP-System, konzipiert ist.

5 Evolution von Data Warehouse-Systemen

In den vorigen Abschnitten wurden die Verarbeitungskonzepte des Online Analytical Processing und die Modellierung multidimensionaler Datenstrukturen vorgestellt. Die folgenden Abschnitte behandeln *Data Warehouse-Systeme* als Grundlage von Entscheidungsunterstützungswerkzeugen.

Seit den 60er Jahren wurde in den Unternehmen immer mehr Software zur Automatisierung der Geschäftsabläufe eingeführt. Diese sogenannten operativen Systeme, z.B. Buchhaltungs- oder Lagerhaltungsprogramme, basieren auf Datenbanken in denen die operativen, transaktionalen Daten gespeichert werden (OLTP; s. Abschnitt 2). Aufgrund des steigenden Wettbewerbs wuchs jedoch im Laufe der Zeit in den Unternehmen der Bedarf, diese Daten für die Entscheidungsunterstützung zu nutzen. Da die für das OLAP notwendigen multidimensionalen Datenmodelle und Operatoren in den transaktionalen Datenhaltungssystemen nicht zur Verfügung stehen, können die multidimensionalen Analysewerkzeuge nur bedingt direkt auf den transaktionalen Rohdaten aufgesetzt werden. Die Verteilung der Unternehmensdaten auf mehrere, meist heterogene und autonome Datenhaltungssysteme macht eine konsolidierte Analyse häufig gänzlich unmöglich. Für eine effektive Entscheidungsunterstützung müssen die unterschiedlichen Datenquellen in eine Informationsquelle transformiert werden, die den Anwendern einen integrierten und konsistenten Zugriff auf die Unternehmensdaten ermöglicht.

Generell sprechen die folgenden fachlichen und technischen Probleme gegen einen Einsatz multidimensionaler Analysewerkzeuge auf der Basis der Rohdaten in den operativen Systemen:

- ***Fachliche Probleme:***
  - *Konsolidierungsprobleme:* Unter Umständen werden Informationen über dasselbe Objekt in den verschiedenen heterogenen Systemen unterschiedlich gespeichert.
  - *Problem der Schemaintegration:* Wie in Abschnitt 4 erläutert, wird zur multidimensionalen Analyse ein einheitliches Datenschema benötigt. Ein einheitliches Gesamtschema kann jedoch auf der Basis der operativen Systeme aufgrund ihrer Heterogenität häufig nicht gewährleistet werden.
  - *Keine historischen Daten:* In den operativen Datenquellen stehen normalerweise keine historisierten Daten über einen längeren Zeitraum zur Verfügung.
  - *Konsistenzprobleme:* Spezielle Anforderungen an die zeitliche als auch an die inhaltliche Konsistenz der Daten können nicht berücksichtigt werden. Beispielsweise werden meist Daten für bestimmte abgeschossene Perioden (z.B. Verkaufstag, -woche, usw.) benötigt.
- ***Technische Probleme:***
  - *Keine effizienten Zugriffsmechanismen:* Da die operativen Systeme auf die transaktionale Verarbeitung der Daten hin ausgerichtet sind, fehlen entsprechende Techniken zur Optimierung der Anfragelaufzeiten hinsichtlich multidimensionaler Analysen.

- *Beeinträchtigung der operativen Systeme:* Die Verdichtung der Daten für multidimensionale Analysen führt aufgrund der Aggregationsoperationen zu sehr langen Anfragelaufzeiten. Diese beeinträchtigen die transaktionale Verarbeitung in den operativen Systemen.
- *Schlechtes Antwortzeitverhalten:* Die häufig sehr langen Anfragen resultieren aus der Laufzeit für die Transformation der multidimensionalen Anfrageoperationen in die jeweilige Anfragesprache, der Dauer für die Berechnung der Teilanfragen auf den operativen Datenquellen und der Konstruktion des einheitlichen Gesamtergebnisses.

Aus diesen Gründen werden Konzepte zur Integration der heterogenen Informationsquellen benötigt. Generell lassen sich dafür zwei verschiedene Ansätze unterscheiden: *Mediatoren* und *Data Warehouse-Systeme*. ***Mediatoren*** stellen den Analysewerkzeugen ein virtuelles Gesamtschema zur Verfügung, transformieren die Anfragen in Teile für die jeweiligen heterogenen Datenquellen und erzeugen abschließend aus den unterschiedlichen Anfrageergebnissen ein einheitliches Anfrageresultat [QRS+95]. Die Verwendung eines Mediators ist für die Analysewerkzeuge vollständig transparent. Ein Beispiel für die Realisierung eines Mediatorsystems ist der *The Stanford-IBM Manager of Multiple Information Sources* (TSIMMIS) [GPQ+97]. Die dynamische Transformation kann jedoch ausschließlich die fachlichen Probleme bezüglich der Konsolidierung und der Schemaintegration beseitigen. Im Gegensatz zu Mediatoren ermöglichen ***Data Warehouse-Systeme*** einen effizienten Zugriff auf integrierte und historische Informationen aus unterschiedlichen, teilweise heterogenen und autonomen Informationsquellen. Zu diesem Zweck werden die Daten aus den operativen Quellen extrahiert, aufbereitet und re-dundant im Data Warehouse gespeichert. Spezielle Speicher- und Zugriffsstrukturen gewährleisten die für Datenanalysen erforderlichen Antwortzeiten. In seiner Grundidee geht das Data Warehouse-Konzept auf den EBIS-Ansatz (Europe / Middle East / Africa Business Information System) der IBM aus dem Jahr 1988 zurück [Toto97].

Nach Inmon [Inmo96] sind die folgenden Eigenschaften wesentlich für ein Data Warehouse:

- *subject-oriented:* Das konzeptuelle Schema eines Data Warehouses ist auf die zu analysierenden Themen und nicht hinsichtlich der operativen Anwendungsstrukturen ausgerichtet.
- *integrated:* Sowohl syntaktische als auch semantische Inkonsistenzen in den Daten werden im Data Warehouse beseitigt.
- *time-varying* und *non-volatile:* Um Zeitreihenanalysen zu ermöglichen, werden, im Gegensatz zu den operativen Datenquellen, im Data Warehouse die Informationen über längere Zeiträume hinweg gespeichert. Da keine aktuellen Daten (im Sinne von aktuellen Transaktionen) benötigt werden, erfolgt die Aktualisierung des Data Warehouses zu definierten Zeitpunkten (z.B. Tag, Monat). Gespeicherte Informationen werden normalerweise nicht wieder aus dem Data Warehouse entfernt.

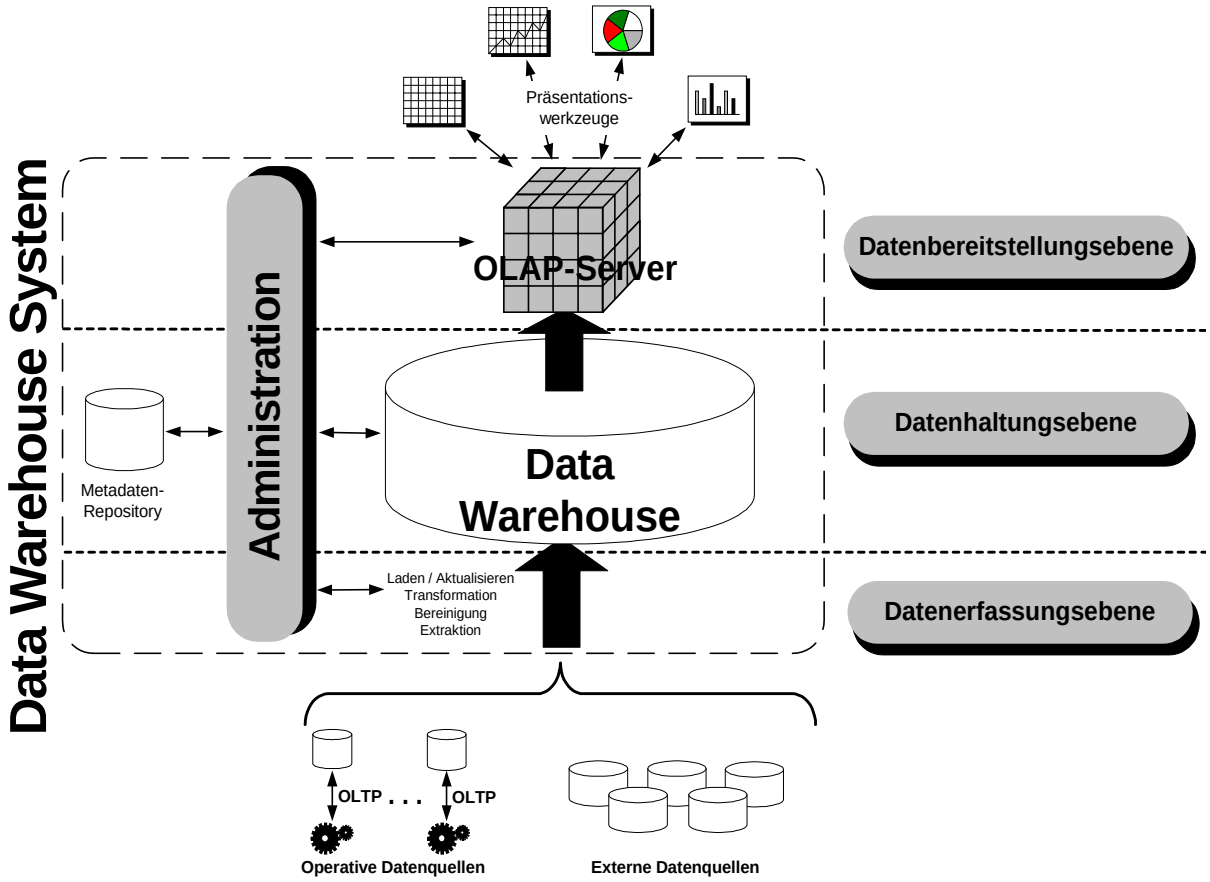

Abb. 9: Überblick über die Data Warehouse-System Architektur

In der Literatur finden sich unterschiedliche Definitionen des Begriffs *Data Warehouse*. Im folgenden wird analog zur Definition von *Datenbank* und *Datenbanksystem* [LoDi87] zwischen einem *Data Warehouse* und einem *Data Warehouse-System* unterschieden. Das *Data Warehouse-System* umfaßt sowohl das Data Warehouse-Management-System mit Extraktions-, Bereinigungs-, Datenbereitstellungs- und Administrationsfunktionen als auch das eigentliche Data Warehouse, den gegenüber operativen Systemen redundant gehaltenen Datenpool.

6 Architektur von Data Warehouse-Systemen

Die logische Architektur eines Data Warehouse-Systems ist in Abbildung 9 dargestellt. Das System läßt sich zunächst in drei Ebenen untergliedern: *Datenerfassungsebene* mit der Schnittstelle zu den operativen Systemen, *Datenhaltungsebene* mit dem eigentlichen Data Warehouse und *Datenbereitstellungsebene* mit den Schnittstellen zu den Endanwendungen und Präsentationswerkzeugen. Die Datenerfassungsebene beinhaltet Werkzeuge zur Extraktion der Daten aus den operativen Quellen, zur Datenaufbereitung und zum Laden bzw. Aktualisieren des Data Warehouses. Auf der *Datenhaltungsebene* werden spezielle Indizierungsverfahren und die explizite Verwendung von Redundanzen zur Erhöhung der Zugriffszeiten eingesetzt. Die *Datenbereitstellungsebene* besteht meist aus einem oder mehreren OLAP-Servern, die die mul-

tidimensionalen Strukturen für die Endanwendungen und Präsentationswerkzeuge zur Verfügung stellt. Allen Ebenen sind Administrationsfunktionen zugeordnet, die durch eine *Metadatenbank* bzw. *Metadaten-Repository* unterstützt werden. Dieses Repository enthält Informationen über die im Data Warehouse gespeicherten Daten.

Im weiteren Verlauf des Abschnitts werden die einzelnen Bestandteile dieser Data Warehouse-Systemarchitektur genauer erläutert.

6.1 Datenerfassungsebene

Auf der Datenerfassungsebene werden die für die Nutzer relevanten Daten aus den unterschiedlichen heterogenen operativen Datenquellen zusammengeführt, um sie anschließend im Data Warehouse zu speichern. Nach dem initialen Laden muß das Data Warehouse entsprechend den benutzerdefinierten Aktualitäts- und Konsistenzanforderungen aktualisiert werden. Dafür existieren zwei unterschiedliche Strategien: das *neue, vollständige Laden* (*reloading*) und die *inkrementelle Aktualisierung* (*incremental maintenance*). Statt einen Teil der Daten im Warehouse zu löschen und neu zu laden, werden bei der inkrementellen Aktualisierung lediglich die Änderungen der operativen Daten in das Data Warehouse eingebracht.

Die Datenerfassung kann in die Abschnitte *Extraktion* - dem Filtern der relevanten Daten aus den operativen Datenquellen, *Datenbereinigung* - der syntaktischen und semantischen Datenaufbereitung und in die abschließende *Datenübernahme* in das Data Warehouse untergliedert werden. Im folgenden werden diese Abschnitte genauer betrachtet.

6.1.1 Extraktion

Zur inkrementellen Aktualisierung müssen zunächst die für das Data Warehouse relevanten Änderungen aus den operativen Systemen extrahiert werden. Abhängig vom jeweiligen operativen Datenhaltungssystem gibt es dafür die folgenden Möglichkeiten:

- ***Trigger:*** Unterstützt das operative Datenhaltungssystem die Verwendung von SQL-Triggern, dann können diese zur Extraktion der Daten verwendet werden. Trigger erlauben die Definition von Datenmanipulationsoperationen, die aufgrund bestimmter Ereignisse automatisch ausgeführt werden. Zur Extraktion der Daten werden Trigger verwendet, die bei jeder Änderung des operativen Datenbestands ausgelöst werden, um diese Modifikationen in einer entsprechenden Modifikationstabelle zu speichern [CeWi91]. Die Datensätze in den Modifikationstabellen können periodisch an das Data Warehouse übertragen werden.
- ***Protokolldateien:*** In [KaRi87] wird gezeigt, wie Protokolldateien (*Logs*) der operativen Datenbanksysteme für die Extraktion eingesetzt werden können. Die meisten Datenbanksysteme unterstützen Protokolldateien, um einen konsistenten Zustand der Datenbank nach einem Systemausfall wiederherzustellen (*Recovery*). Mit speziellen Programmen können die Protokolldateien ausgewertet und die darin aufgezeichneten Datenmanipulationsoperationen extrahiert werden. Anschließend erfolgt die Übertragung der Daten zum Data Warehouse.

- ***Monitorprogramme:*** Unter Umständen können weder Trigger noch Protokolldateien zur Extraktion der Daten eingesetzt werden, sondern es ist lediglich der Zugriff auf den operativen Datenbestand möglich. In diesem Fall können die relevanten Modifikationen mit Hilfe spezieller Monitorprogramme extrahiert werden, die periodisch einen Abzug der Datenbasis erzeugen und anschließend die Differenz zum vorherigen Abzug berechnen (*snapshot differential algorithm*) [LaGa96].
- ***Modifikation der operativen Anwendungssysteme:*** Unterstützt das entsprechende operative Datenhaltungssystem weder Trigger noch Protokolldateien und ist zudem der Zugriff auf die Daten mit Hilfe eines Monitorprogramms nicht möglich, dann muß notfalls das operative Anwendungssystem selbst so modifiziert werden, daß jede durchgeführte Änderung zusätzlich in einer separaten Datenbank protokolliert wird. Die relevanten Modifikationen können wiederum periodisch an das Data Warehouse übertragen werden.

Zur Aktualisierung des Data Warehouses ist unter Umständen neben dem Filtern der relevanten Änderungen auch die Berechnung von Anfragen auf den operativen Daten erforderlich [ZhGW96]. Aufgrund der Heterogenität der operativen Systeme ist häufig die Transformation der Anfragesprachen und -resultate notwendig. Dazu können die für die Mediatoren entwickelten Methoden eingesetzt werden [WGLZ96] (s. Abschnitt 5).

Ein weiterer Aspekt der Datenerfassung, der meist dem Extraktionsprozeß zugeordnet wird, ist die Datenübertragung. Um die Anzahl der für die inkrementelle Aktualisierung zu übertragenen Modifikationen zu reduzieren, können die irrelevanten Änderungen mit Hilfe spezieller Filtertechniken [LeSa93] eliminiert werden. In der Regel kommen für die Übertragung der Daten Standardinterfaces und -gateways zum Einsatz (z.B. Microsoft ODBC bzw. OLE-DB, Oracle Open Connect, Informix Enterprise Gateway) [ChDa97]. Darüber hinaus können Replikationsmechanismen (z.B. ORACLE Replication Server oder SYBASE Replication Server) eingesetzt werden, wenn das entsprechende operative Datenbanksystem diese Technologie unterstützt.

6.1.2 Datenbereinigung

Während der Konstruktionsphase wird für das Data Warehouse ein einheitliches Datenschema festgelegt. Die Daten aus den operativen Systemen müssen dann gegebenenfalls in das Format der im Data Warehouse verwendeten Datentypen transformiert werden. Darüber hinaus führen das große Datenvolumen im Data Warehouse und die Heterogenität der Datenquellen häufig zu Anomalien und Inkonsistenzen. Beispielsweise können die folgenden Probleme auftreten: Verschiedene Zeichensätze, Verwendung unterschiedlicher Attributwerte für gleiche Attribute in den operativen Systemen, fehlende Werte, Verletzung von Integritätsbedingungen usw.. Zur Gewinnung eines korrekten und konsistenten Data Warehouses müssen diese Fehler beseitigt werden. In [CeMc95] wird zur Einordnung der am Markt verfügbaren Datenbereinigungswerkzeuge die folgende Klassifizierung vorgeschlagen:

- *Data-Migration-Werkzeuge*: Diese Kategorie der Datenbereinigungswerkzeuge erlaubt die Definition von einfachen Transformationsregeln, um die Daten aus den Quellsystemen in das Zielformat zu transformieren (z.B. die Transformation des Attributwertes 'm' in 'männlich').
- *Data-Scrubbing-Werkzeuge*: Data-Scrubbing-Werkzeuge verwenden bereichsspezifisches Wissen zur Bereinigung der Daten aus den operativen Quellen. Beispielsweise können Anschriften mittels einer Tabelle über alle Postleitzahlen und den dazugehörigen Orts- und Straßennamen überprüft und korrigiert werden. Technologisch basieren diese Werkzeuge häufig auf Fuzzy-Logik und Neuronalen Netzen.
- *Data-Auditing-Werkzeuge*: Mit Data-Auditing-Werkzeugen können Regeln und Beziehungen zwischen den Daten erkannt werden. Anschließend wird untersucht, ob die festgestellten Regeln verletzt wurden. Da eine definitive Fehlersituation nicht bestimmt werden kann, werden die aufgetretenen Unstimmigkeiten lediglich dem Administrator zur manuellen Korrektur gemeldet. Beispielsweise können solche Werkzeuge auf der Basis von statistischen Auswertungen feststellen, daß aufgrund sehr hoher Unterschiede zwischen bestimmten Werten falsche Eingaben vorliegen könnten. Zur Analyse der Daten werden häufig Data-Mining-Techniken eingesetzt (Abschnitt 6.3).

6.1.3 Datenübernahme

Nach der Extraktion der relevanten Daten und der anschließenden Datenbereinigung erfolgt die eigentliche Übernahme der Daten in das Warehouse. Während der Datenübernahme müssen häufig die folgenden Aufgaben durchgeführt werden: *Überprüfen von Integritätsbedingungen*, *Sortieren der Daten*, *Berechnen von Aggregationen*, *Generieren von Zugriffsstrukturen* (z.B. Indexe) und *Partitionieren* der Daten für einen effizienten Zugriff [ChDa97].

Die Datenübernahme in das Data Warehouse soll zunächst von der Übermittlung der Daten aus den operativen Systemen entkoppelt werden. Dazu werden *Aktualisierungstabellen* (*Update-Tables*) [TeUl97] bzw. sogenannte *Operational Data Stores* [IRBS99] verwendet. In diesen werden die Daten aus den operativen Systeme so lange gespeichert, bis sie in das Data Warehouse überführt werden können. Um die Dauer des Ladevorgangs zu verkürzen, können innerhalb der Aktualisierungstabellen bereits Datenbereinigungs- und Aggregationsoperationen ausgeführt werden.

Die Datenübernahme in das Data Warehouse erfolgt auf der Basis der benutzerdefinierten Aktualitäts- und Konsistenzanforderungen. Um inkonsistente Anfrageergebnisse zu vermeiden, ist während der Aktualisierung der Zugriff auf die Daten im Warehouse nicht möglich. Aus diesem Grund werden die meisten Data Warehouses zu einem Zeitpunkt aktualisiert, während dessen kein Zugriff auf die Daten erfolgt (z.B. nachts, am Wochenende). Um die Störung der Endanwender zu vermeiden, muß der Aktualisierungsvorgang so kurz wie möglich sein. Daher werden Werkzeuge benötigt, die mittels Parallelität den Ladevorgang entsprechend verkürzen.

In bestimmten Fällen ist jedoch der Zugriff auf die Daten im Data Warehouse rund um die Uhr (24 Stunden und 365 Tage im Jahr) erforderlich. Hierzu werden in [QuWi97] und [TeUl98]

Methoden vorgestellt, mit denen auch in diesem Fall der konsistente Zugriff auf die Daten gewährleistet werden kann. Neben der Aktualisierung des eigentlichen Data Warehouses ist es während der Datenübernahme ebenfalls notwendig die Datenbasis für das verwendete OLAP-System (Abschnitt 6.3) zu aktualisieren. Für relationale OLAP-Systeme ist diese Datenbasis normalerweise mit dem Data Warehouse identisch. Multidimensionale OLAP-Systeme basieren jedoch auf speziellen multidimensionalen Datenbanksystemen, die zusätzlich aktualisiert werden müssen. In diesem Zusammenhang wird in [GPQ+97] auf Probleme beim Sperren multidimensionaler Datenbanksysteme hingewiesen. Auf die unterschiedlichen OLAP-Technologien wird in Abschnitt 6.3.1 ausführlich eingegangen.

6.2 Datenhaltung

Grundlegend besteht ein Data Warehouse aus Relationen, die von den entsprechenden Quellrelationen in den operativen Systemen abgeleitet sind. Datenbanktechnisch werden solche Relationen auch als *materialisierte Sicht* (*materialized view*) bezeichnet [GuMu95]. Im Gegensatz zu *virtuellen* bzw. herkömmlichen *Sichten* (*views*) in Datenbanken, bei denen lediglich die Definition der Anfrage gespeichert und das Anfrageergebnis bei jedem Zugriff auf die Sicht erneut berechnet wird, speichern materialisierte Sichten das Anfrageergebnis in einer eigene Relation. Beim Zugriff auf die materialisierte Sicht wird daher die Berechnung der Anfrage vermieden. Ändern sich jedoch die Daten in den Quellrelationen, so müssen diese Modifikationen entsprechend den benutzerdefinierten Aktualitätsanforderungen in der materialisierten Sicht nachgeführt werden. In Data Warehouse-Systemen werden meist keine aktuellen Daten benötigt. Aus diesem Grund müssen die materialisierten Sichten im Data Warehouse nicht bei jeder Änderung der operativen Daten aktualisiert werden, sondern es ist lediglich eine zeitversetzte, periodische Aktualisierung erforderlich. Damit kann die Aktualisierung des Data Warehouses auf die *Aktualisierung materialisierter Sichten* (*materialized view maintenance*) [GuMu95] zurückgeführt werden. Der Datenhaltungsteil des Data Warehouse-Systems entspricht daher einer Menge von materialisierten Sichten, die auf den Rohdaten in den Quellsystemen basieren.

Ein Data Warehouse-System soll den effizienten Zugriff auf integrierte und historische Informationen gewährleisten. Um zeitliche Verläufe der Daten zur Verfügung zu stellen, muß das Data Warehouse sehr große Datenmengen von mehreren hundert Megabyte bis zu einigen Terabyte aufnehmen können. Diese großen Datenbeständen erfordern spezielle Techniken für eine effiziente Anfragebearbeitung. Zwei Anfrageoptimierungsmöglichkeiten werden im folgenden Abschnitt kurz erläutert: die Verwendung von *Aggregationstabellen* und spezielle *Indizierungsmethoden*. Anschließend werden die Organisations- bzw. Verteilungsformen der Datenhaltungsebene innerhalb des Data Warehouse-Systems und der Begriff der *Data Marts* geklärt.

6.2.1 Techniken zur Anfrageoptimierung

Neben den materialisierten Sichten im Data Warehouse, die auf den Quellrelationen basieren, können zusätzlich sogenannte Aggregationstabellen bzw. Preaggregationen zur Anfrageoptimierung verwendet werden. Dabei handelt es sich um materialisierte Sichten, die häufig benö-

tigte voraggregierte Daten enthalten. Statt die benutzerdefinierten Anfragen auf der Basis der Detaildaten zu berechnen, kann direkt auf die voraggregierten Werte zugegriffen werden. Neben dem zusätzlichen Speicherbedarf ist als weiterer Nachteil der zusätzliche Aufwand während der Aktualisierung des Data Warehouses zu nennen.

Bei der Verwendung von Aggregationstabellen ergeben sich zwei Probleme: die *Auswahl der zu materialisierenden Sichten* mit Rücksicht auf den Speicher- und Aktualisierungsaufwand und die *Verwendung dieser Sichten zur Anfragebearbeitung* [TeUl97]. Aufgrund des zusätzlichen Speicherbedarfs und des Aktualisierungsaufwands ist die vollständige Materialisierung aller Aggregationskombinationen nicht möglich. Darüber hinaus wird meist nur ein geringer Teil dieser Kombinationsmöglichkeiten von den Anwendern wirklich benötigt. In [HaRU96] wird ein Algorithmus vorgestellt, mit dessen Hilfe die materialisierten Sichten ausgewählt werden, die zur optimalen Anfrageoptimierung für eine gegebene Speicherrestriktion führen. Nachdem die gewünschten Aggregationstabellen gespeichert wurden, müssen diese bei der Anfragebearbeitung verwendet werden. In [YaLa87], [CKPS95], [GuHQ95] und [LeMS95] werden Methoden zur Verwendung von materialisierten Sichten bei der Anfragebearbeitung beschrieben.

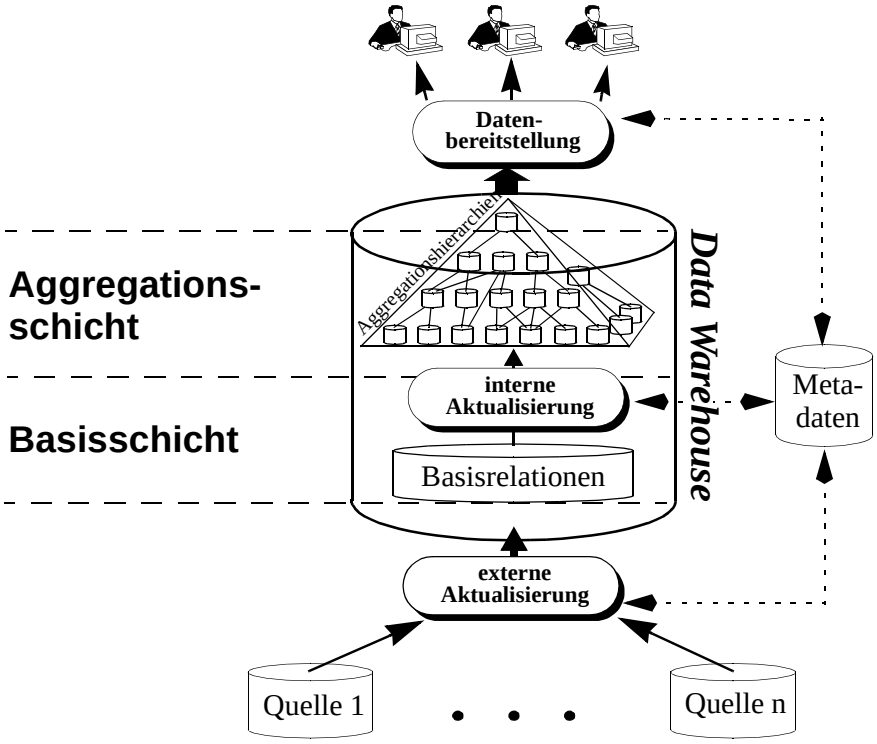

Abb. 10: Basis- und Aggregationsschicht im DWH [TeUl97]

Mit der Trennung zwischen Basistabellen und Aggregationstabellen ergibt sich die in Abbildung 10 dargestellte Struktur für den inneren Aufbau eines Data Warehouse ([TeUl97]). Alle materialisierten Sichten, die direkt aus den Relationen der Quellsysteme abgeleitet sind, werden als Basisrelationen der Basisschicht des Data Warehouses zugeordnet. Diese Relationen dienen als konsistente Grundlage für die weiteren Tabellen im Data Warehouse. Zur Vermeidung von Änderungsanomalien bei der Aktualisierung ([ZhGW96]) und zur besseren Anpaßbarkeit des Data Warehouses werden in den Basisrelationen häufig detailliertere Daten gespei-

chert, als eigentlich für die Endanwender benötigt werden. Die Aggregationsschicht enthält die Aggregationstabellen zur Anfragebearbeitung. Da Aggregationstabellen wiederum auf der Basis anderer Aggregationstabellen definiert werden können, entstehen sogenannte Aggregationshierarchien. In [CoBS98] wird der Informationsfluß innerhalb solcher Aggregationshierarchien, von detaillierten Daten zu aggregierten Daten, als *Upflow* bezeichnet. Bei der Verwendung von relationalen OLAP-Werkzeugen enthält die Aggregationsschicht meist die für das Anfragewerkzeug notwendigen Datenstrukturen (Abschnitt 6.3). Mit dieser Abgrenzung in Basis- und Aggregationsschicht ist es außerdem ohne Beeinträchtigung der Konsistenz möglich, den Anwendern gleichzeitig unterschiedliche Aktualitätsniveaus zur Verfügung zu stellen ([TeUl97], [ZhWG97], [TeUl98]).

Zusätzlich zur Einführung von Redundanzen durch die Verwendung von Aggregationstabellen, können spezielle Indizes zur Beschleunigung der Anfragelaufzeiten im Data Warehouse eingesetzt werden. In [GHRU97] wird die effiziente Verwendung dieser Techniken speziell für den Bereich des Online Analytical Processing behandelt. Neben herkömmlichen Indexstrukturen haben sich im Data Warehouse-Bereich zwei Versionen besonders hervor getan: der *Bitmap-Index* [ONQu97] und der *Join-Index* [ONGr95].

Mit einem Bitmap-Index kann sehr effizient festgestellt werden, welche Datensätze eine bestimmte Ausprägung für ein Attribut haben. Beispielsweise wird ein Bitmap-Index für das Attribut *Geschlecht* in einer Kundentabelle verwendet. Daraus ergibt sich ein Bit-Vektor mit dem Wert 1 für *weiblich* und 0 für *männlich*. Die Selektion aller weiblichen Kunden erfordert lediglich einen booleschen Vergleich mit dem Wert 1 und ist dadurch sehr effizient. Außerdem ist der zusätzliche Speicherbedarf für diesen Index sehr gering, da es sich lediglich um einen Bitvektor handelt. Bei der Verwendung von mehreren Bitmap-Indizes können mit Hilfe der booleschen Operationen AND und OR sehr effizient Schnitt- und Vereinigungsmengen berechnet werden, beispielweise bei der Selektion aller weiblichen Kunden, die verheiratet sind. Der Bitmap-Index ist aber nicht nur auf zwei-elementige Ausprägungsmengen ((weiblich, männlich), (ledig, verheiratet)) beschränkt. Zur Erfassung aller Ausprägungsmöglichkeiten eines Attributs werden lediglich mehrere Bits benötigt [IRBS99]. Trotzdem eignen sich Bitmap-Indizes nur für Attribute mit einer relativ geringen Anzahl unterschiedlicher Möglichkeiten für den Attributwert, sog. niedrige Selektivität.

Im Gegensatz zu herkömmlichen Indexstrukturen, die sich nur auf eine Tabelle beziehen, wird beim Join-Index die Verbindung zweier Relationen bezüglich einer Fremdschlüsselbeziehung indiziert [ChDa97]. Hierzu werden zu jedem Primärschlüsselattribut der Referenztabelle die Verweise auf alle Datensätze mit zugehörigem Fremdschlüsselattribut gespeichert. Dadurch können Verknüpfungen (Join) zwischen zwei Tabellen sehr effizient berechnet werden. Die Erweiterung des Join-Index-Konzeptes auf das Star Schema (Abschnitt 4.1) wird als Star-Index bezeichnet. Mit diesem Index werden alle Fremdschlüsselbeziehungen zwischen einer Faktentabelle und den dazugehörigen Dimensionstabellen gespeichert. Der Star-Index wird dazu verwendet, um für ein oder mehrere Dimensionselemente sehr effizient die zugehörigen Datensätze in der Faktentabelle zu bestimmten.

6.2.2 Organisationsformen der Datenhaltungsebene

Zur Realisierung der Datenhaltungsebene von Data Warehouse-Systemen lassen sich im wesentlichen folgende vier Organisationsformen unterscheiden ([Inmo96] [MuBe98] [ScBa98]) (Abbildung 11):

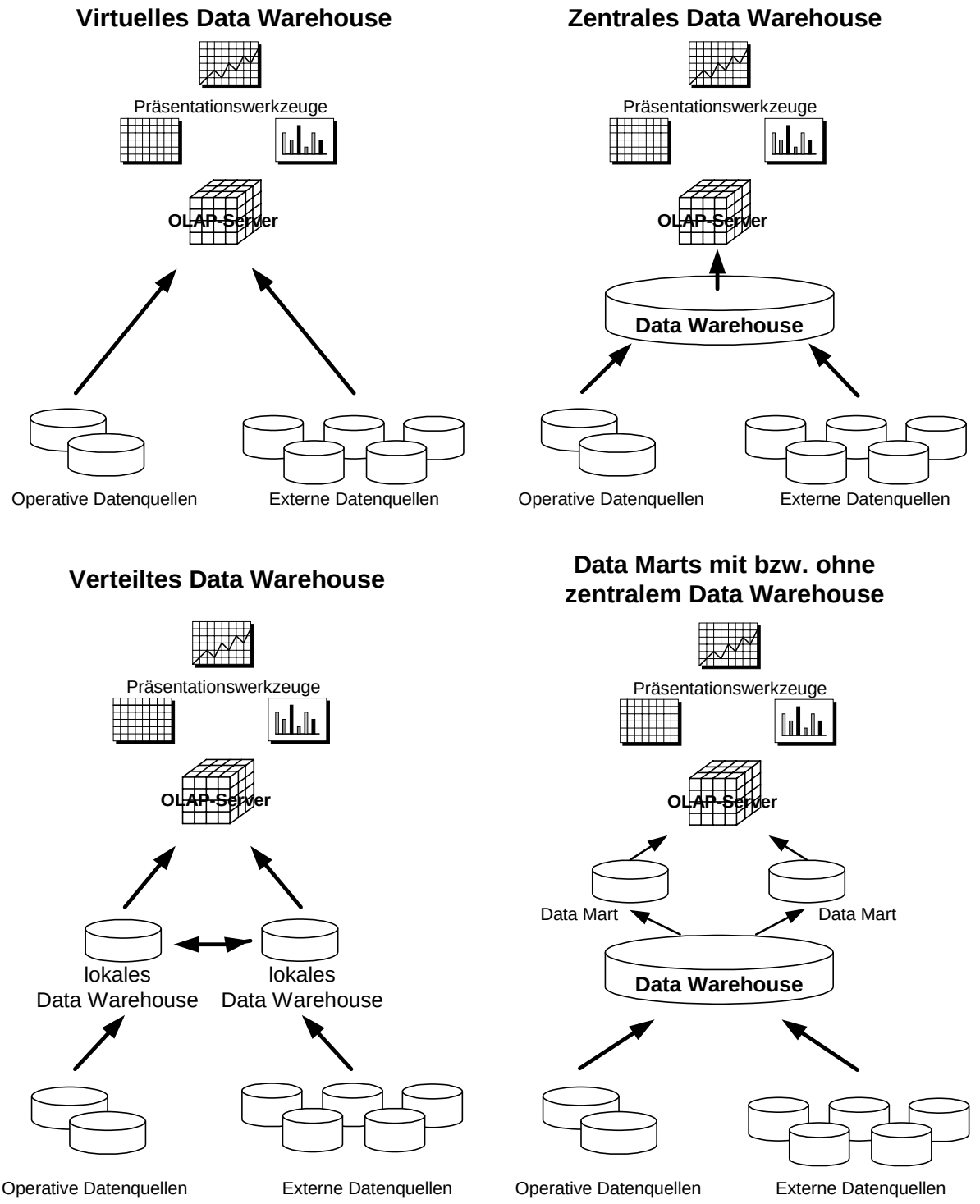

Abb. 11: Organisationsformen von Data Warehouse-Systemen

- *"Virtuelles Data Warehouse-System"*: Diese Organisationsform verzichtet im Gegensatz zu den übrigen Organisationsformen auf eine separate redundante Datenhaltung. Vielmehr werden die benötigten Daten unter direktem Zugriff auf die heterogenen, operativen Systeme gewonnen. Da nur eine geringe Infrastruktur benötigt wird, läßt sich diese Lösung schnell und kostengünstig realisieren; sie belastet jedoch die operativen Systeme. Eine umfassende Konsolidierung und Historisierung der Daten ist auf Basis der operativen Systeme nur begrenzt möglich.
- *Zentrales Data Warehouse-System*: In einem zentralen Data Warehouse werden alle Daten in einer zentralen physischen Datenbank verwaltet. Der Aufwand für die Bereitstellung und Pflege der zusätzlich redundant gehaltenen Daten wird aufgewogen durch die Möglichkeit, eine einzige historisierte und konsolidierte Datenbasis für das gesamte Unternehmen zu schaffen. Problematisch ist allerdings die inhärente Komplexität einer derartigen Gesamtlösung.
- *Verteiltes Data Warehouse-System*: Ein verteiltes Data Warehouse-System nutzt Mechanismen verteilter Datenbanksysteme, wobei für eine geeignete Synchronisation zwischen den einzelnen dezentralen Teilsystemen gesorgt werden muß. Weiterhin ist eine Aufteilung der Data Warehouse-Daten auf die Teilsysteme vorzunehmen. Der Einsatz verteilter Data Warehouse-Systeme bietet sich besonders bei stark dezentral organisierten Unternehmen an.
- *Data Mart mit oder ohne zentralem Data Warehouse*: Data Marts sind abteilungs- bzw. bereichsspezifische Data Warehouse-Lösungen. Sie erlauben den sukzessiven Aufbau von Data Warehouse-Systemen. Data Marts mit zentralem Data Warehouse ermöglichen die Replikation von themenspezifischen Data Warehouse-Daten für einzelne Gruppen von Entscheidungsträgern. Ohne zentrales Data Warehouse können allerdings Konsolidierungs- oder Synchronisationsprobleme auftreten.

6.3 Datenbereitstellung

In [CoBS98] werden die folgenden Endbenutzerwerkzeuge unterschieden:

- ***Bericht- und Anfragewerkzeuge:*** Dabei handelt es sich um Programme zur einfachen Erstellung von Berichten und zur Definition von Anfragen.
- ***Werkzeuge des Online Analytical Processing (OLAP):*** Als Standardendbenutzerschnittstelle haben sich im Data Warehouse-Bereich OLAP-Anwendungen etabliert. Sie ermöglichen die in Abschnitt 2 erläuterte multidimensionale Sicht auf die Daten im Data Warehouse. OLAP-Werkzeuge werden im weiteren Verlauf dieses Abschnitts ausführlich behandelt.
- ***Executive Information Systems (EIS):*** Ursprünglich wurden EIS-Tools als Werkzeuge zur Entscheidungsunterstützung des Top-Managements definiert. Allerdings weitete sich der Anwendungsbereich schnell auf das gesamte Management aus. Um eine Abgrenzung zwischen OLAP- und EIS-Tools zu erhalten, werden heutzutage fertige Anwendungssyteme mit

vordefinierten Berichten für bestimmte Betriebsbereiche, z.B. Verkauf, Marketing, Finanzen, als EIS-Tools bezeichnet. Der ORACLE Financial Analyser ist beispielsweise ein EIS-Tool [CoBS98].

- ***Data Mining Werkzeuge:*** Unter Data Mining versteht man den Prozeß des Auffindens von Beziehungen und Trends in großen Datenmengen. Dazu werden sowohl statistische, mathematische als auch Techniken aus dem Bereich der künstlichen Intelligenz (KI) eingesetzt. Diese Kategorie von Anwendungssystemen wird im weiteren Verlauf diese Artikels nicht weiter betrachtet. Allerdings wird davon ausgegangen, daß mit steigender Popularität von Data Warehouse-Systemen der Gebrauch dieser Anwendungssysteme weiter zunehmen wird.
- ***Werkzeuge zur Anwendungsentwicklung:*** Diese Kategorie von Präsentationswerkzeugen umfaßt alle Anwendungssysteme, die individuell für die Informationsauswertung erstellt wurden. Zur Entwicklung dieser Programme sind Anwendungsentwicklungswerkzeuge notwendig, die den Zugriff auf das Data Warehouse-System unterstützen. Häufig sind heutzutage jedoch bereits Anwendungsentwicklungswerkzeuge in gängigen OLAP-Tools integriert.

Im weiteren Verlauf dieses Abschnitts werden lediglich OLAP-Systeme behandelt. Diese ermöglichen den Anwendern die multidimensionale Sicht auf die Daten im Data Warehouse. OLAP-Produkte können hinsichtlich der verwendeten Technologie (Abschnitt 6.3.1) und der Client-/Server-Architektur (Abschnitt 6.3.2) unterschieden werden.

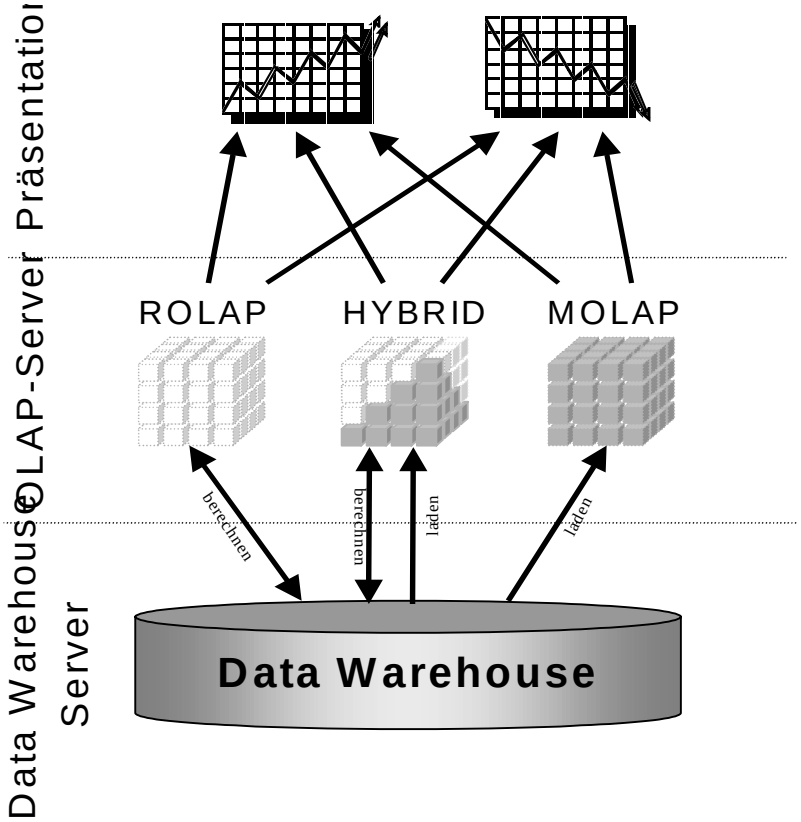

Abb. 12: OLAP-Technologien

6.3.1 OLAP-Technologien

Für OLAP-Produkte existieren zwei unterschiedliche Speichertechnologien, in denen die multidimensionalen Strukturen gespeichert werden ([ChDa97], [ChGl98], [DSHB98]): *relationale* und *multidimensionale Datenbanken*. Neben relationalen und multidimensionalen OLAP-Systemen gibt es jedoch zusätzlich noch Systeme, die beide Speichertechnologien einsetzen, sogenannte hybride OLAP-Systeme (Abbildung 12). Im folgenden werden die unterschiedlichen OLAP-Technologien genauer betrachtet:

- ***multidimensionales OLAP (MOLAP):*** MOLAP-Server ermöglichen die multidimensionale Sicht auf die Daten mittels multidimensionaler Speicherstrukturen bzw. multidimensionaler Datenbanksysteme (MDDBS). Aus der direkten multidimensionalen Abbildung der Daten resultiert eine sehr hohe Performance für die Anfragebearbeitung. Der Hauptnachteil von MOLAP-Systemen ist der große Speicherbedarf. Mit steigendem Speichervolumen nimmt die Anfrageperformance jedoch drastisch ab. In [ChGl98] wird eine Grenze von 20 Gigabyte genannt, ab der der Einsatz von MOLAP-Systemen nicht mehr sinnvoll ist. Darüberhinaus sind in multidimensionalen Würfeln häufig die Zellen nicht vollständig gefüllt (*sparcity*). In multidimensionalen Datenbanken müssen die leeren Zellen trotzdem gespeichert werden. Daher führen speziell dünnbesetzte Hypercubes zu einer großen Speicherplatzverschwendung. Moderne MOLAP-Systeme basieren jedoch auf zweistufigen Speicherstrukturen [Shos97], wodurch das Speichern von leeren Zellen mit Hilfe von Kompressionsverfahren umgangen wird. Ein weiteres Problem multidimensionaler OLAP-Systeme sind die zusätzlichen Ladelaufzeiten. Nach der Aktualisierung des Data Warehouses müssen zusätzlich die multidimensionalen Speicherstrukturen neu gefüllt werden. Es entfällt jedoch zumindest ein Teil der Aggregationsschicht (Abschnitt 6.2.1) des Warehouses, da diese Strukturen direkt im MDDBS implementiert werden. In [DSHB98] wird darüber hinaus die unzureichende Unterstützung folgender Datenbankfunktionalität als Schwachpunkte von multidimensionalen Datenbanksystemen gegenüber relationalen Datenbanksystemen angesehen: transaktionale Verarbeitung und Recovery, Versionierung, Realisierung von benutzerspezifischen Sichten und Verteilungsaspekte. Als weiterer Nachteil wird häufig der fehlende Standard für multidimensionale Datenbanken genannt und die bewußte Geheimhaltung der verwendeten internen Speicherstrukturen durch die Hersteller. Generell existieren die folgenden vier Grundkonzepte zur internen Speicherorganisation in MOLAP-Systemen: Vollarray mit direkter Adreßberechnung, Vollarray mit Leerstellensubstitution, indizierte Speicherung und verkettete Speicherung. Für eine ausführliche Gegenüberstellung dieser Konzepte sei auf ([Cham98], [ChGl98]) hingewiesen.
- ***relationales OLAP (ROLAP):*** ROLAP-Server transformieren die multidimensionalen Anfragen und Operationen der Endanwender in Konstrukte relationaler Anfragesprachen (i.d.R. SQL). Damit ermöglichen diese Systeme den Zugriff auf ein virtuelles multidimensionales Datenmodell, obwohl die relationale Datenbanktechnologie als Speichermedium verwendet wird. Allerdings benötigen die meisten Produkte spezielle Datenstrukturen, wie beispielweise Star- bzw. Snowflake-Schemata (Abschnitt 4.1). Statt zusätzliche Datenbanken anzulegen, werden diese Strukturen meist direkt im Aggregationsteil des Data Ware-

house gespeichert. Informationen über die Transformation zwischen multidimensionalem und relationalem Datenmodell wird in einem Metadatenmodell hinterlegt. In diesen Modellen ist die multidimensionale Struktur und deren relationale Repräsentation gespeichert. Es ist jedoch ersichtlich, daß die dynamische Transformation der ROLAP-Systeme zu erheblich längeren Anfragelaufzeiten führt als bei MOLAP-Systemen. Um diesen Umstand wettzumachen, werden spezielle Optimierungstechniken eingesetzt, wie beispielsweise die Verwendung spezieller Indeizes oder der Einsatz von Aggregationstabellen (Abschnitt 6.2.1). Da die komplexen Anfragen von relationalen Datanbankmanagementsystemen verarbeitet werden, müssen deren Query-Optimizer entsprechend angepaßt sein. Um einen optimalen Ausführungsplan zu gewährleisten, teilen manche ROLAP-Server (z.B. DSS Server von MicroStrategy) komplexe Anfragen in mehrere kleinere Teile auf (Multi-Pass-SQL). Die Ergebnisse der Teilanfragen werden zunächst in temporären Tabellen gespeichert und abschließend zum Endergebnis zusammengefaßt. Manche multidimensionalen Operationen können jedoch nicht optimal in SQL umgesetzt werden. Beispielsweise kann die *Top-N-Operation* (z.B. Anfrage über die drei meist verkauften Artikel) in Standard-SQL nicht adäquat umgesetzt werden. Solche Operationen müssen direkt in der ROLAP-Engine verarbeitet werden. Einige Datenbankhersteller statten bereits ihre relationalen Datenbanksysteme mit SQL-Erweiterungen aus, die solche Operationen beinhalten. Mit speziellen Datenbanktreibern können diese Spracherweiterungen von den ROLAP-Servern genutzt werden. Die Vorteile der relationalen OLAP-Technologie ergeben sich aus den Fortschritten, die im Bereich der relationalen Datenbanksysteme gemacht wurden. Es sind insbesondere die hohe Skalierbarkeit, die Unterstützung transaktionaler Konzepte, Versionierung und die Verwendung von benutzerspezifischen Sichten. Darüber hinaus entfällt die zusätzliche Datenhaltung und Aktualisierung außerhalb des Data Warehouses, wie es bei MOLAP-Systemen erforderlich ist.

- ***hybrides OLAP (HOLAP):*** HOLAP-Systeme verbinden die Vorteile von multidimensionalen und relationalen OLAP-Systemen. Normalerweise werden die Anfragen auf der Basis das Data Warehouses berechnet (ROLAP). Häufig benötigte Daten können jedoch zusätzlich in einer multidimensionalen Datenbank abgelegt werden. Dadurch sinkt die Anfragelaufzeit für diesen Teil der Daten beträchtlich. Die Kombination beider Technologien erlaubt eine optimale Konfiguration bezüglich der Skalierbarkeit, der Anfragelaufzeiten und des Aktualisierungsaufwands. Hierfür werden jedoch spezielle Administrationswerkzeuge benötigt.

6.3.2 Client-/Server-Architektur

OLAP-Systeme basieren immer auf einer logischen 3-Schichten Client-/Server-Architektur. Der Datenhaltungsteil entspricht dem Data Warehouse, der Anwendungsteil ist der OLAP-Server und die Präsentationswerkzeuge gehören zum Kommunikationsteil. Zusätzlich zur verwendeten Technologie kann jedoch die unterstützte physische Client-/Server-Architektur als Unterscheidungsmerkmal für OLAP-Produkte herangezogen werden. Das bedeutet die Möglichkeiten die Komponenten der logischen Client-/Server-Architektur auf physische Rechner zu

verteilen. Da sich dafür unterschiedliche Alternativen anbieten, wird in diesem Abschnitt lediglich die physische Client-/Server-Architektur mit den entsprechenden Vor- und Nachteilen betrachtet. Die folgenden Varianten können unterschieden werden (Abbildung 13):

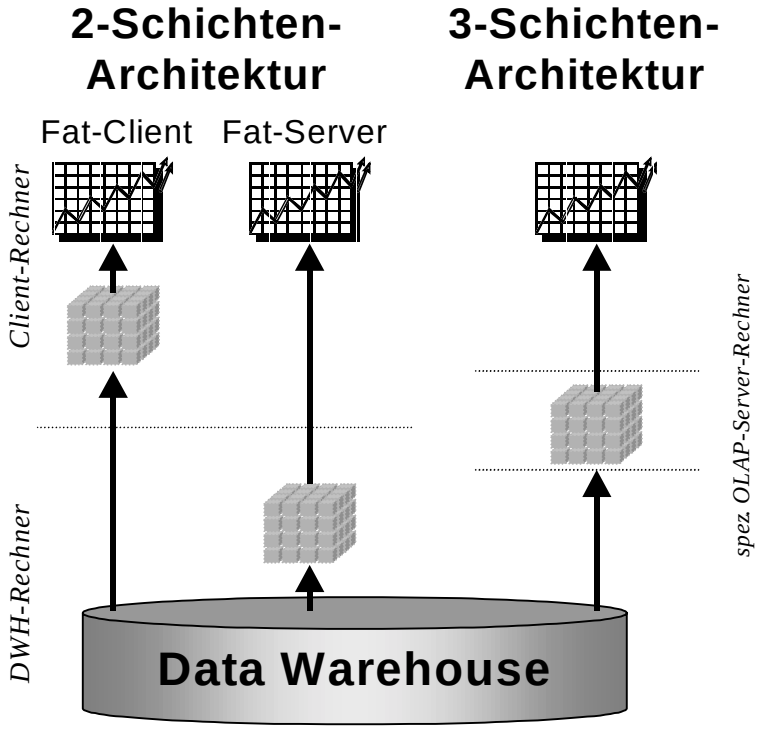

Abb. 13: Physische Client-/Server-Architekturen für OLAP

- ***2-Ebenen-Architektur:***
  - ***Fat-Client:*** Die Fat-Client-Architektur wird häufig auch als *Desktop-OLAP* (DOLAP) bezeichnet. Die komplette multidimensionale Verarbeitung erfolgt hierbei auf dem Rechner des Endanwenders. Neben dem erhöhten Ressourcenverbrauch auf dem Client-Rechner wird speziell beim Einsatz der MOLAP-Technologie zusätzlicher Speicher für die multidimensionalen Datenstrukturen benötigt, die lokal in multidimensionalen Datenbanken abgelegt sind. Zwar wird die Netzlast zwischen OLAP-Server und Präsentationswerkzeugen eliminiert, allerdings kann diese gegenüber dem Netzverkehr zwischen OLAP-Server und Data Warehouse generell vernachlässigt werden. Speziell die Beanspruchung des Netzes aufgrund der Kommunikation zwischen OLAP-Server und Data Warehouse ist einer der größten Nachteile dieser Architektur, insbesondere bei einer größeren Anzahl von Client-Rechnern, da für jeden Client ein eigener OLAP-Server zum Einsatz kommt. Weil bei der MOLAP-Technologie nur beim Aktualisieren auf das Data Warehouse zugegriffen werden muß, ist sie beim Einsatz einer Fat-Client-Architektur der ROLAP-Alternative vorzuziehen.
  - ***Fat-Server:*** In diesem Fall befindet sich der OLAP-Server mit dem Data Warehouse auf dem selben physischen Rechner. Der Hauptvorteil dieser Architektur ist die Eliminierung des Netzverkehrs zwischen Data Warehouse und OLAP-Server. Dadurch wird die Kommunikation zwischen diesen Komponenten erheblich beschleunigt. Dies wird insbesondere beim Einsatz der ROLAP-Technologie deutlich. Nachteilig ist jedoch der zusätzliche

Ressourcenbedarf auf der Seite des Data Warehouse-Servers. Zu der häufig sehr hohen Rechenlast verursacht durch das Data Warehouse (Aktualisierung, Verwaltung, Anfragebearbeitung) muß das Serversystem die Berechnung der multidimensionalen Anfragen bewältigen. Wird die MOLAP-Technologie eingesetzt, muß außerdem der zusätzliche Speicherbedarf berücksichtigt werden. Generell läßt die Fat-Server-Architektur keine Skalierung zu. Es kann nicht effizient auf sich ändernde Benutzerzahlen und -anforderungen reagiert werden, weil der Data Warehouse-Server als auch OLAP-Server auf einem einzigen Rechner platziert sind.

- ***3-Ebenen-Architektur:*** Die 3-Ebenen-Architektur macht eine optimale Konfiguration der Einzelbausteine - Präsentationswerkzeuge, OLAP-Server und Data Warehouse-Server - möglich. Dies führt zur größtmöglichen Skalierbarkeit des Systems. Es kann auf die technischen Anforderungen der einzelnen Komponenten entsprechend reagiert werden. Bei steigenden Benutzerzahlen kann zudem ein OLAP-Server je Benutzergruppe eingesetzt werden. Nebenbei erhält man dadurch eine bessere Ausfallsicherheit. Die Kommunikation zwischen Data Warehouse- und OLAP-Server kann durch physische Nähe und durch den Einsatz spezieller Kommunikationssysteme reduziert werden. Allerdings ist der größere Investitionsbedarf dieser Architektur zu berücksichtigen. Die 3-Ebenen-Architektur eignet sich insbesondere für sehr viele Endbenutzer.

Die meisten kommerziellen OLAP-Produkte unterstützen alle drei Architektur-Alternativen. Neben der Verwendung spezieller OLAP-Präsentationswerkzeuge wird heutzutage die Darstellung der Anfrageergebnisse im Internet immer wichtiger. Diese Technologie reduziert den Administrationsaufwand für die Client-Rechner und ermöglicht dadurch die Verwendung der OLAP-Technologie für sehr große Benutzergruppen. Die damit verbundene Erweiterung der Architektur auf mehr als drei Ebenen bietet zudem weitere Konfigurationsmöglichkeiten.

6.4 Administration

Auf allen Ebenen des Data Warehouse-Systems sind spezielle Werkzeuge zur Administration erforderlich (siehe Abbildung 9). Die zu unterstützenden Aufgaben können bezüglich der jeweiligen Ebene der Data Warehouse Architektur (Abbildung 9) kategorisiert werden. Hinzu kommen die Werkzeuge zur Unterstützung der Entwicklung des Data Warehouse-Systems:

- ***Konstruktion:*** z.B. Definieren des Data Warehouse-Schemas; Definieren der OLAP-Strukturen mit den entsprechenden Dimensionen und Hierarchien usw.
- ***Datenerfassung:*** z.B. Design und Anpassung von Extraktion-, Transformations- und Datenbereinigungsprozeduren; Überwachung der Aktualisierungsvorgänge und der Datenqualität usw.
- ***Datenhaltung:*** z.B. Überwachung des Speicherbedarfs; Konstruktion der Aggregationstabellen; Ausführung von Archivierungs- und Backup-Aufträgen usw.
- ***Datenbereitstellung:*** z.B. Benutzerverwaltung; Überwachung der Anfragelaufzeiten usw.

Der zentrale Bestandteil der Administration des Data Warehouse-Systems ist das Metadaten-Repository. Dieses enthält Informationen über die im Data Warehouse gespeicherten Daten. In [ChDa97] werden die folgen Typen von Metadaten unterschieden:

- ***Administrative Metadaten:*** Diese Klasse von Metadaten beinhaltet alle Informationen, die für die Entwicklung und Nutzung des Data Warehouses benötigt werden. Diese Daten können entsprechend dem Ort der Entstehung weiter untergliedert werden:
  - ***Konstruktion:*** z.B. Definition des Data Warehouse-Schemas; Definition der OLAP-Schemata mit den entsprechenden Dimensionen und Hierarchien usw.
  - ***Datenerfassung:*** z.B. Beschreibung der Datenquellen; Aktualisierungszeitpunkte; Datenbereinigungs- und Transformationsregeln usw.
  - ***Datenhaltung:*** z.B. verwendete Indexe; Definition der Aggregationstabellen; Partitionstabellen usw.
  - ***Datenbereitstellung:*** z.B. vordefinierte Berichte und Anfragen; Benutzerdaten und -profile mit den entsprechenden Berechtigungen usw.
- ***Operative Metadaten:*** Operative Metadaten sind Daten, die während des Betriebs des Data Warehouse-Systems angelaufen sind. Ebenso wie die administrativen Metadaten können diese entsprechend des Entstehungsorts klassifiziert werden:
  - ***Datenerfassung:*** z.B. Überwachungsdaten bei der Extraktion, Datenbereinigung und Datenübernahme; Änderungen in den Quellsystemen
  - ***Datenhaltung:*** z.B. Zustand des Data Warehouses, Überwachungsdaten über den Speicherbedarf
  - ***Datenbereitstellung:*** z.B. Benutzerstatistiken, fehlerhafte Anfragen, Informationsbedarf
- ***Business Metadaten:*** Diese Kategorie von Metadaten umfaßt Informationen über das Geschäftsfeld, wie beispielsweise Definitionen, Synonyme und Homonyme in diesem Geschäftsbereich, sowie dazu gehörenden Regeln für die semantische Interpretation.

Ein zentrales Metadaten-Repository wäre für eine effiziente und benutzerfreundliche Administration des Data Warehouses wünschenswert. Aufgrund eines fehlenden Standards verwenden die Hersteller der einzelnen Produkte jedoch individuelle Speicherstrukturen. Um den Austausch dieser Daten bzw. die Integration der Metadaten zu einem gemeinsamen Metadatenrepository zu ermöglichen, wurde die *Meta Data Coalition* (http://www.mdcinfo.com/) von einigen Herstellern gegründet. Im August 1997 spezifizierte diese Organisation die *Meta Data Interchange Specification* Version 1.1 [MeDC97].

7 Forschungsbedarf und Probleme des Data Warehousing

Die grundlegenden Prinzipien des Data Warehousing sind aus Fragestellungen der Praxis hervorgegangen und haben erst in den letzten Jahren verstärkt Einzug in den wissenschaftlichen Forschungsbereich gehalten. Bei einem Data Warehouse-System handelt es sich um kein

schlüsselfertig kaufbares Produkt. Vielmehr verkörpert ein Data Warehouse-System eine vielschichtiges und komplexes System, das an die jeweils vorliegende individuelle Problemstellung anzupassen ist. Folglich können auch unterschiedliche Fachgebiete, v. a. Informatik, Betriebswirtschaftslehre und Wirtschaftsinformatik, einen essentiellen Beitrag zum Gelingen einer umfassenden und interdisziplinären Data Warehouse-Lösung leisten. Im folgenden werden einige ausgewählte aktuelle Fragestellungen und Problemfelder kurz vorgestellt:

- Die Entwicklung von Mechanismen zur Integration von Daten und Schemata aus vorgelagerten operativen Systemen in eine konsolidierte Data Warehouse-Struktur und die damit verbundenen Probleme der Datenbereinigung werden zur Zeit intensiv diskutiert.
- Ein umfassender und ganzheitlicher Modellierungsansatz für Data Warehouse-Strukturen, der die konzeptuelle, logische und physische Entwurfsebene durchgängig berücksichtigt, fehlt (vgl. Abschnitt 4.1).
- Weiterhin besteht Forschungsbedarf bei einer geeigneten Repräsentationsform für ein Modell auf der konzeptuellen Entwurfsebene, das die Fachtermini multidimensionaler Modellierung für die Diskussion mit Entscheidungs- und Führungskräften adäquat aufbereitet.
- Ein Bezug zwischen multidimensionaler Modellierung und Geschäftsprozeßmodellierung wurde noch nicht hergestellt. Eine Einordnung in ein geschäftsprozeßorientiertes Vorgehensmodell, das zur Identifikation initialer Data Warehouse-Strukturen dienen kann, birgt großes Forschungspotential.
- Die Unterstützung des Data Warehouse-Modellierers durch geeignete computergestützte Werkzeuge ist als rudimentär einzustufen. Erste Ansatzpunkte sind bei den Werkzeugen ERwin von Platinum Technology, Data Warehouse Architect von Sybase und Constructa von Anubis zu finden.
- Spezielle Anforderungen an die Speicherung von Data Warehouse-Strukturen (vgl. Abschnitt 2) erfordern neue Indizierungs- und ausgefeilte Partitionierungsstrategien. Auf Data Warehouse-Strukturen ausgerichtete Indizierungsverfahren, wie z.B. Bitmap und Join-Indexe (vgl. Abschnitt 6.2.1), stellen nur einen ersten Schritt dar.
- Die Strukturdynamik bei Data Warehouse-Daten stellt ein weitgehend ungelöstes Problem dar. Erste Lösungsvorschläge wurden von [Kimb96b] („slowly changing dimensions“) und [ChSt98] („Temporale Aspekte in Data Warehouse-Systemen“) unterbreitet.
- Der Problembereich der materialiserten Sichten (vgl. Abschnitt 6.2.1) ist ein zentraler Forschungsschwerpunkt in der Informatik. Dabei stehen v.a. die Aspekte Auswahl, Aktualisierung und Verwendung von materialisierten Sichten im Mittelpunkt.

Auffällig ist die überwiegend statische Betrachtung des Data Warehouse-Konzeptes in der gängigen Literatur. Zu fordern ist jedoch eine stärker an den dynamischen Aspekte eines Data Warehouse-Systems ausgerichtete Betrachtungsweise. Diese muß sich an ständig ändernde Umweltfaktoren orientieren, wie z.B. wechselnder und wachsender Informationsbedarf eines Unternehmens.

8 Literatur

AHS+97 Altenpohl, U.; Huhn, M.; Schwab, W.; Zeh, T.: Datenmodellierung Data Warehouse - ein Lösungsvorschlag mittels ER-Modellierung, Guide Share Europe, 1997.

AnMu97 Anahory, S.; Murray, D.: Data Warehouse - Planung, Implementierung und Administration, Addison-Wesley, Bonn, 1997.

BeHo98 Becker, J.; Holten, R.: Fachkonzeptuelle Spezifikation von Führungsinformationssystemen, in: Wirtschaftsinformatik, 6/1998, S. 483-492.

BoUl99a Böhnlein, M.; Ulbrich-vom Ende, A.: Using the Conceptual Data Models of the Operational Information Systems for the Construction of Initial Data Warehouse Structures, in: Proceedings der Konferenz Modellierung betrieblicher Informationssysteme (MobIS'1999, Bamberg, 14.-15. Oktober), 1999.

BoUl99b Böhnlein, M.;Ulbrich-vom Ende, A.: Deriving Initial Data Warehouse Structures from the Conceptual Data Models of the Unterlying Operational Information Systems, in: Proceedings of the ACM Second International Workshop on Data Warehousing and OLAP (DOLAP'1999, Kansas City, USA, 6. November), 1999.

BuFo98 Bulos, D.; Forsman, S.: Getting Started with ADAPT - OLAP Database Design, Symmetry Corporation, 1998.

Bulo96 Bulos, D.: A New Dimension, in: Database Programming & Design, 6/1996.

CeMc95 Celko, J.; McDonald, J.: Don’t Warehouse Dirty Data, in: Datamation, Oktober 1995, http://www.datamation.com/PlugIn/issues/1995/oct15/10bsw100.html.

CeWi91 Ceri, S.; Widom, J.: Deriving Production Rules for Incremental View Maintenance, in: Proceedings of the 17th International Conference on Very Large Data Bases (VLDB’91, Barcelona, Spanien, 3.-6. September), 1991, S. 577-589.

Cham98 Chamoni, Peter: Entwicklungslinien und Architekturkonzepte des On-Line Analytical Processing. In: Chamoni, Peter/Gluchowski, Peter (Hrsg.): Analytische Informationssysteme. Data Warehouse, On-Line Analytical Processing, Data Mining, Berlin, Springer, 1998, S. 231-250.

ChDa97 Chaudhuri, S.; Dayal, U.: An Overview of Data Warehousing and OLAP Technology, in: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data (SIGMOD’97, Tucson, USA, 13.-15. Mai), 1997, S. 65-74.

ChGl98 Chamoni, P.; Gluchowski, P.: On-Line Analytical Processing (OLAP), In: Muksch H., Behme W. (Hrsg.): Das Data Warehouse-Konzept - Architektur, Datenmodelle, Anwendungen, 3. Auflage, Gabler, Wiesbaden, 1998, S. 401-444.

Chen76 Chen, P. P.-S.: The Entity-Relationship Model - Toward a Unified View of Data, in: ACM Transactions on Database Systems, Vol. 1, No. 1, 1976, S. 9-36.

ChSt98 Chamoni, P.; Stock, S.: Modellierung temporaler multidimensionaler Daten in Analytischen Informationssystemen, Proceedings des Workshops Data Mining und Data Warehousing der GI-Jahrestagung 1998 (Informatik'98, 28. Jahrestagung der Gesellschaft für Informatik, Magdeburg, 22. September),1998.

CKPS95 Chaudhuri, S.; Krishnamurthy, R.; Potamianos, S.; Shim, K.: Optimizing Queries with Materialized Views, in: Proceedings of the 11th International Conference on Data Engineering (ICDE 1995, Taipei, Taiwan, 6.-10. März), 1995, S. 190-200.

CoBS98 Connolly, T.; Begg, C.; Strachan, A.: Database Systems - A Practical Approach to Design, Implementation and Management, 2. Auflage, Addison-Wesley, Harlow, 1998

CoCS93 Codd, E.F.; Codd, S.B.; Salley, C.T.: Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate, E.F. Codd & Associates, White Paper, 1993.

DSHB98 Dinter, B.; Sapia, C.; Höfling, G.; Blaschka, M.: The OLAP Market: State of the Art and Research Issues, in: Proceedings of the ACM First International Workshop on Data Warehousing and OLAP (DOLAP 1998, Washington, D.C., USA, 7. November), 1998.

EhHe98 Ehrenberg, D.; Heine, P.: Konzept zur Datenintegration für Management Support Systeme auf der Basis uniformer Datenstrukturen, in: Wirtschaftsinformatik, 6/ 1998, S. 503-512.

FeSi98 Ferstl, O.K.; Sinz, E.J.: Grundlagen der Wirtschaftsinformatik Band 1, 3. Aufl., Oldenbourg, München, 1998.

GaGl97a Gabriel, R.; Gluchowski, P.: Management Support Systeme, Teil I, in: WiSt Heft 6, Juni 1997, S. 308-313.

GaGl97b Gabriel, R.; Gluchowski, P.: Semantische Modellierungstechniken für multidimensionale Datenstrukturen, in: HMD - Theorie und Praxis der Wirtschaftsinformatik 195, 1997, S. 18-37.

GHRU97 Gupta, H.; Harinarayan, V.; Rajaraman, A.; Ullman, J. D.: Index Selection for OLAP, in: Proceedings of the 13th International Conference on Data Engineering (ICDE 1997, Birmingham, U.K., 7.-11. April), 1997, S. 208-219.

GlGC97 Gluchowski, P.; Gabriel, R.; Chamoni, P.: Managment Support Systeme, Springer, Berlin, 1997.

GoMR98 Golfarelli, M.; Maio, D.; Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schemes, Proceedings of the Hawaii International Conference on System Sciences (HICS 1998, Kona, Hawai, 6.-9. Januar), 1998.

GPQ+97 Garcia-Molina, H.; Papakonstantinou, Y.; Quass, D.; Rajaraman, A.; Sagiv, Y.; Ullman, J.; Vassalos, V.; Widom, J.: The TSIMMIS approach to mediation: Data models and Languages, in: Journal of Intelligent Information Systems, Volume 8, Number 2, March/April 1997, S 117-132.

GuHQ95 Gupta, A.; Harinarayan, V.; Quass, D.: Aggregate-Query Processing in Data Warehousing Environments, in: Proceedings of the 21th International Conference on Very Large Data Bases (VLDB 1995, Zürich, Schweiz, 11.-15. September), 1995, S. 358-369.

GuMu95 Gupta, A.; Mumick, I.: Maintenance of Materialized Views: Problems, Techniques and Applications, in: IEEE Data Engineering Bulletin, Spezial Issue on Materialized Views & Data Warehousing 18(2), June 1995.

HaRU96 Harinarayan, V.; Rajaraman, A.; Ullman, J.: Implementing Data Cubes Efficiently, in: Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD 1996, Montreal, Canada, 4.-6. Juni), 1996, S. 205-216.

Holt97 Holthuis, J.: Multidimensionale Datenstrukturen - Modellierung, Strukturkomponenten, Implementierungsaspekte, in: Muksch, H./Behme, W. (Hrsg): Das Data Warehouse Konzept: Architektur - Datenmodelle - Anwendungen, 3. Aufl., Wiesbaden, Gabler, 1998, S. 143-193.

Info98 Informix: Warehouse Manager's Guide - MetaCube ROLAP Option for Informix Dynamic Server, Informix, 1998.

Inmo96 Inmon, W. H.: Building the Data Warehouse, Second Edition, Wiley & Sons, New York, 1996.

IRBS99 Inmon, W. H.; Rudin, K.; Buss, C. K.; Sousa, R.: Data Warehouse Performance, Wiley & Sons, New York, 1999.

JaGK96 Jahnke, B.; Groffmann, H.-D.; Gruppa, S.: On-Line Analytical Processing, in: Wirtschaftsinformatik, 38. Jg (1996), S. 321.-324.

KaRi87 Kähler, B.; Risnes, O.: Extending Logging for Database Snapshot Refresh, in: Proceedings of the 13th International Conference on Very Large Data Bases (VLDB 1987, Brighton, England, 1.-4. September), 1987, S. 389-398.

Kena95 Kenan Technologies: An Introduction to Multidimensional Database Technology, White Paper, Kenan Technologies, 1995.

Kimb96a Kimball, R.: The Data Warehouse Toolkit, Wiley & Sons, 1996.

Kimb96b Kimball, R.: Slowly Changing Dimensions, in: DBMS online, http://www.dbmsmag.com/9604d05.html.

Kimb97 Kimball, R.: A Dimensional Modeling Manifesto, in: DBMS online, http://www.dbmsmag.com/9708d15.html.

LaGa96 Labio, W. J.; Garcia-Molina, H.: Efficient Snapshot Differential Algorithms for Data Warehousing in Proceedings of the 22th International Conference on Very Large Data Bases (VLDB 1996, Bombay, Indien, 3.-6. September), 1996, S. 63-74.

LeMS95 Levy, A. Y.; Mendelzon, A.; Sagiv, Y.: Answering Queries Using Views, in: Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1995, San Jose, USA, 22.-25. Mai), 1995, S. 95-104.

LeSa93 Levy, A. Y.; Sagiv, Y.: Query Independent of Updates, in: Proceedings of the 19th International Conference on Very Large Data Bases (VLDB 1993, Dublin, Irland, 24.-27. August), 1993, S. 171-181.

LoDi87 Lockemann P.C., Dittrich K. R.: Architektur von Datenbanksystemen. In Lockemann P. C., Schmidt J.W. (Hrsg.): Datenbankhandbuch. Springer, Berlin, 1987, S. 85-161.

MaDL87 Mayr, H.C.; Dittrich, K. R.; Lockemann, P. C.: Datenbankentwurf, in (Hrsg: Lokkemann, P.C.; Schmidt, J.W.): Datenbankhandbuch, Springer, Heidelberg, 1987, S. 481-557.

McGu96 McGuff, F.: Hitchhiker's Guide to Decision Support, http://members.aol.com/fmcguff/dwmodel/frtoc.htm.

MeDC97 Meta Data Coalition: Meta Data Interchange Specification, Version 1.1, 1997, http://www.mdcinfo.com.

Mert95 Mertens, P.: Integrierte Informationsverarbeitung 1 - Administrations- und Dispositionssysteme in der Industrie, 10. Auflage, Gabler, Wiesbaden, 1995.

MuBe98 Muksch H., Behme W.: Das Data Warehouse-Konzept als Basis einer unternehmensweiten Informationslogistik. In: Muksch H., Behme W. (Hrsg.): Das Data Warehouse-Konzept - Architektur, Datenmodelle, Anwendungen. 3. Auflage, Gabler, Wiesbaden 1998, S. 33-100.

OLA95 OLAP Council: OLAP and OLAP Server Definitions, Whitepaper, 1995, http://www.olapcouncil.org/research/glossaryly.hm.

ONGr95 O'Neil, P.; Graefe, G.: Multi-Table Joins through Bitmapped Join Indeces, SIGMOD Record, Volume 24, Number 3, September 1995, S. 8-11.

ONQu97 O'Neil, P.; Quass, D.: Improved Query Performance with Variant Indices, in: Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD 1997, Tucson, USA, 13.-15. Mai), 1997, S. 38-49.

OrSö89 Ortner, E.; Söllner, B.: Semantische Datenmodellierung nach der Objekttypenmethode, in: Informatik-Spektrum, 12/1989, S. 31-42.

PeCr95 Pendse, N.; Creeth, R.: Synopsis of the OLAP Report, Business Intelligence, 1995, http://www.busintel.com.

Pilot98 Pilot Software: A Introduction to OLAP Multidimensional Terminology and Technology, White Paper, Pilot Software, http://www.pilot.sw.com/olap/olap.htm.

Poe96 Poe, V.: Building a Data Warehouse for Decision Support, Prentic Hall, New Jersey, 1996.

QRS+95 Quass, D.; Rajaraman, A.; Sagiv, Y.; Ullman, J.; Widom, J.: Querying Semistructured Heterogeneous Information, in: Proceedings of the 4th International Conference on Deductive and Object-Oriented Databases (DOOD 1995, Singapore, Singapore, 4.-7. Dezember), 1995, S. 319-344.

QuWi97 Quass, D.; Widom, J.: On-Line Warehouse View Maintenance for Batch Updates, in: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data (SIGMOD 1997, Tucson, USA, May 13-15), 1997, S. 393-404.

Rade95 Raden, N.: Star Schema 101, http://members.aol.com/nraden.str.htm.

Rade96 Raden, N.: Modeling the Data Warehouse, http://techweb.cmp.com/iw/564/64oldat.htm.

Raut98 Rautenstrauch, C.: Modellierung und Implementierung von Data-Warehouse-Systemen, Arbeitspapier, Otto-von-Guericke-Universität, Magdeburg, 1997.

SAP97 SAP AG: Business Information Warehouse - Technology, White Paper, SAP AG, 1997.

SAP98a SAP AG: Data Modeling with BW - ASAP for BW Accelerator, Business Information Warehouse Online Support Center, 1998.

SAP98b SAP AG: Business Information Warehouse, White Paper, SAP AG, 1998.

Sapi98 Sapia, C.: Babel Fish - A Model-Driven Data Warehouse Design Methodology - Konzeptuelle Datenmodellierung mit dem ME/R Modell, Vortrag auf dem 4. Workshop des GI Arbeitskreises Multidimensionale Datenbanken am 27.4.1998 in Darmstadt.

ScBa98 Schinzer H. D., Bange C.: Werkzeuge zum Aufbau analytischer Informationssysteme. In: Chamoni P., Gluchowski P. (Hrsg.): Analytische Informationssysteme - Data Warehouse, On-Line Analytical Processing, Data Mining. Springer, Berlin, 1998, S. 41-58.

Shos82 Shoshani, A.: Statistical Databases: Characteristics, Problems and some Solutions, in: Proceedings of the 8th International Conference on Very Large Data Bases (VLDB 1982, Mexico City, Mexico, 8.-10. Sept.),1982, S. 208-222.

Shos97 Shoshani, A.: OLAP and Statistical Databases: Similarities and Differences, in: Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1997, Tucson, USA, 13.-15. Mai), 1997, S. 185-196.

TeUl97 Teschke, M.; Ulbrich vom Ende, A.: Using Materialized Views to Speed Up Data Warehousing, Technical Report, IMMD 6, Universität Erlangen-Nürnberg, 1997.

TeUl98 Teschke, M.; Ulbrich vom Ende, A.: Concurrent Warehouse Maintenance without Compromising Session Consistency, Proceedings of the 9th International Conference on Database and Expert Systems Applications (DEXA'98, Wien, Österreich, 24.-28. August), 1998, S. 776-785.

ToJa98 Totok, A.; Jaworski, R.: Modellierung von multidimensionalen Datentstrukturen mit ADAPT - Ein Fallbeispiel, Arbeitsbericht des Insituts für Wirtschaftswissenschaften, Abteilung Controlling und Unternehmensrechnung, Braunschweig, 1998.

Toto97 Totok, A.: Data Warehouse und OLAP als Basis für betriebliche Informationssysteme, Arbeitsbericht des Insituts für Wirtschaftswissenschaften, Abteilung Controlling und Unternehmensrechnung, Universität Braunschweig, 1997.

Voss99 Vossen, G.: Datenbankmodelle, Datenbanksprachen und Datenbankmanagementsysteme, 3. Auflage, Oldenbourg, München, 1999.

WGLZ96 Wiener, J.; Gupta, H.; Labio, W.; Zhuge, Y.: A System Prototype for Warehouse View Maintenance, Technical Report, Department of Computer Science, Standford University, 1996.

YaLa87 Yang, H.Z.; Larson, P.A.: Query Transformations for PSJ Queries, in: Proceedings of 13th International Conference on Very Large Data Bases (VLDB 1987, Brighton, England, 1.-4. September), 1987, S. 245-254.

ZhGW96 Zhuge, Y.; Garcia-Molina, H.; Wiener, J.: The Strobe Algorithms for Multi-Source Warehouse Consistency. In: Proceedings of the 4th International Conference on Parallel and Distributed Information Systems (PDIS 1996, Miami Beach, USA, 18.-20. Dezember), 1996.

ZhWG97 Zhuge, Y.; Wiener, J; Garcia-Molina, H.: Multiple View Consistency for Data Warehousing. In: Proceedings of the 13th International Conference on Data Engineering (ICDE 1997, Birmingham, U.K, 7.-11. April), 1997, S. 289-300.