UNIVERSITÄT KOBLENZ · LANDAU

Institute for Management

Fachbereich 4: Informatik

Optimierung der Datenhistorie in einem Business Intelligence-System

Abbildung von Slowly Changing Dimensions

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science
im Studiengang Informationsmanagement

vorgelegt von

Waldemar Bergen

Betreuer: Dipl. Kfm. Werner Gauer, EDS Mid-market Solutions GmbH

Betreuer: Dr. Michael Möhring, Universität Koblenz-Landau, Institut für Wirtschafts- und Verwaltungsinformatik

Erstgutachter: ebenso

Zweitgutachter: Prof. Dr. Klaus G. Troitzsch, Universität Koblenz-Landau, Institut für Wirtschafts- und Verwaltungsinformatik

Koblenz, im Oktober 2008

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich — nicht — einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich — nicht — zu.

Koblenz, den 14. Okt. 2008 <Unterschrift>

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

Abb.	Abbildung
BI	Business Intelligence
BIS	Business Intelligence System
BSc	Bachelor of Science
BSC	Balanced Score Card
CRM	Customer Relationship Management
DML	Data Manipulation Language
DWH	Data Warehouse
ERP	Enterprise Ressource Planning
et al.	und andere
ETL	Extract-Transform-Load
KDD	Knowledge Discovery in Databases
MIS	Management Information System
MSS	Management Support System
MQE	Managed Query Environment
OLAP	Online Analytical Processing
OLTP	Online Transactional Processing
SQL	Structured Query Language
SSIS	SQL Server Integration Services®
SSMS	SQL Server Management Studio®

1 Einleitung

In diesem Kapitel wird der Leser zuerst zur Problemstellung hingeführt. Danach wird die Zielsetzung aufgeführt und näher erläutert. Anschießend wird der Aufbau der Arbeit beschrieben um eine einfachere Handhabe zu Gewährleisten und die thematische Entwicklung darzustellen.

1.1 Problemstellung

Die IT-basierte Managementunterstützung hat eine lange Historie. Bereits in den 60er Jahren des letzten Jahrhunderts begannen erste Versuche, die Führungskräfte mit Hilfe von Informationssystemen zu unterstützen. Diese ersten Ansätze scheiterten jedoch. Erst im Laufe der Jahre wurden Systeme entwickelt, die im Management eingesetzt werden konnten. In den 80er entwickelte sich für die Informations- und Kommunikationssysteme der Begriff „Management Support System (MSS)“. Obwohl es besonders im letzten Jahrzehnt umfangreiche technologische Entwicklungen im Bereich der IT-basierten Managementunterstützung gegeben hat, wird der Sammelbegriff „Management Support System“ vor allem in der Wissenschaft noch häufig verwendet. In der betrieblichen Praxis hat sich seit Mitte der 90er die Begrifflichkeit „Business Intelligence (BI)“ etabliert (Kemper et al., 2006, S. 1).

Heutzutage ist eine Firma ohne IT-System kaum mehr zu führen. Von kleinen Handwerkerbetrieben, bis zu multinationalen Konzernen kann kein Unternehmen mehr auf die Hilfe von modernen Informationssystemen verzichten. Neben den Warenwirtschaftssystemen oder Produktionsplanungssystemen, die an den betrieblichen Transaktionen orientiert sind, setzen immer mehr Unternehmer für die Unterstützung ihrer Entscheidungen die eben angesprochenen analyseorientierten BI-Systeme ein (Gauer, 2006, S. 1).

Damit BI-Systeme überhaupt eine Unterstützung sein können, stellen sie u.a. historische Unternehmensdaten schnell und übersichtlich zur Verfügung. Mit diesen historischen Daten wird dem Management eine Grundlage für ihre Entscheidung geliefert. Dabei ist es essentiell, dass die vom BI-System zur Verfügung gestellten Daten zuverlässig sind. Diese Zuverlässigkeit äußert sich nicht nur in der Korrektheit von Fakten, sondern erfordert auch, dass Dimensionen richtig und verlässlich aufgeführt sind. In der Praxis verhält es sich jedoch so, dass die Dimensionen sich langsam verändern bzw. ihre Werte wechseln. Um Auswertungen über längere Zeiträume zu gewährleisten, innerhalb welcher sich die Dimensionen geändert haben, ist in vielen Fällen die Dokumentation von strukturellen

und inhaltlichen Veränderungen in der Datenhaltung erforderlich (Kemper et al., 2006, S. 78). Es ist denkbar, dass die sich veränderte Dimension einfach als neue Dimension aufgeführt werden könnte. Jedoch gibt es Abfragen bei denen die Veränderung der Dimension keine Rolle spielt bzw. die alte und neue Dimension als Einheit gesehen werden muss. Dabei kommt es auf eine korrekte Modellierung der Datenbank im Data Warehouse an. Ziel muss sein, das Data Warehouse und damit das BI-System so zu konfigurieren, dass selbst sich langsam verändernde Dimensionen, im angelsächsischen Sprachgebrauch auch „Slowly Changing Dimensions“ genannt, zu jeder Zeit nachvollzogen bzw. die Fakten zur alten und neuen Dimension je nach Bedarf gemeinsam oder von einander getrennt verwendet werden können.

1.2 Zielsetzung

Auf den Erkenntnissen der Problemstellung aufbauend, kann das Hauptziel dieser Ausarbeitung mit der Sicherstellung einer korrekten Historie in einem Business Intelligence System bei sich langsam verändernden Dimensionen beschrieben werden.

Zur Erreichung des Hauptziels lassen sich folgende Teilziele abgrenzen: Zum einen soll zuallererst einiges Grundlegendes zu BI-Systemen, die eine Datenhistorie bereitstellen, beschrieben werden. Hierbei wird auch motiviert wie solche Systeme eingesetzt und verwendet werden können. Außerdem soll der Unterschied zu transaktionsorientierten ERP-System deutlich werden. Das zweite Teilziel beschreibt die Anwendungen einer Datenhistorie, die den Entscheidern eine Grundlage und Basis für ihre Entscheidungen liefert. Mit diesen beiden Teilzielen als Grundlage, wird als Hauptziel dargestellt, das selbst in einem BI-System, das ja eigentlich schon auf historischen Daten gründet, Historisierungskonzepte nötig sind. Vor allem der Umgang mit Slowly Changing Dimensions wird hier beschrieben und erläutert. Auch soll hier anhand eines Beispiels gezeigt werden, wie dieses bewerkstelligt werden kann.

1.3 Aufbau der Arbeit

Die oben beschriebenen Zielsetzungen bewirken eine Gliederung in fünf Kapitel (siehe auch Abb. 1):

Das erste Kapitel führt in das Thema der Bachelorarbeit ein; das Hauptziel, Teilziele und der Aufbau der Arbeit werden dargelegt. Im zweiten Kapitel werden die technischen Grundlagen erläutert, die als Basis der Arbeit betrachtet werden. Nach der Definition von BI-Systemen und der Abgrenzung von ERP-Systemen wird auf die einzelnen Teilbereiche des BI-Prozesses eingegangen. Dabei werden unter anderem der ETL-Prozess, das Data Warehouse und das Online Analytic Processing (OLAP) beschrieben. In Kapitel drei werden Anwendungen aufgeführt, die dazu beitragen, dass Vergangenheitsdaten eingesetzt werden können um einen eventuellen Informationsvorsprung gegenüber Konkurrenten, Debitoren und Kreditoren zu bekommen. Die Optimierung einer Datenhistorie in einem BI-System durch die Realisierung und Modellierung von Slowly Changing Dimensions wird im vierten Kapitel dargestellt.

Den Abschluss der Arbeit bildet Kapitel fünf mit einer Zusammenfassung der Inhalte und einem abschließenden Fazit.

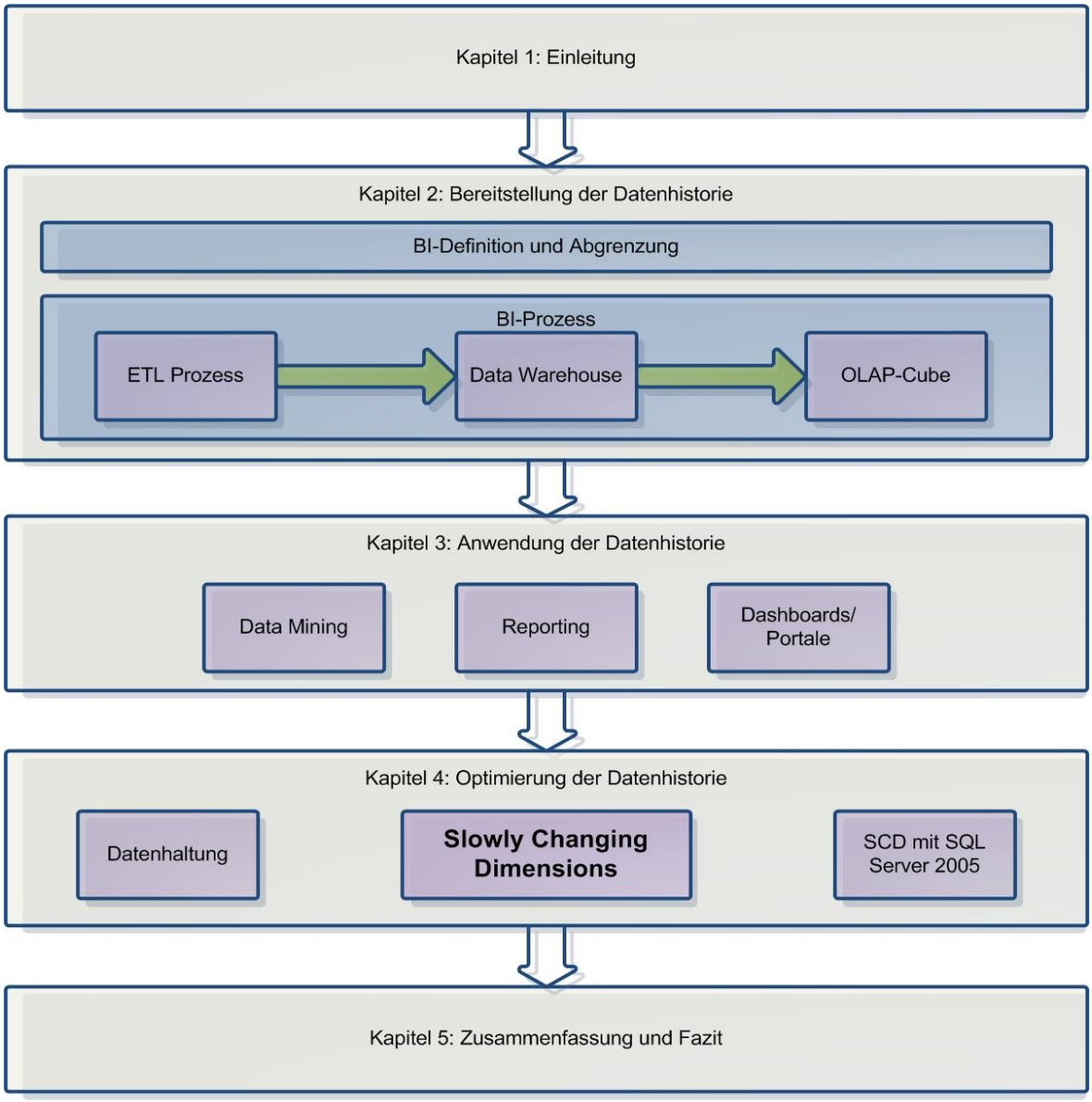

Abb. 1: Aufbau der Arbeit

(Quelle: eigene Darstellung)

2 Bereitstellung der Datenhistorie

In diesem Kapitel werden die essentiellen Grundlagen zu Business Intelligence Systemen dargestellt, die Daten bzw. eine Datenhistorie für Anwender bereitstellen. Die Daten müssen aufbereitet und visualisiert werden. Zuerst ist es nötig, den Begriff „BI“ zu definieren und von den vielen verschiedenen Begrifflichkeiten die im Business Intelligence Umfeld gebräuchlich sind, abzugrenzen. Dann wird ein sogenannter BI-Prozess beschrieben, bei dem Rohdaten nach und nach an Qualität gewinnen, um schließlich als Information und Wissen im Unternehmen zu dienen.

2.1 Definition und Abgrenzung

Dieser Abschnitt soll, wie bereits angedeutet, eine Grundlage zum Verständnis der Begrifflichkeit „Business Intelligence (BI)" dienen. In der modernen Wirtschaft kursieren die verschiedensten Namen, mit welchen häufig ähnliche Produkte und Dienstleistungen gemeint sind. Dabei sind viele Bezeichnungen lediglich Ideen und Konzepte von Vertriebsleitern und Marketern. Zuvor wird jedoch eine Abgrenzung von ERP-Systemen durchgeführt.

2.1.1 Abgrenzung BI – ERP-Systeme

Ein Enterprise-Ressource-Planning-System (ERP-System) deckt Funktionen aus mehreren Unternehmensbereichen ab. Die häufigsten Einsatzbereiche in Unternehmen sind: Fertigung, Vertrieb, Rechnungswesen, Finanzwesen und Personalwesen. Die Hauptaufgaben sind dabei die Administration, Disposition, Information und die Analyse. Das wesentliche Merkmal von ERP-Systemen, ist die Integration von den verschiedenen Funktionen, Aufgaben und Daten in ein Informationssystem. Die vielen Anwendungsfunktionen bauen auf einer Datenbank auf und bilden Geschäftsprozesse, auch über Abteilungsgrenzen hinaus, ab. Wesentliche Vorteile von ERP-Systemen werden in der Automatisierung und Standardisierung von betrieblichen Abläufen und Prozessen gesehen (Gronau, 2004, S. 3).

Jedoch können ERP-Systeme nicht den unternehmerischen Weitblick liefern, da der Informationsmangel zu erheblich ist. Die nötigen Daten sind nur sehr aufwändig aus einem ERP-System herauszuholen und für Managemententscheidungen und -prozesse wenig geeignet. Der Grund hierfür liegt vor allem in der Ausrichtung der der IT-Unterstützung auf verschiedene logistische Prozesse, die unterschiedliche betriebswirtschaftliche Bedeu-

tungen haben. Daraus folgernd können diese Daten nicht für eine „maschinelle Auswertung“ herangezogen werden, die für Managemententscheidungen als Grundlage dienen könnten. Dies ist jedoch wenig überraschend, denn aus Informationssicht, sind Logistik und Unternehmensführung grundsätzlich unterschiedlich (Kaiser, 2006, S. 3).

Weiterhin muss hier auch die Frage nach den Anwenderwünschen gestellt werden. Sachbearbeiter im Logistik-Bereich benötigen beispielsweise genaue Daten und Informationen, während für das Management nicht alle möglichen Details zur Analyse nötig sind, sondern lediglich die Daten des Entscheidungsgebiets bzw. die Daten, die einen Überblick liefern und als Grundlage für Entscheidungen dienen können.

Anfragen	**transaktional**	**analytisch**
Fokus	Lesen, Schreiben, Modifizieren, Löschen	Lesen, (periodisches) Hinzufügen
Transaktionsdauer und -typ	kurze Lese-/ Schreibtransaktionen	lange Leseaktionen
Datenvolumen einer Anfrage	wenige Datensätze	viele Datensätze/ bzw. viele Datensätze zu wenigen kumuliert
Daten	**transaktional**	**analytisch**
Datenquellen	meist eine	mehrere
Eigenschaften	nicht abgeleitet, zeitaktuell, autonom, dynamisch	abgeleitet, konsolidiert, historisiert, integriert, stabil
Datenvolumen	Megabyte – Gigabyte	Gigabyte – Terrabyte
Anwender	**transaktional**	**analytisch**
Anwendertyp	Ein-/Ausgabe durch Sachbearbeiter	Auswertung durch Manager, Controller, Analysten
Anwenderzahl	sehr viele	eher wenig

Abb. 2: Gegenüberstellung von transaktionalen und analytischen Systemen

(Quelle: modifiziert übernommen aus: Bauer et al., 2004, S. 9f)

Generell lassen sich ERP-Systeme, oder transaktionale bzw. operative Systeme, oft auch als Online Transactional Processing (OLTP) bezeichnet, von analyseorientierten managementunterstützenden Systemen vor allem in den Bereichen Anfragen, Daten und Anwender unterscheiden. Abb. 2 stellt die wichtigsten Klassifizierungen dar. So liegt der Fokus von transaktionalen Systemen, die auch operative Systeme genannt werden, auf einem Schreib- und Lesezugriff, während bei analytischen Systemen der Fokus auf dem Lesen und periodischen Hinzufügen von Daten liegt. Die Quelldaten für operative Systeme befinden sich meist in einer Datenbank, während analytische Systeme viele Quellen benutzen, um eine möglichst objektive Entscheidungsgrundlage zu liefern. Fast schon selbster-

klärend ist, dass transaktionale Systeme eher von Sachbearbeitern und analytische Systeme von Führungskräften benötigt werden(Bauer et al., 2004, S. 8f).

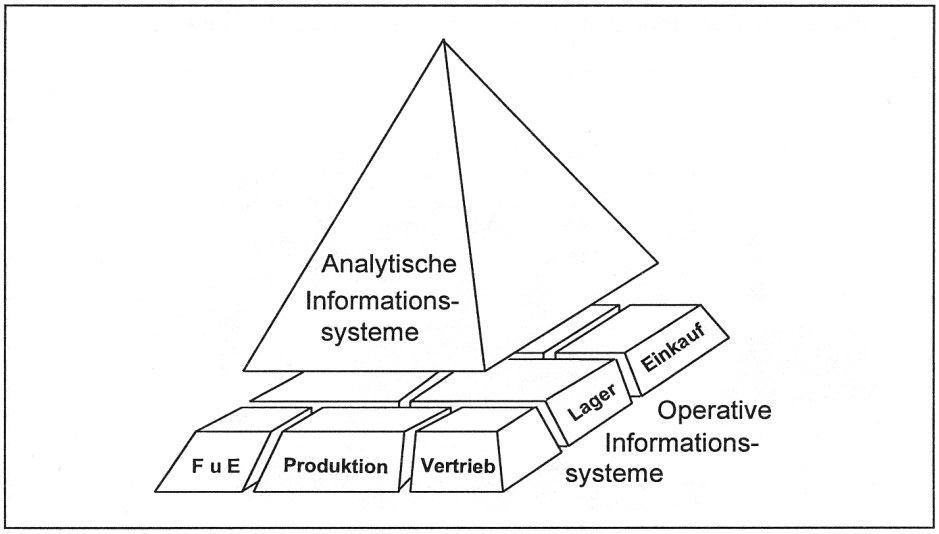

Abb. 3: Analytische Informationssysteme vs. Operative Informationssysteme

(Quelle: Chamoni et al., 1998, S. 11)

Inhaltlich bilden analytische Informationssysteme die logische Erweiterung zu operativen Informationssystemen und sind somit Grundlage und Datenquelle von diesen, wie Abb. 6 deutlich veranschaulicht (Chamoni et al., 1998, S. 11).

2.1.2 Definition der Begrifflichkeit „Business Intelligence“

Wie der Leser zu diesem Zeitpunkt sicherlich schon feststellen musste, sind in der Wissenschaft und in der Wirtschaft viele Begriffe geläufig, die alle ähnliche Produkte, Systeme oder Umgebungen beschreiben. Für diese Arbeit soll der BI Begriff nun definiert werden, um sicherzustellen, dass Leser und Autor dasselbe Verständnis dieses Begriffes haben.

Der Begriff „Business Intelligence“ entstand in den 90er Jahren, als sich der IT-Markt gravierend veränderte. So revolutionierten die zunehmende IT-Unterstützung unternehmensspezifischer Geschäftsprozesse und die weltweite, kommerzielle Nutzung der Internettechnologie die Möglichkeiten der Managementunterstützung. Um den geänderten Rahmenbedingungen Rechnung zu tragen entstand diese Begrifflichkeit, die maßgeblich auf Überlegungen der Gartner Group[1] zurückzuführen ist (Gluchowski et al., 2006, S. 12).

[1] Die Gartner Group ist ein führender Anbieter von Marktforschung und Analyse in der internationalen Technologie-Industrie.

So stellten sie im Jahre 1996 fest: “Data analysis, reporting and query tools can help business users wade through a sea of data to synthesize valuable information from it – today these tools collectively fall into a category called 'Business Intelligence'”(Anandarajan et al., 2004, S. 18f).

Mittlerweile hat sich dieser Begriff in der IT-Landschaft fest etablieren können, auch wenn keine einheitliche, allgemein anerkannte Definition existiert und die Begriffsverwendung sehr unterschiedlich ist. Die Einordnung und Definition erweist sich auch nicht als trivial, weil jede Definition angreifbar bleibt (Gluchowski, 2001, S. 5f). Allgemein kann man Business Intelligence Systeme als Systeme bezeichnen, „die auf der Basis interner Kosten und Leistungsdaten, sowie externer Marktdaten in der Lage sind, das Management in seiner planenden, steuernden und koordinierenden Tätigkeit zu unterstützen (Chamoni et al., 2004, S. 119).

Um die Bedeutung des BI-Begriffs zu verdeutlichen und zu präzisieren, hat Mertens bei einer Untersuchung gängiger BI-Abgrenzungen in Fachliteratur und kommerziellen Prospekten, folgende sieben unterschiedlich Möglichkeiten des BI-Verständnis identifiziert (Mertens, 2002, S. 4):

1. BI als Fortsetzung der Daten- und Informationsverarbeitung: IV für die Unternehmensleitung
2. BI als Filter in der Informationsflut
3. BI = Management Information System (MIS), aber besonders schnelle/ flexible Auswertungen
4. BI als Frühwarnsystem (“Alerting“)
5. BI = Data Warehouse
6. BI als Informations- und Wissensspeicherung
7. BI als Prozess: Symtomerhebung → Diagnose → Therapie → Prognose → Therapiekontrolle

Eine treffende Strukturierung der möglichen Sichtweisen liefert Gluchowski mit Hilfe eines zweidimensionalen Ordnungsrahmens (vgl. Abb. 4). Auf der vertikalen Seite werden die Datenverarbeitungsprozesse von der Bereitstellung bis zur Auswertung aufgetragen, während die horizontale Achse den Schwerpunkt zwischen Technik- und Anwendungsorientierung definiert. So befinden sich im oberen Teil der Abbildung die Ansätze, die eine reine Speicherung und Bereitstellung analyserelevanter Daten abdecken. Im unteren Bereich sind die Aspekte dargestellt, bei denen die methodische Komponente stärker im

Vordergrund steht und die das zuvor bereitgestellte Datenmaterial eher als Ausgangspunkt für weiterführende Analysen nutzen. Der mittlere Bereich ist durch Werkzeuge geprägt, die das Datenmaterial mit vergleichsweise wenigen eigenen Aufbereitungsfunktionen anzeigen können (Gluchowski, 2001, S. 8).

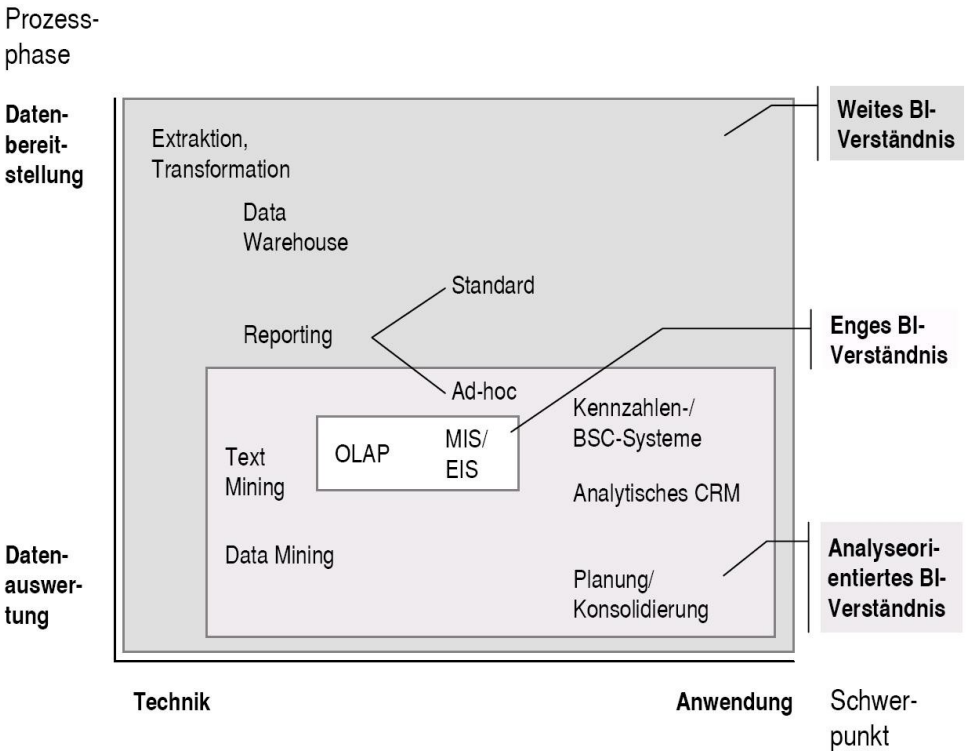

Abb. 4: Unterschiedliche Facetten von Business Intelligence

(Quelle: Kemper et al., 2006, S. 4, dort modifiziert übernommen aus Gluchowski, 2001, S. 7)

Außerdem ergeben sich durch diese Art der Gliederung drei gängige Typen von Definitionsansätzen(Kemper et al., 2006, S. 3f):

Weites BI-Verständnis

Unter diesem Definitionsansatz werden alle direkt und indirekt für die Entscheidungsunterstützung eingesetzten Anwendungen verstanden. Dieses beinhaltet neben den Auswertungs- und Präsentationsfunktionen auch die Datenaufbereitung und -speicherung.

Analyseorientiertes BI-Verständnis

Business Intelligence im analytischen Sinne umfasst alle Anwendungen, bei denen der Entscheider direkt mit dem System arbeitet, also einen direkten Zugriff auf eine Benutzerfläche mit interaktiven Funktionalitäten hat. Hierzu gehören neben OLAP auch Systeme des Text Mining und des Data Mining, sowie der Bereich des analytischen Customer Relationship Management und einige weitere Anwendungen.

Enges BI-Verständnis

Hiermit sind nur die Kernanwendungen gemeint, die eine Entscheidungsfindung unmittelbar unterstützen. Das sind vor allem das OLAP und die Management Information Systems.

Kemper et al. sehen nach der Auseinandersetzung mit der BI-Definitionsvielfalt in BI-Systemen einen integrierten Gesamtansatz, der die Einzelsysteme ersetzten soll, um den immer höher werdenden Anforderungen an die Managementunterstüzung gerecht zu werden. Vor allem betonen sie, dass punktuelle Lösungsansätze nicht ausreichend sind, da sie nur einzelne Aspekte behandeln und häufig auf isolierten Datenbasen aufbauen. Somit definieren sie Business Intelligence als einen:

> ***„integrierten, unternehmensspezifischen IT-Gesamtansatz zur betrieblichen Entscheidungsunterstützung*** *(Kemper et al., 2006, S. 8)****".***

In Abgrenzung zu vielen anderen Definitionen dienen BI-Werkzeuge bei diesem Ansatz ausschließlich als Entwicklungshilfen von speziellen BI-Anwendungen. Damit meinen sie, dass beispielsweise Tools zum Aufbau von Data Warehouses, OLAP-Front-Ends oder Portalsoftware lediglich unterstützenden oder mittelbaren Charakter haben. Außerdem stellen sie fest, dass einzelne Anwendungssysteme wie Data Mart-basierte Controllinganwendungen oder CRM-Lösungen jeweils nur Teilaspekte des BI-Gesamtansatzes abbilden (Kemper et al., 2006, S. 7f).

Dieser von Kemper et al. entwickelte integrierte Gesamtansatz stellt eine Lösung dar, der Business Intelligence als eigenständiges Konzept der Managementunterstützung darstellt und sich qualitativ von althergebrachten Ansätzen unterscheidet. Für den weiteren Verlauf dieser Arbeit soll diese Definition als Grundlage verstanden und verwendet werden von welcher ausgehend weitere Einzelheiten beschrieben und diskutiert werden sollen.

2.2 BI-Prozess

In diesem Abschnitt wird der Prozess der Informationsgenerierung beschrieben. Dabei geht es vor allem um die einzelnen Stationen, welche die Daten aus einem transaktionsorientierten ERP-System durch das BI-System hindurch durchlaufen, bis sie als Entscheidungsgrundlage in optisch hochwertiger Form für die Entscheider zur Verfügung stehen.

Die einzelnen Stationen oder Phasen des BI-Prozesses bauen aufeinander auf bzw. greifen ineinander ein. Auf diese Weise bilden sie die Architektur des Business Intelligence „Hauses“, welches in Abb. 5 dargestellt ist. Das Architekturschema entspricht einem Schichtenmodell. Die Abhängigkeit der Bausteine besteht darin, dass die Systeme der jeweiligen Schicht ihre Aufgabe als Voraussetzung für die Arbeitsgänge oder die Operationen der nächst höheren Schicht erledigen müssen. Der ETL-Prozess schafft durch den Export, beispielsweise aus einem ERP-System, und die Ablage der operativen Daten im Data Warehouse und den Data Marts ein solides Fundament des BI-Hauses.[2] Darauf aufbauend kann das, was den Begriff „Intelligence“ ausmacht, stattfinden, also die Generierung von Informationen, bzw. das Gewinnen von Erkenntnissen, welche durch die Technologien der obersten Schicht dem Anwender präsentiert werden.

Die Meta-Daten[3] fungieren als Säule des BI-Hauses, da alle Schichten auf sie zurückgreifen und von ihnen abhängig sind. Die Darstellung der Anwendungsbereiche als Säule zeigt einerseits, dass jede Technologie ohne Verwendungszweck nicht viel Sinn macht. Anderseits verdeutlicht die vertikale Anordnung, dass sie alle Ebenen der BI-Konzepte gleichermaßen verwenden (Bäumer, 2006, S. 7f).

[2] Der ETL-Prozess gehört nach der Definition von Kemper auch schon zum BI-System (vgl. Kapitel 2.1.2), denn Tools zum Aufbau von Data Warehouses gelten als Entwicklungshilfen von speziellen BI-Anwendungen. Hier muss der Leser davon ausgehen, dass das Fundament (der ETL-Prozess) zum Haus (das BI-System) dazugehört.

[3] Metadaten sind Informationen über andere Daten z.B. die Information über die Zugriffsrechte oder die letzte Änderung einer Datei. Dabei ist die Datei die eigentliche Information und die Zugriffsrechte und das letzte Änderungsdatum sind Metadaten.

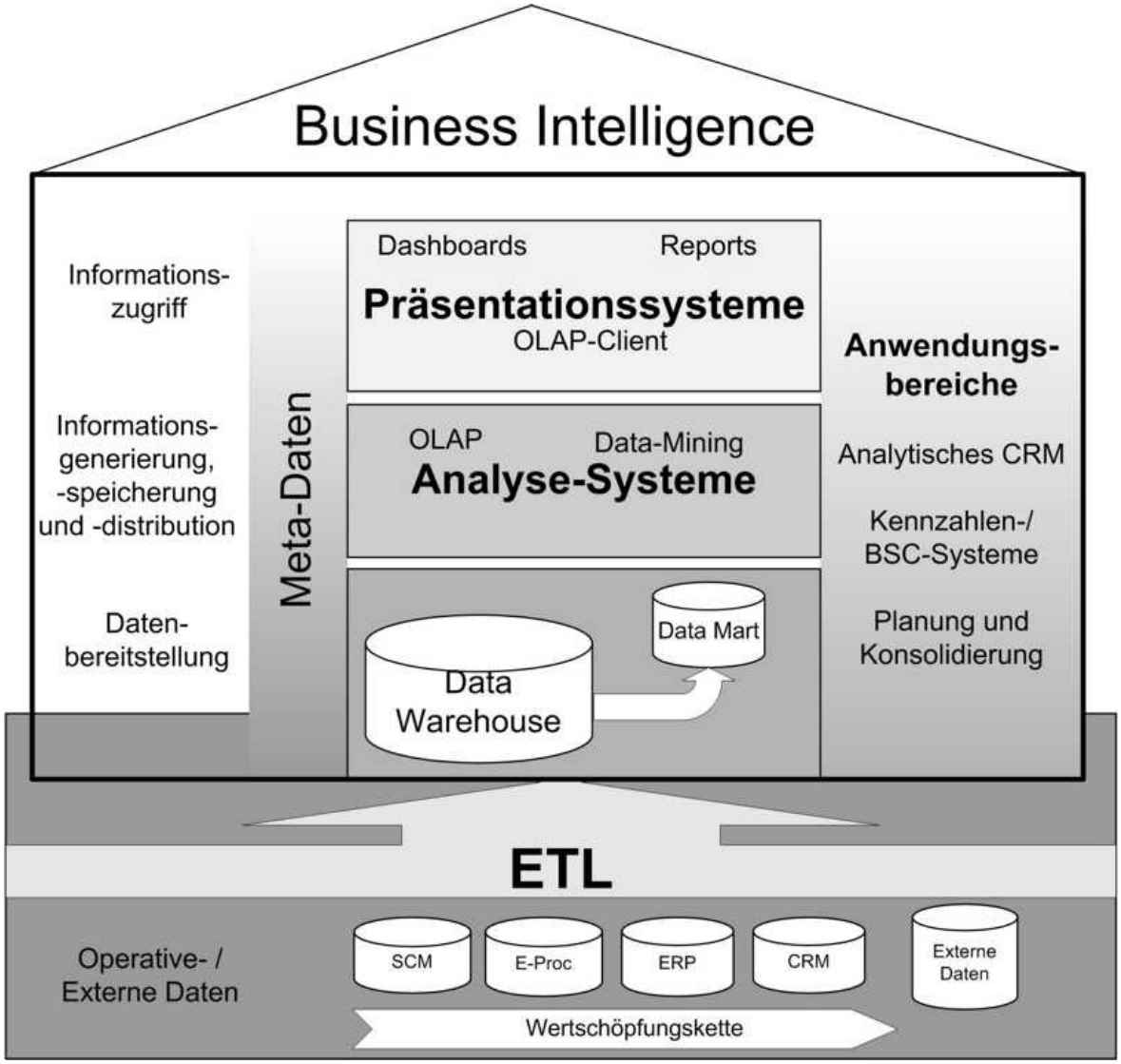

Abb. 5: Business Intelligence Architektur

(Quelle: Bäumer, 2006, S. 9)

2.2.1 Extract-Transform-Load-Prozess (ETL-Prozess)

Um Daten in ein BI-System zu laden, müssen sie aus einem operativen System extrahiert werden. Da Informationen meist aus mehreren Quellsystemen geladen werden, müssen die Daten der jeweiligen Systeme aneinander angepasst werden. Anschießend werden die Daten in ein Data Warehouse geladen. Dieser Prozess wird als ETL-Prozess oder auch als Datenbeschaffungsprozess bezeichnet. Abb. 6 veranschaulicht diesen Vorgang.

Wichtig ist an dieser Stelle zu wissen, dass managementunterstützende Systeme nur in Ausnahmefällen direkt auf die operativen Daten aufsetzen, da wichtige Historienbetrachtungen in ERP-Systemen nicht möglich sind, und die Daten aus diesen Systemen nur mit umfangreichen Transformationsregeln in brauchbare Managementinformationen umge-

wandelt werden können (Kemper et al., 2006, S. 23). Deshalb ist es nötig die Daten zentral in einem Data Warehouse zu speichern, auf welchen das System aufsetzen kann. Außerdem hat das operative System dadurch nicht unter Performance Einbußen zu leiden.

Die einzelnen Schritte des ETL-Prozesses, der die Daten aus den operativen Systemen in das Data Warehouse lädt, sollen jetzt erläutert werden. Auch hierbei liegt das Schichtenmodell zugrunde, bei dem die vorläufigen Prozesse als Grundlage der nachläufigen Prozesse gelten.

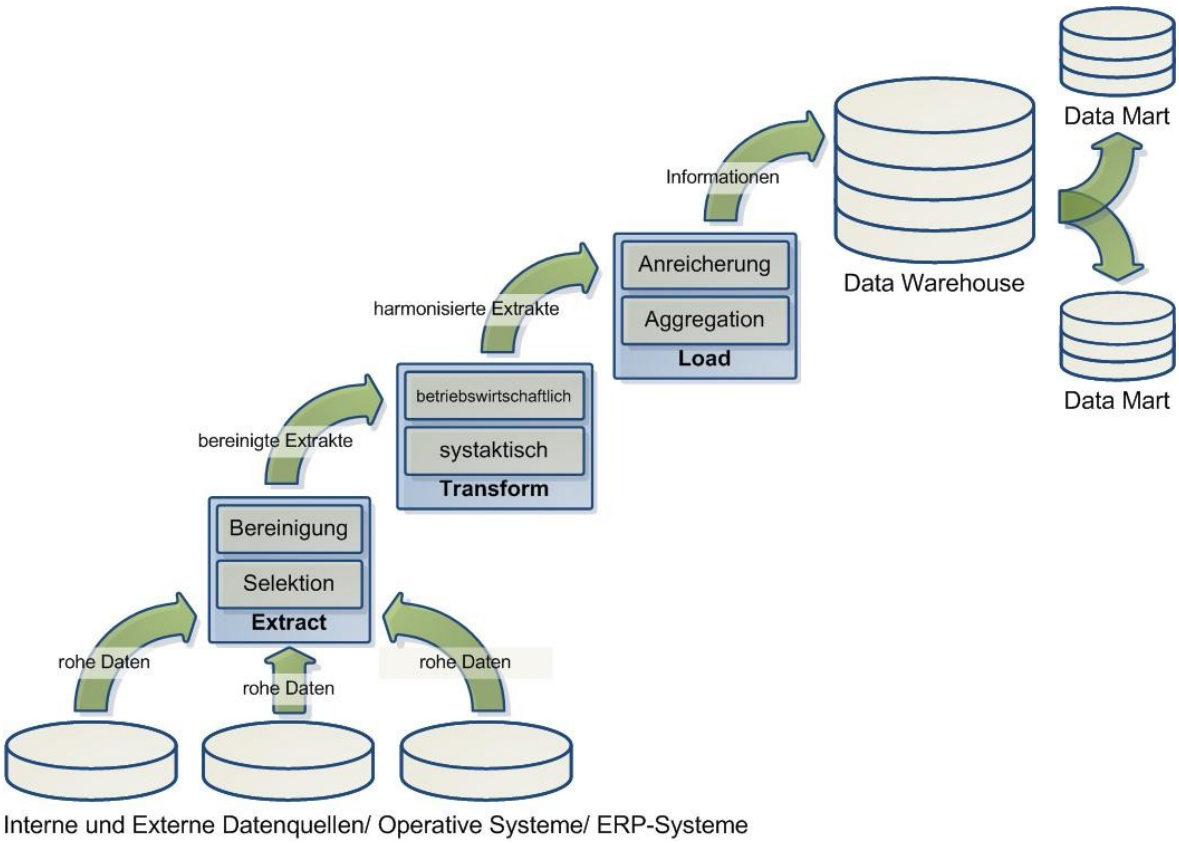

Abb. 6: ETL-Prozess

(Quelle: eigene Darstellung in Anlehnung an Bäumer, 2006, S. 11)

Extract

Der erste Schritt im Rahmen des ETL-Prozesses ist die Schaffung einer Zugriffsmöglichkeit auf die operativen Datenquellen. Diese Datenquellen können beispielsweise aus relationalen Datenbanken oder flachen ASCII-Dateien bestehen. Auch kommen die Daten nicht selten aus unternehmensexternen Quellen (Kemper et al., 2006, S. 24). Die Quellsysteme setzten i.d.R. auf unterschiedlichen Hardwareplattformen auf und werden von verschiedenen Betriebssystemen unterstützt. So existieren bei nicht wenigen Unternehmen Altsysteme mit veralteter Datenhaltung, die trotzdem eine relevante Informationsquelle für ein BI-System sind, welches ja historische Daten zur Entscheidungsunterstützung be-

nötigt (Mucksch et al., 2000, S.34f). Eine grundlegende Entscheidung bei der Extraktion ist, welche Datenquellen und welche Ausschnitte aus diesen in ein Data Warehouse-System integriert werden sollen. Diese Auswahl hängt stark davon ab, wie relevant die Inhalte für die geplanten Auswertungen sind. Bei der Selektion muss außerdem die Qualität der Quelldaten beachtet werden (Bauer et al., 2004, S. 81).

Die Zeitpunkte an denen eine der Daten aus den operativen Systemen extrahiert werden, können je nach Bedarf gewählt werden (Bauer et al., 2004, S. 82):

- Periodisch: Die Extraktion wird periodisch in Abhängigkeit von den Anforderungen an die Aktualität der Daten durchgeführt, z.B. im Nachtjob.
- Anfragegesteuert: Die Extraktion wird in diesem Fall durch eine explizite Anfrage angestoßen. Beispiel: Wird im Quell-ERP-System ein neuer Kunde hinzugefügt, kann das Extraktionstool angewiesen werden, die Daten in das Data Warehouse zu übertragen.
- Ereignisgesteuert: Es ist oft sinnvoll, den Vorgang durch ein bestimmtes Ereignis auszulösen. Als Beispiel kann eine im Vorfeld festgelegte Anzahl von Änderungen aufgeführt werden. Streng genommen sind periodische und anfragegesteuerte Extraktionen auch Ereignisgesteuert, da sie von einem bestimmten Zeitpunkt oder Nutzer angestoßen werden.
- Sofort (Real-Time): Bei hohen Anforderungen an die Aktualität der Daten z.B. bei Börsenkursen, ist es manchmal erforderlich, Änderungen der Quelldaten sofort in das Data Warehouse zu übernehmen.

Weiterhin erfolgt in diesem Prozessschritt eine erste Bereinigung der Extrakte. Die zu übernehmenden Daten müssen auf mögliche Defekte hin untersucht und entsprechend korrigiert werden. Hierbei kann man zwei unterschiedliche Bereinigungsklassen feststellen. Bei der Defekterkennung mit automatischer Korrektur während der Extraktionsphase können nur solche Defekte berichtigt werden, die von Korrekturregeln erkannt werden können. Beispielsweise wird bei einer Stornobuchung ein Sonderzeichen im ERP-System gesetzt, welches im Data Warehouse nicht interpretiert werden kann. Solche Fehler können in der Regel relativ problemlos, einfach und nutzertransparent mit Umsetzungstabellen (Mapping-Tables) bereinigt werden.

Bei der Defekterkennung mit manueller Korrektur nach dem Extraktionsvorgang können die Fehler bereinigt werden, die bei der automatischen Korrektur nicht beseitigt werden

können. Diese Fehler werden in einem technischen Logbuch dokumentiert. Nach dem Extraktionsprozess können Spezialisten diese Fehler anhand des Logbuchs dann manuell korrigieren (z.B. können Wertober- oder Wertuntergrenzen überschritten worden sein) Hierbei ist es durchaus denkbar, unrealistische Werte, z.B. durch Verschiebungen der Dezimalstellen zu entdecken und zu korrigieren. Auch können bei manuellen Korrekturen, Regeln für automatische Korrekturen gefunden werden (Kemper et al., 1998, S. 68f).

Transform

Transformation im ETL-Prozess heißt, sowohl Daten und Schemata, als auch die Datenqualität an Anwenderanforderungen anzupassen (Bauer et al., 2004, S. 83). Dieser Prozess wird auch Harmonisierung genannt. Hierbei geht es vor allem darum, aus Daten Informationen zu schaffen. Dazu müssen die bereinigten Extrakte aus den verschiedenen Quellen syntaktisch und betriebswirtschaftlich angepasst werden (Kemper et al., 2006, S. 27f).
Bei der syntaktischen Harmonisierung geht es vor allem um die Korrektur folgender Phänomene:

- Schlüsseldisharmonien: Bei der Übertragung von Daten aus mehreren Datenquellen in Data Warehouse können Schlüssel von Quelldatensätzen oft nicht übernommen werden, da sie nicht mehr global eindeutig sind. Im Rahmen der Transformation werden die Schlüssel aus den Quelldaten auf künstlich erzeugte, dafür aber einheitliche Schlüssel abgebildet (Bauer et al., 2004, S. 84). Als Beispiel soll ein ETL-Prozess dienen, bei dem die rohen Daten aus einem Call-Center, einem Außendienstsystem und einem Abrechnungssystem kommen. In jedem System gibt es unterschiedliche Primärschlüssel für Kunden. Um die Schlüsseldisharmonien zu eliminieren, wird den Kunden ein neuer, künstlicher Schlüssel in der Zieltabelle generiert und die Schlüssel aus den operativen Systemen werden als Fremdschlüssel mitgeführt (Kemper et al., 2006, S. 28).
- Unterschiedlich kodierte Daten: Um die Korrektur dieses Phänomens sicherzustellen, werden die Kodierungen aus den unterschiedlichen Quellen in eine gemeinsame Kodierung transformiert. Beispielsweise wird im Quellsystem A das Geschlecht der Kunden in der Datenbank mit „0“ oder „1“ festgehalten. Im Quellsystem B werden jedoch die Zeichen „M“ oder „W“ hinterlegt (Kemper et al., 1998, S. 70). Mittels Zuordnungs - bzw. Mapping-Tabellen werden die Zieldaten verein-

heitlicht (Kemper et al., 2006, S. 29). Ähnliches muss mit weiteren Daten wie z.B. dem Datum passieren.

- Synonyme: Hierbei handelt es sich um Attribute, die zwar unterschiedliche Namen im Quellsystem besitzen, jedoch dieselbe Bedeutung und dieselbe Kodierung haben. Informationen über Mitarbeiter können beispielsweise als „Personal" oder als „Mitarbeiter" geführt werden. Auch hier ist es verhältnismäßig einfach die Daten mit Mapping-Tabellen zu homogenisieren (Kemper et al., 2006, S. 29).
- Homonyme: Homonyme weisen zwar denselben Attributnamen auf, haben aber unterschiedliche Bedeutungen. Z.B. kann es in zwei Quelldaten die Bezeichnung Geschäftspartner geben. Bei der einen Quelle ist damit der Lieferant und bei der anderen der Kunde gemeint. Dies wird ähnlich wie die anderen Phänomene korrigiert (Kemper et al., 2006, S. 30).

Um die betriebswirtschaftliche Harmonisierung zu gewährleisten müssen folgende Fehler beseitigt werden:

- Abgleichung betriebswirtschaftlicher Kennzeichen: Da Unterschiede verschiedener Quellsysteme leider nicht nur technischer Natur sind, muss auch der Abgleich betriebswirtschaftlicher Kennzeichen beachtet werden. Vor allem muss gewährleistet werden, dass das gesamte Unternehmen, auf konsistente Daten zugreifen kann. Dazu müssen z.B. die Währungen oder die Periodenzuordnung (Quellsystem A benutzt die Quartale Q1 bis Q4 während Quellsystem B die Halbjahre H1 und H2 verwendet) in einheitliche Werte überführt werden (Kemper et al., 2006, S. 29).
- Granularität (Rasterung): Um die operativen Daten in die gewünschte Granularität zu überführen, sind weitere Transformationsregeln nötig (Kemper et al., 2006, S. 31). Damit wird die Detailliertheit der Zieldaten bestimmt. Dies ist eines der wichtigsten Korrekturen, da hier das Volumen der Zielsysteme mitbestimmt wird (Inmon, 1996, S.46). Zum einen möchte man detaillierte Daten bekommen, um später möglichst genaue Abfragen machen zu können, zum anderen möchte man das Volumen der Zielsysteme nicht zu groß werden lassen um nicht zu viele Performanceeinbußen hinnehmen zu müssen. Hierbei können z.B. Tagesumsätze zu Wochenumsätzen kumuliert werden, damit nicht zu große Zieldaten entstehen. Die

Detailliertheit der Rasterung bzw. der Granularität muss durch die bereits erwähnten Transformationsregeln bestimmt werden (Bauer et al., 2004, S.89).

Load

Die Ladephase umfasst die Übertragung der aufbereiteten Daten in das Data Warehouse. Hier muss zwischen dem ersten Laden (Initial-Load) und dem späteren Aktualisieren des Data Warehouses unterschieden werden (Bauer et al., 2004, S. 95).

Bevor die Daten jedoch endgültig im Data Warehouse abgelegt werden, müssen sie vorher noch verdichtet werden. Zu diesem Zweck werden in der Regel einige Dimensionshierarchietabellen angelegt, um später die benötigten Auswertungen zu ermöglichen. „Gesamt", „Jahr", „Monat", „Woche" und „Tag" kann als Beispiel einer einfachen Hierarchie gelten, wobei „Tag" hier die kleinste Einheit, also die kleinste Granularität darstellt. Die hier angelegten Dimensionshierarchien, können im Laufe der Zeit jedoch verändert werden. Meist ist dies auf sich ändernde Rahmenbedingungen im Unternehmen zurückzuführen (Kemper et al., 2006, S. 31). Dieses Phänomen muss jedoch akribisch historisiert werden, damit auch später noch die alten Zustände analysiert werden können. Genaueres wird dazu in Kapitel erörtert.

Schließlich werden die die Daten, nachdem sie den gesamten ETL-Prozess durchlaufen sind, in das Zielsystem, dem Data Warehouse, übertragen. Unter mithilfe dieser Phase werden betriebswirtschaftliche Kennzahlen berechnet. Diese Daten können Werte auf Basis von harmonisierten Daten in der gewünschten Granularität (Detailliertheit) oder auch auf Basis von bereits aggregierten Zusammenfassungstabellen (z.B. Tagesumsätze werden zu Wochenumsätzen aufsummiert) berechnet werden und werden anschließend als Attribute gespeichert. Anschließend wird das Data Warehouse mit den Werten (Attributen/ Daten) angereichert.

Da die Daten jetzt vollständig aufbereitet sind und im Data Warehouse „lagern", ist es nun vonnöten die Spezifikationen des DWH genauer zu betrachten. Dies soll im nächsten Abschnitt geschehen.

2.2.2 Data Warehouse

In diesem Abschnitt soll zuerst ein Data Warehouse motiviert und definiert werden. Im nachfolgenden Teil werden Anforderungen an ein DWH aufgeführt, die vor allem Inmon gefordert hat. Danach soll der Aufbau mit den jeweiligen Topologien beschrieben werden. Wie schon im Abschnitt 2.2.1 erwähnt, ist ein Data Warehouse nötig, da managementunterstützende Systeme nicht auf operativen Systemen aufbauen sollten um unter anderem Performance Vorteile zu gewinnen. Diese Aufteilung der Daten ist vor allem historisch entstanden. Zu Beginn der Datenbankdiskussion war die Auffassung sehr verbreitet, dass ein allumfassendes Datenbanksystem im Zentrum aller Anwendungssysteme eines Unternehmens stehen könnte. Die Probleme der konventionellen Dateiverarbeitung, wie z.B. Redundanz sollten schnell gelöst werden. Jedoch stellte sich schnell heraus, dass sich die Anforderungen an Datenbanksysteme für operative Systeme sehr stark von den Anforderungen an managementunterstützende Systeme unterscheiden. Die Unterschiede liegen vor allem in der Benutzergruppe, den Verarbeitungsmöglichkeiten und der zu verarbeitenden Daten (Groffmann, 1997, S. 8). Eine genauere Differenzierung zwischen operativen und analyseorientierten Systemen wurde bereits in Kapitel 2.1.1 dargelegt. Aus diesen Gründen ist eine Trennung von operativen Datenbanken und dem Data Warehouse sinnvoll und nötig.

Definition

Um ein Data Warehouse zu definieren muss vorher klargestellt werden, dass auch bei dieser Begrifflichkeit unterschiedliche Auslegungsmöglichkeiten existieren. So ist das Data Warehouse einerseits eine physische Datenbank, bei der das analyseorientierte Schema im Vordergrund steht (Bauer et al., 2004, S. 526). Auch Mertens definiert ein Data Warehouse als „zentralen Informationsspeicher." Weiter liegt die Herausforderung eines DWH laut Mertens darin, "mithilfe moderner informatischer und wirtschaftsinformatischer Methoden qualitative, quantitative, interne und externe Informationen zu verbinden" (Mertens, 2002, S.5). Etwas präziser wird das DWH von Muksch und Behme abgegrenzt: „Mit dem Begriff Data Warehouse i.e.S. wird generell eine von den operativen DV-Systemen isolierte Datenbank beschrieben, die als unternehmensweite Datenbasis für alle Ausprägungen managementunterstützender Systeme dient und durch ein strikte Trennung von operationalen und entscheidungsunterstützenden Daten und Systemen gekennzeichnet ist (Mucksch et al., 2000, S.6)."

Anderseits gibt es die weit verbreiteten Begriffe „Data Warehousing“ und „Data Warhouse-Prozess“. Diese Termini werden weitgehend synonym verwendet(Bauer et al., 2004, S. 526). Unter ihnen wird das Benutzen bzw. Verwenden eines DWH verstanden. Weiterhin gibt es den Fachausdruck „Data Warehouse-System“. Jedoch versteht man darunter meist mehr als nur eine physische, zentrale Datenspeicherung mit Analyseorientierung. Dazu gehören auch die vor- und nachgelagerten Schichten des Data Warehouses im BI-Prozess, wie Datenbeschaffung, Aufbereitung und Analyse (Bauer et al., 2004, S. 526).
Der Umgang mit ähnlich lautenden Begriffen die eine verschiedene Semantik haben, ist auch hier nicht unproblematisch. So kommt es in der einschlägigen Fachliteratur immer wieder zum unterschiedlichen Gebrauch der Begrifflichkeiten. Wie bereits in Abschnitt 2.1.2 veranschaulicht, wird in manchen Definitionen oder Verständnissen sogar der Begriff Business Intelligence auf ein Data Warehouse minimiert. Dabei ist das DWH lediglich ein Teil des BI-Prozesses. In der vorliegenden Arbeit soll deshalb, wenn von Data Warehouse die Rede ist, stets eine physische Datenbank gemeint sein, wie sie Muksch und Behme definieren.

Anforderungen

Das Data Warehouse-System speichert sowohl Bewegungsdaten, welche aus den operativen Systemen extrahiert wurden, als auch die aggregierten Werte und die im Rahmen der Anreicherung berechneten betriebswirtschaftlichen Kennzahlen (Bäumer, 2006, S. 13). Genauere Definitionen der Anforderungen und Spezifikationen eines DWH existieren von Inmon, der diesen Begriff nachhaltig geprägt hat: „A data warehouse is a subject oriented, integrated, nonvolatile and time variant collection of data in support of management's decisions (Inmon, 1996, S. 33).“

- subject oriented → Themen- bzw. Subjektorientierung: Klassische, operationale Systeme sind gewöhnlich an den Anwendungen und damit an den Wertschöpfungsketten der Unternehmen orientiert (Inmon, 1996, S. 33). Im Gegensatz dazu, ist ein DWH an den Informationsbedarfen des Entscheiders ausgerichtet. Er soll in die Lage versetzt werden, Informationen über die ihn interessierenden Themen (subjects) zu recherchieren. Solche Themen können Produktstrukturen (Produktgruppe, Produkt), Regionalstrukturen (Land, Region, Filiale) und Zeitstrukturen (Jahre, Monate) sein. Diesen werden dann Informationen (z.B. Gewinne oder Um-

sätze) und deren Ausprägung (z.B. Plan, Soll, Ist) zugeordnet (Kemper et al., 2006, S. 17f).

- integrated → Vereinheitlichung, Integration: Diese Eigenschaft und Anforderung an das Data Warehouse ist laut Inmon die wichtigste von allen hier aufgeführten Attributen (Inmon, 1996, S. 33). Die wesentliche Aufgabe bei der Erstellung eines DWH ist die Integration der unterschiedlichen Daten aus operativen und externen Datenquellen. Diese Aufgabe ist in der Realität meist sehr komplex, da die historische gewachsenen operativen Systeme häufig Datenredundanzen, Inkonsistenzen und semantische Widersprüche aufweisen (Kemper et al., 2006, S. 18). Diese Integration des DWH wird schon in der vorgelagerten Phase im BI-Prozess, dem ETL Prozess, durchgeführt. Dieser Vorgang und auch die zu vereinheitlichen Phänomene wurden bereits im Abschnitt 2.2.1 unter 'Transform“ diskutiert.

- nonvolatile → Nicht-Volatilität, Dauerhaftigkeit, Stabilität: Mit dem Begriff der Volatilität wird der Grad beschrieben mit dem sich Daten im Laufe der normalen Nutzung ändern. Bestimmt wird dieser Grad mit der durchschnittlichen Anzahl der Änderungen oder mit der absoluten Anzahl der Änderungen in bestimmten Zeiträumen (Mucksch et al., 2000, S. 13).

  Auf operationale Daten wird regelmäßig zugegriffen und genauso regelmäßig werden diese Daten auch verändert und aktualisiert (Inmon, 1996, S. 36). Somit repräsentieren sie den jeweils aktuellen Zustand des Unternehmens und den aktuellen Zustand innerhalb eines Geschäftsprozesses. Die Historie, also der Verlauf der Zustände der Geschäftsprozesse, wird jedoch nicht gespeichert. Lediglich aus Gründen der Datensicherung (z.B. für das Wiederaufsetzen der operativen Systeme nach technischen Defekten) erfolgt meist eine Speicherung der Daten über einen begrenzten Zeitraum (Kemper et al., 2006, S. 19). Im Gegensatz dazu, werden die im DWH gespeicherten Daten nach der fehlerfreien Übernahme und gegebenenfalls nötigen Korrektur und Transformation i.d.R. nur in Ausnahmefällen aktualisiert und verändert. Diese Korrekturen sind nur dann zulässig, wenn beispielsweise im Rahmen des ETL-Prozesses Fehler aufgetreten sind, oder wenn im operativen System fehlerhafte Daten eingegeben wurden und dort erst nach der Durchführung des ETL-Prozesses korrigiert wurden (Mucksch et al., 2000, S. 13).

Weiterhin weisen die im DWH abgelegten, integrierten Daten die Besonderheit auf, mit ihrer Dauerhaftigkeit für künftige betriebswirtschaftliche Analysen zur Verfügung zu stehen. Um das Datenwachstum im DWH und damit im BI-System zu begrenzen, müssen sinnvollen Historisierungskonzepte entwickelt und implementiert werden. So sind z.B. Überlegungen nötig, ob Daten, die für aktuelle Analysen schon zu alt sind, nicht in verdichteter Form und somit komprimiert abzulegen sind – beispielsweise ab welchem Alter Datenbestände zu archivieren sind (Kemper et al., 2006, S. 19). Jedoch kann laut Muksch und Behme, der Forderung Inmons nach Nicht-Volatilität eines Data Warehouses nur bedingt zugestimmt werden, da es auch bspw. möglich sein muss Plandaten im DWH unterzubringen(Mucksch et al., 2000, S. 78). Das diese nach Ablauf des Planungszeitraums verändert bzw. gelöscht werden müssen ist in diesem Zusammenhang selbstredend.

- time variant → Zeitorientierung der Information: Der Zeithorizont eines Data Warehouses ist deutlich und entscheidend länger als bei operativen Systemen. Während ein Zeitraum von sechzig bis neunzig Tagen für ein operatives System normal ist, sind beim Data Warehouse Zeiträume von fünf bis zehn Jahren üblich. Wie schon unter dem Abschnitt 'nonvolatile' erwähnt, bildet ein operatives System lediglich einen aktuellen Schnappschuss des Unternehmens ab. Als solcher kann und wird dieser Schnappschuss aktualisiert und verändert. Daten in einem DWH sind im Grunde nicht mehr als eine komplexe Aneinanderreihung solcher Schnappschüsse, die im jeweiligen Zeitpunkt vom aktuellen Stand des operativen Systems 'geschossen' wurden. Dazu ist es erforderlich, dass die Schlüsselstrukturen des Data Warehouses immer Zeitelemente beinhalten. Solche Zeitelemente können das Jahr, der Monat oder der zu dem Schnappschuss zugehörige Tag sein. Dies ist bei operativen Systemen nicht zwingend erforderlich. Sie können, müssen aber keine Zeitelemente in der Schlüsselstruktur haben (Inmon, 1996, S. 36).

Struktur des Data Warehouses

Viele Unternehmen bauen und verwalten ein einziges und zentrales Data Warehouse. Hierbei handelt es sich um eine physische Datenbank, die zusätzlich zu den operativen Datenbeständen existiert (Schnizer et al., 1998, S. 42) und deren Quelle eben diese operativen Datenbestände sind. Inmon motiviert einige Gründe, warum ein einziges, unterneh-

mensweites, zentrales DWH in dem die kompletten aufbereiteten Daten gelagert werden, Sinn macht. Als ersten Vorteil nennt er, dass das DWH im Hauptsitz des Unternehmens installiert wird wo i.d.R. die Verwaltung stattfindet und das Management ihren Sitz hat. Im Hauptsitz wird das DWH als Grundlage der Analysesysteme am meisten benötigt(Inmon, 1996, S. 197). So ist ein zentrales DWH gerade für solche Unternehmen empfehlenswert, deren operationales System, auch zentralisiert ist (Jarke et al., 1999, S. 11). Weiterhin betont Inmon, dass das Volumen der Daten eines DWH so groß ist, dass eine zentrale Speicherung Sinn macht. Sogar wenn die Daten in verschiedenen Lokalitäten gelagert wären und in ein DWH integriert werden könnten, ist der Zugriff beschwerlicher als der eines zentralen DWH. Zusammenfassend kann man sagen, dass laut Inmon die politischen, ökonomischen und die technologischen Motive ein einziges, zentrales DWH favorisieren (Inmon, 1996, S. 197). Diese Vorteile können jedoch teilweise hinfällig werden, wenn die physische Datenbank so groß wird, dass Performance-Nachteile entstehen. Auch zunehmende Benutzerzahlen würden diesen Effekt verstärken. Außerdem ist die Implementierung einer rein zentralen Lösung mit erheblichen konzeptionellen und technischen Problemen behaftet. Viele Unternehmen bevorzugen daher eine Lösung, die eine Verteilung der Verarbeitungs- und Administrationslast präferiert (Kemper et al., 2006, S. 20).

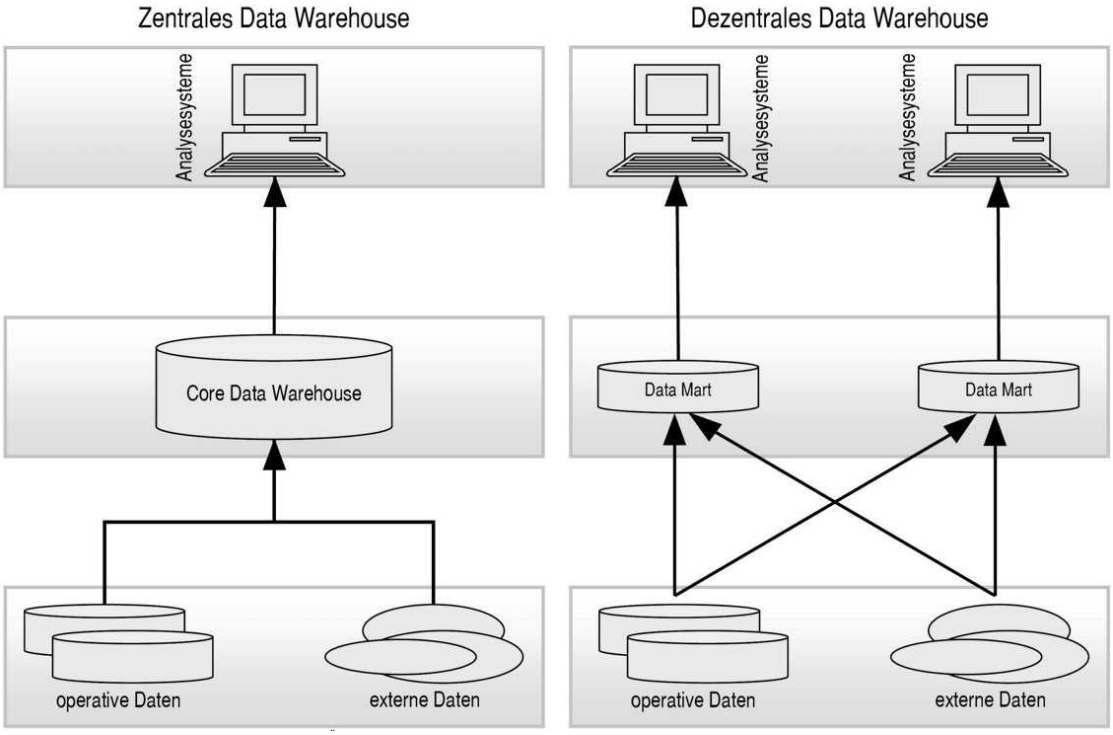

Abb. 7: Architekturvarianten von DWH

(Quelle: Kemper et al., 2006, S. 20)

Eine dezentrales DWH besteht aus isolierten Data Marts. Dabei werden die im ETL-Prozess transformierten Daten, nicht in das Core-DWH (Kern-Data Warehouse) sondern in einzelne, isolierte Data Marts geladen, wie Abb. 7 zeigt. Sie besitzen häufig eine Datenhaltung mit hoher Zweckorientierung für einzelne Fachabteilungen (z.B. Controlling oder Marketing) oder für Querschnittsthemen (z.B. Rechnungswesen). Jedoch ist an dieser Stelle darauf aufmerksam zu machen, dass eine Anhäufung von isolierten Data Marts Nachteile mit sich bringen. So fehlt dabei eine integrierte Sichtweise auf das Gesamtunternehmen, was unternehmensweite Analysen erschwert. Außerdem ist die evtl. nötige Erweiterung zu einem unternehmensweiten Data Warehouse kaum noch möglich (Kemper et al., 2006, S. 20f).

Data Marts

Der Begriff Data Mart soll an dieser Stelle genauer definiert werden. Data Marts sind kleinere Datenpools für eine Klasse von Applikationen, die üblicherweise für einen eingeschränkten Benutzerkreis aufgebaut werden (Kemper et al., 2006, S. 22). Es kann zwischen abhängigen und unabhängigen Data Marts unterschieden werden. Während abhängige Data Marts auf Daten aus dem Data Warehouse beruhen, werden unabhängige Data Marts direkt aus den Datenquellen, welche dem DWH zugrunde liegen, gefüllt (Kamp, 2006, S. 16).

Abhängige Data Marts können damit motiviert werden, dass sie das zentrale DWH entlasten. User müssen nicht mehr auf das zentrale DWH zugreifen, sonder können ihre Anfragen einfach an den Data Mart stellen, auf dem die für sie relevanten Daten gelagert

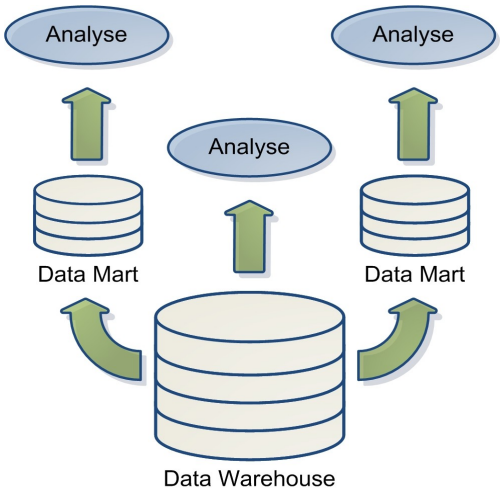

Abb. 8: abhängige Data Marts

(Quelle: eigene Darstellung nach Bauer et al., 2004, S. 61)

sind. Dadurch muss das Core-DWH weniger Anfragen bedienen und die Auslastung wird geringer. Wichtig ist, dass bei dieser Variante nur Extrakte des Data Warehouse enthalten sind, die allerdings auch aggregiert sein können (vgl. Abb. 8). Jedoch finden bei der Befüllung des Data Marts keine Datenbereinigungen und Normierungen mehr statt, sodass die möglichen Analyseergebnisse immer inhaltlich und strukturell mit den Analyseergebnisse, deren Grundlage das zentrale Data Warehouse ist, übereinstimmend sind (Bauer et al., 2004, S. 61).

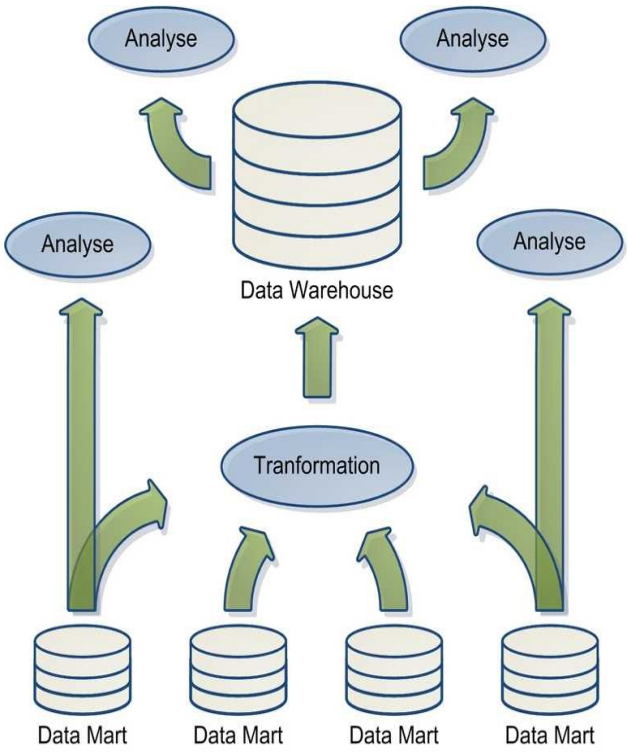

Abb. 9: unabhänige Data Marts

(Quelle: eigene Darstellung nach Bauer et al., 2004, S. 63

Unabhängige Data Marts entstehen oft, wenn in einzelnen Organisationsbereichen (z.B. Abteilungen) unabhängig von einander „kleine" Data Warehouses aufgebaut werden. Solche Vorhaben sind i.d.R. von der Komplexität überschaubar und erzielen meist schon nach kurzen Projektlaufzeiten brauchbare Ergebnisse für die Abteilung. Sollte jedoch die Notwendigkeit bestehen unabhängige Data Marts zu einem zentralen DWH zu integrieren um unternehmensweite Analysen zu ermöglichen, müssen die Daten innerhalb des ETL-Prozesses aufbereitet werden. Unabhängige Data Marts stellen dabei bereits vorverdichtete und zum Teil bereinigte Datenquellen zur Verfügung (vgl. Abb. 9). Der Nachteil der entstehenden Gesamtarchitektur ist jedoch, dass Analyseergebnisse nicht notwendi-

gerweise mit der den Ergebnissen einer Analyse auf Basis eines Data Warehouses vergleichbar sind.(Bauer et al., 2004, S. 62f).

Mit der Definition und Beschreibung von Data Marts sind die Daten letztendlich an ihrem Ziel und können nun, analysiert, visualisiert und ausgegeben werden. Da der BI-Prozess jedoch bisher nicht wirklichen Nutzen bringt, muss die zentralste Anwendung des DWH, das Online Analytic Processing eingebunden werden. Dazu ist es nötig, den Aufbau von Data Warehouses zu verstehen. Die Beschreibung des Aufbaus soll nun folgen.

Datenbankentwurf

Als Datenbankschema für Data Warehouse-Anwendungen hat sich das sogenannte Stern-Schema (engl. star schema) durchgesetzt[4] (vgl. Abb. 10). Dieses Schema besteht aus einer Faktentabelle (im Beispiel: Verkäufe) und mehreren Dimensionstabellen (im Beispiel: Zeit, Produkt, Filiale und Kunde), die über Fremdschlüssel mit der Faktentabelle verbunden sind (Kemper et al., 2004, S. 490).

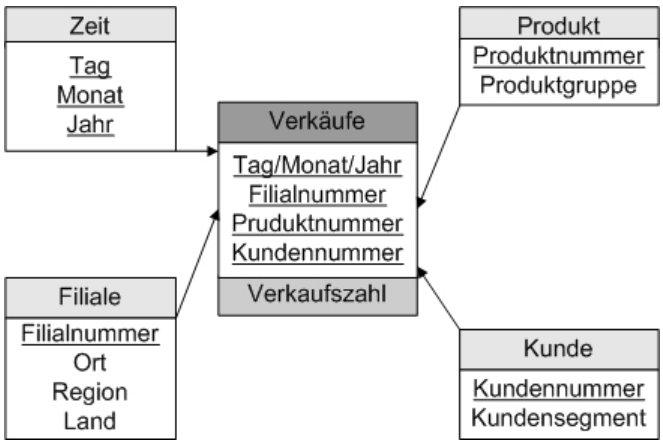

Abb. 10: Beispiel eines Star-Schemas

(Quelle: Bäumer, 2006,S. 17)

Fakten (auch Measures genannt) sind numerische Werte, die den Mittelpunkt der Datenanalyse bilden. Es sind die im ETL-Prozess (Kap. 2.2.1) bereits beschrieben betriebswirtschaftlichen Kennzahlen. Sie repräsentieren in der Regel monetäre Werte bzw. Mengen wie Umsatzerlöse, Umsatzmengen, Einzelkosten oder den Personalbestand. Dimensionen dagegen haben beschreibende Aufgaben. Sie ermöglichen unterschiedliche Sichten auf die Fakten. Nach den Dimensionen können Faktdaten zur Auswertung gruppiert und analysiert werden. Dimensionswerte können Tage, Regionen oder Produkte sein (vgl. Abb. 10) (Kemper et al., 2006, S. 61f). Die Einordnung in Dimensionen ist aus den Forderungen von

[4] Des Weiteren gibt es Snowflake- und Galaxien-Schematas, auf die hier nicht näher eingegangen wird.

Inmon entstanden. Somit ist eine multidimensionale Datenspeicherung möglich. Die Analyse von Fakten macht jedoch erst dann Sinn, wenn sie multidimensional (unter mehreren Dimension) gespeichert werden. So kann zum Beispiel die Zahl 32 für die Umsatzmenge am 09.09.08 in der Filiale mit der Filialnummer 13 des Produktes „XYZ“ stehen. Die Zahl 32 stellt nur das Faktum dar und wird erst durch die Dimensionen eingeordnet. (Bäumer, 2006, S. 16).

Innerhalb von Dimensionen können (vertikale) hierarchische Beziehungen bestehen. Dies ermöglicht eine Betrachtung unter verschiedenen Verdichtungsstufen. So kann die Dimension Zeit hierarchisiert werden in „Tag“, „Monat“, „Quartal“ und „Jahr“. Hierbei sind auch parallele Hierarchien möglich, wenn zum Beispiel anhand der Modellpalette des Automobilherstellers BMW innerhalb der Dimension Modelle zwei verschiedene Verdichtungswege über die Ebenen „Motoren“ und „Modellgruppe“ erstellt werden (siehe Abb. 11) (Kemper et al., 2006, S. 62).

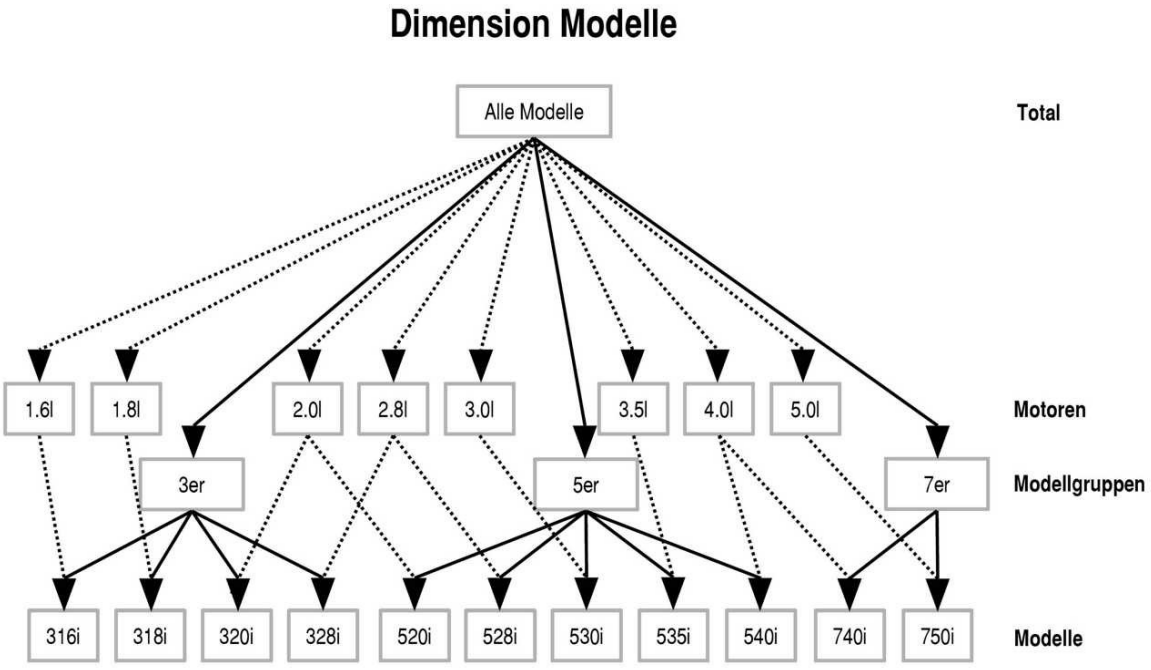

Abb. 11: Parallele Dimensionshierarchien

(Quelle: Kemper et al., 2006, S. 61 dort aus: Hahne, 1998, S. 103)

Die Aufteilung des DWH in Fakten, Dimensionen und Hierarchien werden besonders beim OLAP genutzt. Dazu wird ein OLAP-Cube extrahiert und verwendet. Dieser wird im nächsten Abschnitt genauer dargestellt. Außerdem wird durch OLAP ersichtlich wie die ersten Nutzen aus BI-Systemen gezogen werden können

2.2.3 Online Analytical Processing

In diesem Abschnitt wird das Online Analytical Processing behandelt. Dabei werden zunächst die Anforderungen an OLAP motiviert und erläutert. Anschließend werden verschiedene OLAP-Tools genannt und voneinander abgegrenzt. Schließlich folgt noch ein Beispiel eines OLAP-Tools mit welchem gezeigt wird, wie Spreadsheet-Viewers (Microsoft® Excel 2007®) einen OLAP-Cube visualisiert.

Im eigentlichen Sinne ist Online Analytic Processing bereits eine Anwendung von Business Intelligence Systemen. Jedoch ist BI ohne OLAP nur schwer vorstellbar, da OLAP die wichtigste und grundlegendste Visualisierung des Data Warehouses ist und somit eine weitere Phase im BI-Prozess darstellt.

Definition und Anforderungen

Die Forderung nach benutzerfreundlichen, flexiblen Abfragesystemen für Ad-hoc-Analysen beschäftigt die Wissenschaft und die Praxis schon seit vielen Jahren (Kemper et al., 2006, S. 93). Solche Informationssysteme müssen sich an dem Geschäftsverständnis bzw. an der Sichtweise auf das Unternehmen orientieren. Dazu sind multidimensionale Perspektiven geeignet, die den Mitarbeitern einen flexiblen und intuitiven Zugang zur benötigten Information geben. Unter Multidimensionaltität ist eine logische Anordnung von betriebswirtschaftlichen Variablen bzw. Kennzahlen (z.B. Umsatz, Deckungsbeitrag) zu verstehen, die entlang von unterschiedlichen Dimensionen (z.B. Kunden, Artikel, Niederlassungen, Regionen) aufgliedert werden (Gluchowski et al., 2008, S. 143f; Chamoni et al., 2000, S.334).

Ähnliche, mehrdimensionale Sichtweisen wurden schon in älteren Informationssystemen genutzt, jedoch wurden sie damals nicht konkretisiert. Erst mit das Konzept des Online Analytic Processing (OLAP) wurde die Multidimensionalität ein wichtiges Geschäftskriterium für entscheidungsunterstützende Informationssysteme (Gluchowski et al., 2008, S. 143).

OLAP steht für eine Gattung von Anfragen, die nicht nur einzeln auf einen Wert zugreifen, sondern einen dynamischen, flexiblen und interaktiven Zugriff auf eine Vielzahl von Einträgen erfordert (Bauer et al., 2004, S. 97). Oft wird OLAP fälschlicherweise mit dem Begriff des Data Warehousing (vgl. Abschnitt 2.2.2 → Definition) gleichgesetzt, weil die Datengrundlage für OLAP meist ein Data Warehouse ist (Bauer et al., 2004, S. 98).

So stellte E. F. Codd 1993 in Zusammenarbeit mit S. B. Codd zwölf Anforderungen an das OLAP-Konzept, die zwölf Codd'schen Regeln des Online Analytic Processing, vor und sorgte damit für eine neue Begrifflichkeit in der Diskussion um die Analyse von multidimensionalen Datenräumen (Höhn, 2000, S. 179; Kemper et al., 2006, S. 93f):

- mehrdimensionale konzeptionelle Perspektiven
- Transparenz
- Zugriffsmöglichkeit
- gleichbleibende Antwortzeiten beider Berichterstellung
- Client/-Server Architektur
- generische Dimensionalität
- dynamische Handhabung dünn besetzter Matrizen
- Mehrbenutzerunterstützung
- uneingeschränkte kreuzdimensionale Operationen
- intuitive Datenverarbeitung
- flexible Berichterstellung
- unbegrenzte Dimensions- und Klassifikations- bzw. Aggregationsebenen

Diese Regeln sind teilweise sehr umstritten und wurden schon zu der Zeit heftig kritisiert, da sie auf ein konkretes, kommerziell erwerbbares Datenbankmodell ausgerichtet sind. Jedoch löste dieser Artikel eine Diskussion über IT-basierte Managementunterstützung aus. Die folgende Diskussion brachte letztendlich etwa 300 neue Kriterien im OLAP-Umfeld hervor. 1995 reduzierten Pendse und Creeth die Anforderungen an OLAP auf fünf Kerninhalte die sich mit dem Akronym FASMI abkürzen lassen. Dabei steht FASMI für „Fast Analysis of Shared Multidimensional Information. Diese Kriterien haben sich als Umschreibung des Konzeptes durchgesetzt (Kemper et al., 2006, S. 94f). Der Vorteil dieser Beschreibung von OLAP ist, dass sie pragmatisch ist und im Gegensatz zu den Codd'schen Regeln nicht an eine bestimmte Technologie gebunden ist (Chamoni, 1998, S. 237).

- Fast (Geschwindigkeit): Die Geschwindigkeit des Systems wird hier konkretisiert, uns zwar sollen die meisten Anfragen in ca. fünf Sekunden beantworten werden, wobei die häufigsten Anfragen deutlich schneller geliefert werden sollen. Auch die komplexen Anforderungen sollten spätestens nach zwanzig Sekunden bearbeitet sein (Bauer et al., 2004, S. 102).

- Anaysis (Analyse): Das OLAP-Werkzeug soll anwenderfreundliche und intuitive Verfahren und Techniken zur Analyse der Daten bereitstellen. Dazu müssen dem Anwender Funktionen zur Verfügung gestellt werden, mit denen er beliebige Berechnungen und Strukturuntersuchungen durchführen kann, ohne Programmerkenntnisse zu haben. Außerdem soll der Anwender verschiedene Präsentationsformen nutzen können, die jedoch nicht zwangsläufig vom OLAP-Werkzeug bereitgestellt werden. Es kann sich auch um Einbindungen von externen Werkzeugen, wie z.B. Excel handeln (Bauer et al., 2004, S.102; Chamoni, 1998, S.237).
- Shared (geteilt Nutzung): Ein OLAP-Werkzeug benötigt ein sicheren Mehrnutzerbetrieb mit der Möglichkeit, Zugriffsrechte auf Zellenebene zu vergeben. Für den schreiben Zugriff sollten Sperrverfahren sowie Sicherungs- und Wiederherstellungsverfahren vorhanden sein (Bauer et al., 2004, S.102). Jedoch verzichten die meisten OLAP-Produkte bewusst auf ein update locking[5] , um keine Performance-Verluste zu erleiden (Chamoni, 1998, S.237).
- Multidimensional: Dem Anwender soll eine multidimensionale konzeptuelle Sicht auf die Daten zur Verfügung gestellt werden. Bei der Analyse muss er die Möglichkeit haben beliebige Dimensionen bei Bedarf zu kombinieren. Diese Forderung gilt unabhängig von der physisch zugrunde liegenden Datenbanktechnologie (Bauer et al., 2004, S.102).
- Information (Datenumfang): Der kritische Faktor bei der Beurteilung von OLAP-Werkzeugen ist nicht die Ressourcenbelegung (Arbeitsspeicher und externer Speicher) sondern die Datenmenge. Ein System ist dann von hohem Nutzen, wenn mehr Datenelemente bei stabil bleibenden Antwortzeiten analysiert werden können (Chamoni, 1998, S.237).

OLAP-Werkzeuge

Das Konzept von OLAP, das bereits in Abschnitt 2.2.3 beschrieben wurde, ist inzwischen bei allen Produkten umgesetzt worden. Jedoch liegt bei der Fokus nicht auf der exakten Umsetzung der Codd'schen Regeln oder der FASIM-Kriterien (Schinzer, 2000, S. 420). Vielmehr soll es mittels OLAP möglich sein, große und sehr große Datenmengen analytisch zu bearbeiten. Die Stärken von OLAP liegen im Berichtswesen. Die Hauptaufgabe, die sich grob mit „Datenanzeige“ beschreiben lässt, kann statisch als vorgefertigter Be-

[5] update locking → während dem Update wird der Zugriff auf die Quelldatenbank unterbunden

richt, dynamisch (ad-hoc) oder online gelöst werden. Die Berichte können auf diversen Medien (Bildschirm, Browser, Papier, etc.) in Form von Tabellen, oder Grafiken ausgegeben werden. In diesem Sinne sind OLAP-Werkzeuge Darstellungswerkzeuge, mit deren Hilfe Wissen erzeugt werden kann (Martin, 1998, S. 145). OLAP-Werkzeuge zeichnen sich durch die folgenden Merkmale aus (Martin, 1998, S. 145):

- Visualisierung
- schnelle, interaktive Abfrage-/ Antwortzeit
- Eignung zur Analyse von Daten mit zeitlicher Abhängigkeit
- Eignung zum Aufspüren von Ausreißern

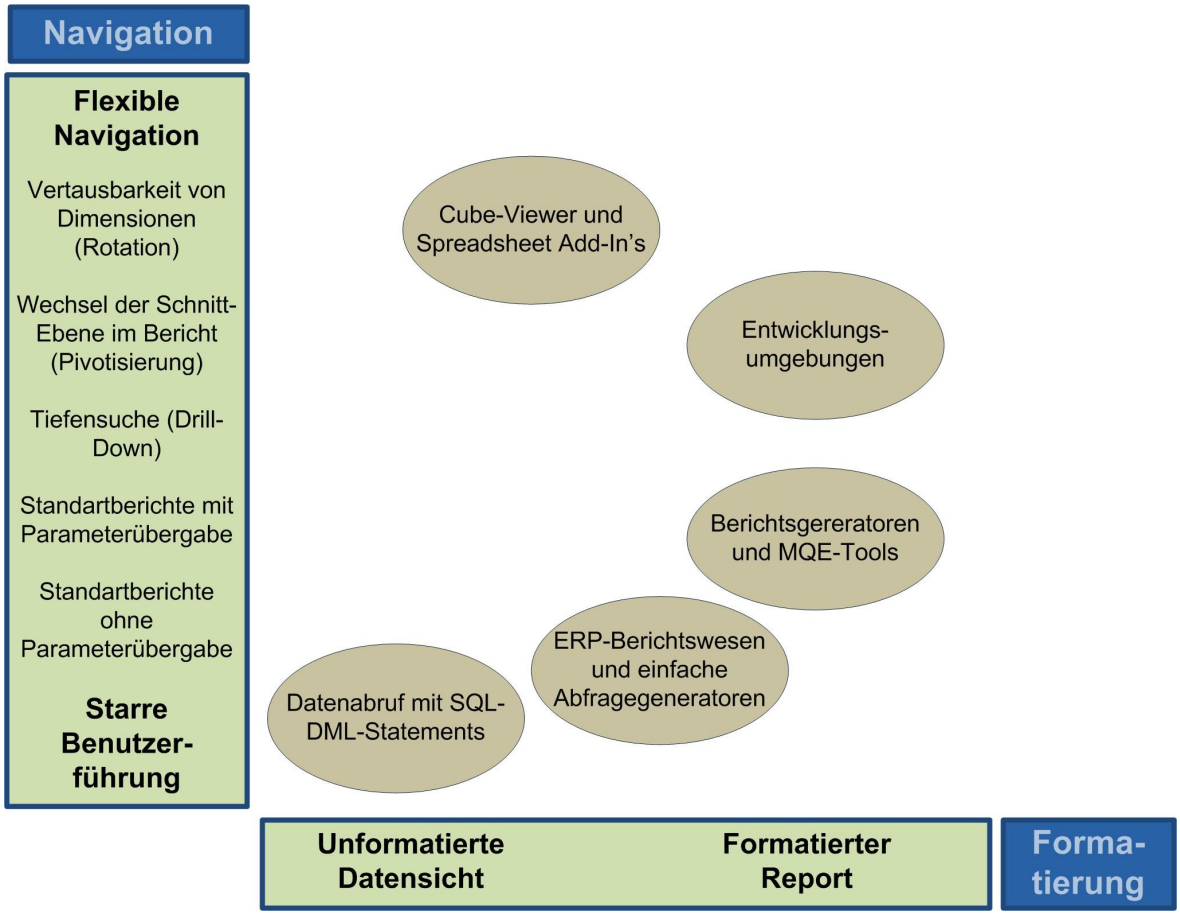

Abb. 12: Kategorisierung von Endbenutzerwerkzeugen

(Quelle: eigene Darstellung nach Gluchowski et al., 2008, S. 181)

Heutzutage adressieren OLAP-Front-End-Tools sehr verschieden Anwendergruppen mit den unterschiedlichsten Anforderungen und Vorkenntnissen. System-unerfahrene Mitarbeiter brauchen intuitiv und mit wenig Einarbeitung zu bedienende Benutzeroberflächen. Dagegen haben Mitarbeiter mit IT-Erfahrung häufig gehobenere Ansprüche an die Navigierbarkeit und die Analysefunktionalität der Tools.

In etwa können die Front-End-Tools wie in Abb. 12 dargestellt mit den Achsen Navigation und Formatierung eingeordnet und abgegrenzt werden. Dabei sind die Datenabrufe mit SQL-DML[6]-Statements und die Berichtswerkzeuge von ERP-Systemen nur eingefügt, um als Vergleichsmaßstab für die anderen OLAP-Tools zu dienen, auch wenn sie teilweise in Unternehmen immer noch zu Auswertungszwecken eingesetzt werden. Vor allem der Einsatz von SQL-Befehlen verschließt den Weg zur Auswertung und Analyse für viele Endanwender. Die Ausgabe folgt dann in unformatierten Datentabellen. Auch Berichtsgeneratoren und MQE[7]-Tools lassen im Hinblick auf die Navigierbarkeit und die Flexibilität Wünsche offen, bieten dagegen aber eine Vielzahl von leistungsstarken Funktionen bei der Umsetzung von Standardberichten.

Maximale Flexibilität bieten Cube-Viewer und Spreadsheet-Add-In's bei der interaktiven Navigation im multidimensionalen Datenmaterial. Diese Tools werden auch als Ad-hoc-Analysewerkzeuge bezeichnet. Cube-Viewer sind alle multidimensionalen Front-End-Tools, die ohne zusätzlichen Entwicklungsaufwand eine direkte, interaktive Nutzung der (multidimensionalen) Daten ermöglichen. Mit diesen Tools können Endanwender u.a. beliebige Schnitte durch die Datenwürfel ziehen, die Perspektive auf die Daten zur Laufzeit variieren, oder die Daten als Geschäftsgrafiken ausgeben (Gluchowski et al., 2008, S. 179f).

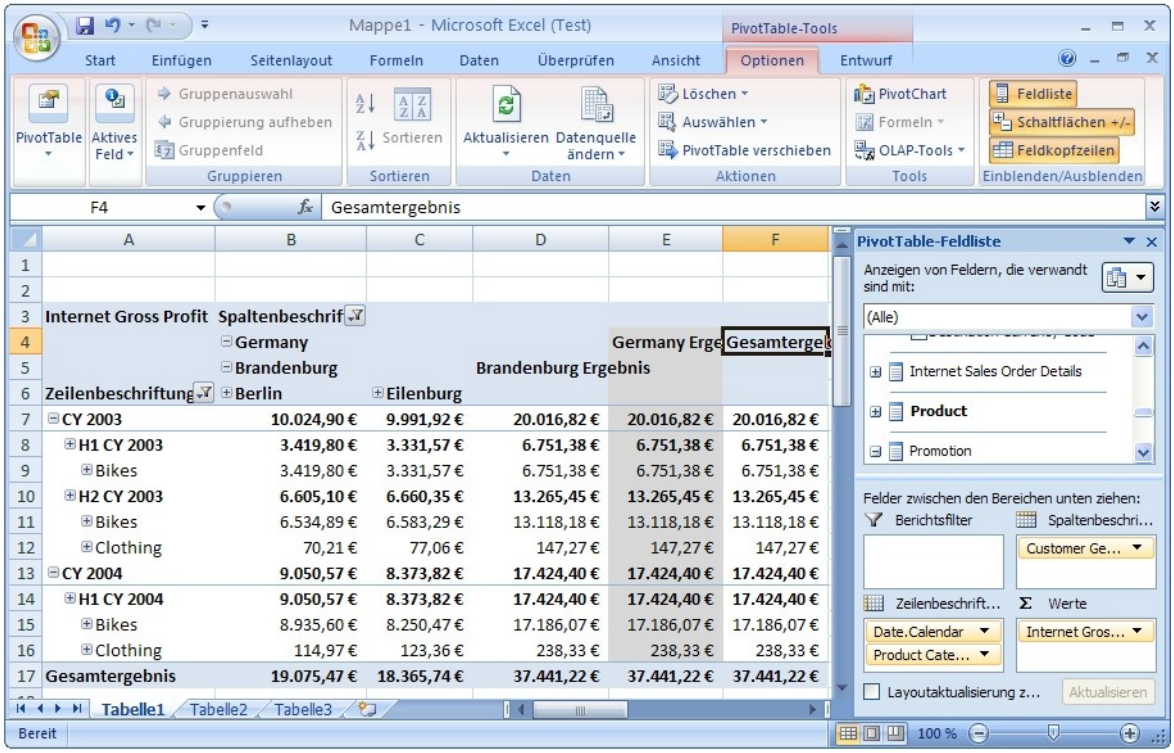

Abb. 13: Microsoft® Excel 2007® als OLAP-Cube-Viewer

(Quelle: eigener Screenshot)

[6] Structured Query Language - Data Manipulation Language

[7] Managed Query Environment

Ein weit verbreitetes Tool zur Analyse und Visualisierung und damit zur Entscheidungsunterstützung besonders im Controlling, sind Tabellenkalkulationsprogramme bzw. Spreadsheet-Viewer (vgl. Abb. 13), die sich fast flächendeckend etabliert haben. Mit ihrer Hilfe kann man sich die Dimensionen des OLAP-Cubes anzeigen lassen und per Drag&Drop auf das Analysefeld ziehen. Dann muss lediglich noch das Faktum ausgewählt werden (ebenfalls per Drag&Drop) (in Abb. 13: Bruttogewinn im Internet) und schon bekommt man auswertbare Ergebnisse. In dieser erzeugten Pivot-Tabelle[8] können die gewählten Dimensionen nun nach Wunsch verändert werden (z.B. die nächst kleinere Einheit: von Jahr 2003 in Halbjahr 1 und Halbjahr 2). Dies passiert alles online und ist damit sehr schnell. Es sind aber noch mehr Operationen mit dem OLAP-Würfel möglich. Dies wird später genauer beschrieben. Nun sollte jedoch zuerst analysiert werden, ein OLAP-Cube konkret aufgebaut wird.

OLAP-Hypercube

Multidimensionale Datenräume bestehen aus Fakten, Dimensionen und Hierarchisierungen (vgl. Kap. 2.2.2). Theoretisch ist die Anzahl von Dimensionen in einem Hypercube

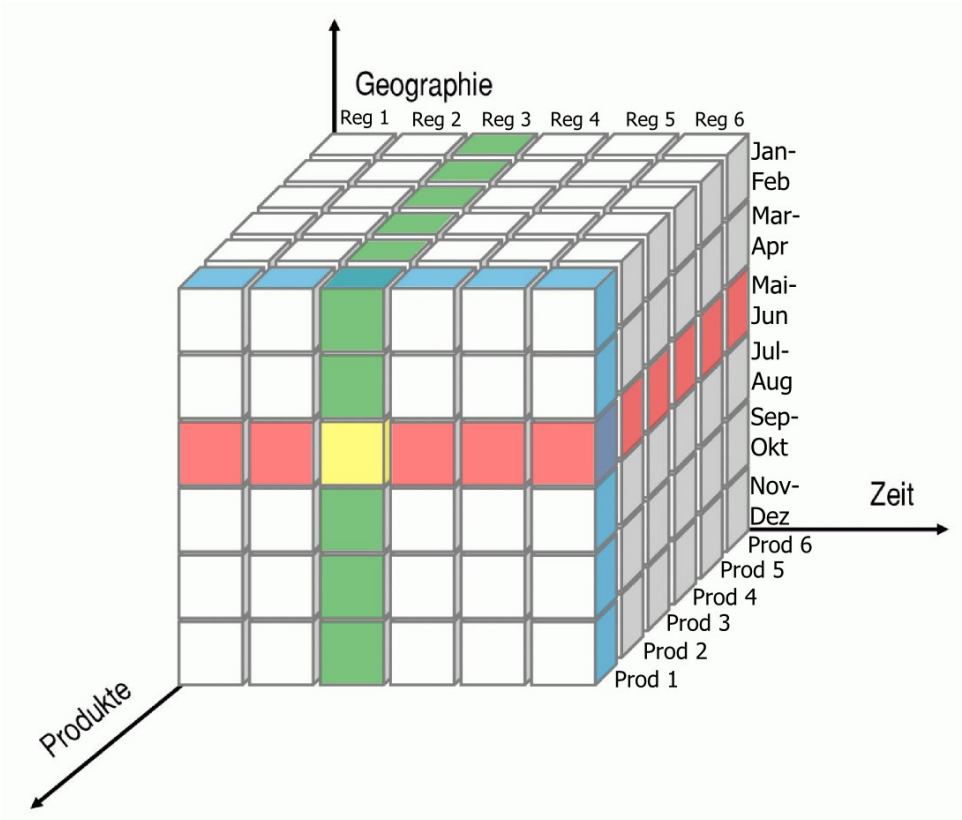

Abb. 14: OLAP-Hypercube mit drei Dimensionen

(Quelle: modifiziert nach: Kemper et al., 2006, S. 95)

[8] franz. (se) pivoter = (sich) drehen) bezieht sich auf die Operationsmöglichkeiten des OLAP-Cubes

nicht begrenzt doch in betriebswirtschaftlicher Anwendung haben die Cubes meist maximal Dimensionen im niedrigen zweistelligen Bereich. Diese Eingrenzung ist nicht technisch sondern vielmehr auf die Problemstellungen zurückzuführen, denn Auswertungen auf der Basis von mehreren Dimensionen auszuführen ist schlichtweg unrealistisch und undurchschaubar für den Analysten.

Unabhängig von der Anzahl der Dimensionen wird stets ein Würfel als Metapher für multidimensionale Datenräume verwendet. Die Begrifflichkeit „Hypercube" basiert auf dieser Vorstellung und deutet bereits die unbeschränkte Anzahl möglicher Dimensionen an (vgl. Abb. 14) (Kemper et al., 2006, S. 95).

An Abb. 14 soll nun beispielhaft gezeigt werden, wie man mittels OLAP Betriebswirtschaftliche Informationen über das zugrunde liegende Unternehmen bekommt. Die Fakten (jeder kleine Würfel des Hypercubes stellt ein Faktum dar) sind in diesem Beispiel Umsätze. Jetzt sollen die Umsätze nach den Dimensionen Geografie, Zeit und Produkte ausgewählt werden. Zuerst werden die Umsätze von Produkt 1 ausgewählt. Der OLAP-Client zeigt jetzt die gesamten Umsätze im Unternehmen für Produkt 1 an. In Abb. 14 sind dies alle Würfel die von den blauen Würfeln eingerahmt werden. Nun soll diese Umsätze auf den Zeitraum von Mai bis Juni begrenzt werden. Dies symbolisieren die von den roten Würfen eingerahmten Würfel. Die Schnittmenge der blauen und der roten „Scheibe" stehen für die Umsätze von Produkt 1 von Mai bis Juni. Letztendlich soll die Analyse auf Region 3 eingegrenzt werden. Region 3 wird durch die von den grünen Würfeln umrandeten Würfel symbolisiert. Die Schnittmenge von allen 3 Farben ergibt genau den gelben Würfel, der die Umsätze von Produkt 1 im Zeitraum von Mai bis Juni in Region 3 darstellt. Der Benutzer kann die Dimensionsattribute nun beliebig verändern (z.B. Produkt 2 auswählen) und er erhält sofort die Umsatzzahlen für diese Schnittmenge. Außerdem könnte er die Schnittmenge noch weiter verkleinern indem er eine weitere Dimension hinzufügt (z.B. Kunden) und die Ausprägung auswählt (z.B. männlich). Als Ergebnis erhält der Benutzer dann alle Umsätze von Produkt 1 im Zeitraum von Mai bis Juni in Region 3 von männlichen Kunden.

OLAP-Operationen

Das bereits erwähnte dynamische Arbeiten im multidimensionalen Datenraum und die Operationen die mit dem OLAP-Cube gemacht werden können, sollen jetzt genauer spezifiziert werden. Zur Veranschaulichung wird der dreidimensionale Raum gewählt.

Oft reicht schon ein zweidimensionaler Ausschnitt aus dem OLAP-Cube für Analysen aus. Unter **Rotation** (auch Pivotierung genannt) versteht man das Drehen des Würfels um eine Achse, sodass zwei andere Dimensionen des zweidimensionalen Raums sichtbar werden (vgl. Abb. 15) (Kemper et al., 2006, S. 96).

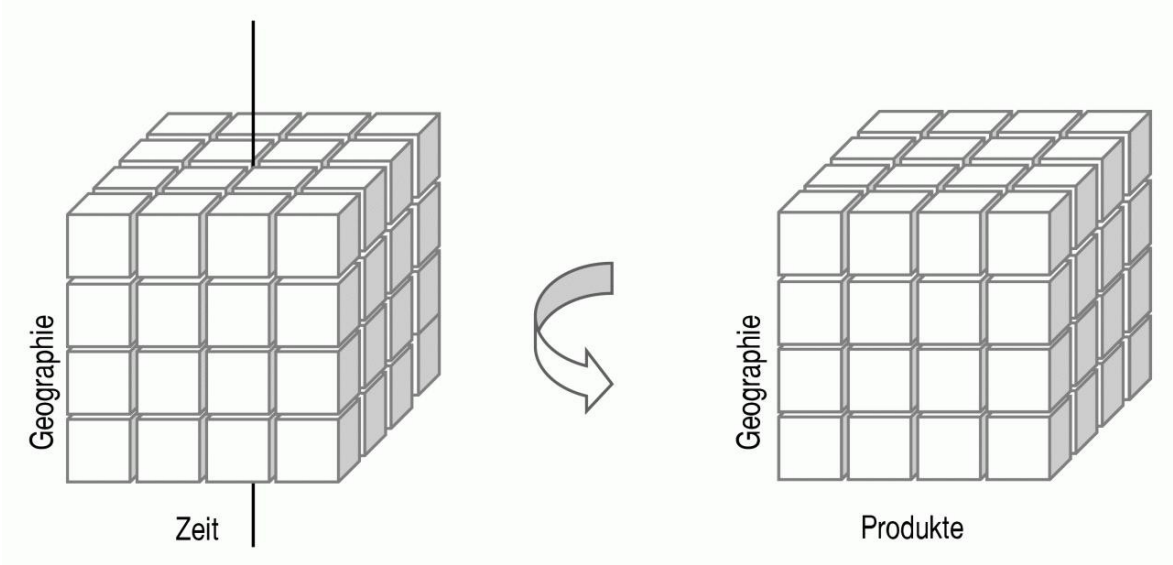

Abb. 15: Rotation des Cubes

(Quelle: Kemper et al., 2006, S. 96)

Um innerhalb der Dimensionshierarchien zu navigieren gibt es zwei mögliche Operationen. Beim **Drill[9]-up** wird die nächst höhere Verdichtungsstufe (z.B. von einzelnen Monaten zu Quartalen) erreicht.

	Produkt A	Produkt B	Produkt C	Produkt D
1. Quartal	140.000	100.000	200.000	120.000
	Drill-Down		Roll-Up	
Januar	40.000	30.000	70.000	40.000
Februar	45.000	35.000	60.000	35.000
März	55.000	35.000	70.000	45.000

Abb. 16: Roll-up & Drill-down

(Quelle: Kemper et al., 2006, S. 97)

[9] engl. für bohren

Das Gegenteil dazu ist der **Drill-down**. Damit gelangt man zur nächst tieferen Verdichtungsstufe (z.B. von Quartalen zu einzelnen Monaten) (vgl. Abb. 16) (Kemper et al., 2006, S. 95f).

Drill-trough und **Drill-across** sind besondere Operationen, da sie den originalen multidimensionalen Datenraum übersteigen. Wenn bei einem Drill-down die feinste Detailliertheit erreicht wird, kann eine weitere Verfeinerung erfolgen. Durch einen Drill-through werden weitere Details verfügbar gemacht, indem eine weitere Datenquelle verwendet wird. Dies geschieht unbemerkt vom Benutzer. Der Drill-across ist ähnlich doch ermöglicht er nicht eine tiefere Analyse, sondern eine breitere. Er macht den Wechsel zwischen OLAP-Cubes möglich. Jedoch müssen beide Cubes dieselben Dimensionen haben. Beispielsweise können dies die OLAP-Cubes des Vertriebs und des Einkaufs eines Unternehmens sein (Kemper et al., 2006, S. 97f).

Die Operation **Slice**[10] die im Beispiel zu Abb. 14 schon benutzt wurde setzt im Prinzip einen Filter auf den Datenbereich. Ein Slice ist eine Scheibe die aus dem Datenwürfel entnommen wird. So kann beispielsweise ein Produktmanager nur die Scheibe seines Produkts herausfiltern (vgl. Abb. 17). (Kemper et al., 2006, S. 98).

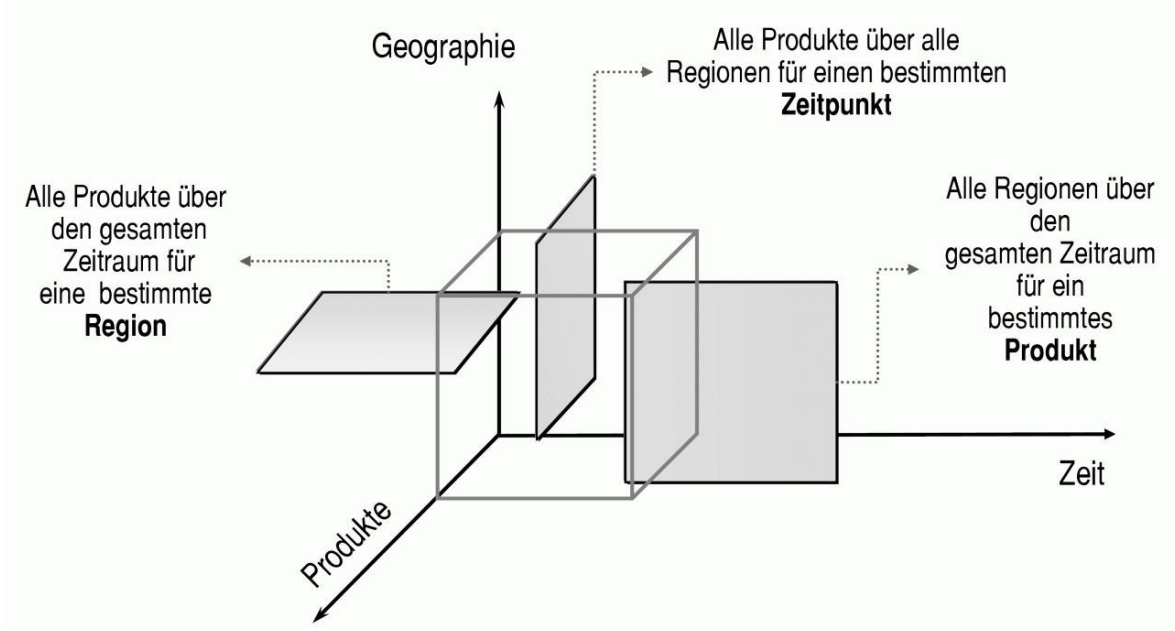

Abb. 17: Slice-Operator

(Quelle: Kemper et al., 2006, S. 98)

[10] engl. für Scheibe, Schnitte

Ein **Dice**[11] ist ein Operator der einen mehrdimensionalen Ausschnitt aus dem gesamten Datenwürfel herausschneidet. Dieser kleine Würfel kann als neuer multidimensionaler Datenraum extrahiert oder weiterverarbeitet werden. Als Extraktion kann dieser kleine Würfel beispielsweise als OLAP-Cube einer Filiale oder einer Abteilung gelten (vgl. Abb. 18) (Kemper et al., 2006, S. 98f).

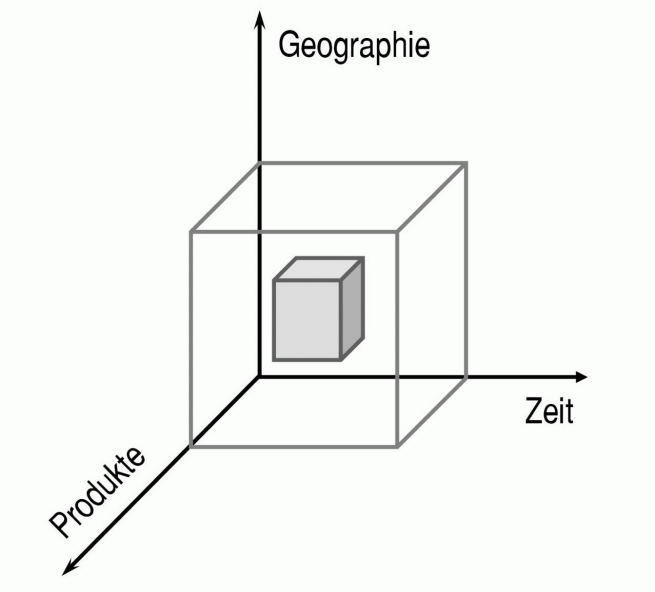

Abb. 18: Dice-Operator

(Quelle: Kemper et al., 2006, S. 98)

Der Operator **Split**[12] wurde auch bereits im Beispiel zu Abb. 14 verwendet. Er ermöglicht das Einfügen von weiteren Dimensionen in ein Analyse und somit eine weitere Detaillierung eines Wertes. Die Umkehrung dazu ist der **Merge**[13]-Operator der Dimensionen entfernt und damit die Detailliertheit der Darstellung verringert (Kemper et al., 2006, S. 99).

Mit dem Abschluss von OLAP ist der BI-Prozess fast abgeschlossen. Die Informationen sind durch die OLAP-Werkzeuge visualisiert wurden und können nun vom Benutzer analysiert werden. Anwendungen wie das Data Mining und die Präsentation der Ergebnisse des BI-Prozesses auch zum eigentlichen Prozess. Diese werden im nächsten Kapitel behandelt.

[11] engl. für Würfel; Spielwürfel

[12] engl. für aufspalten, aufreißen

[13] engl. für fusionieren, zusammenschließen

3 Anwendung der Datenhistorie

Die vom BI-Prozess bereitgestellten Daten können nun angewandt werden. Dies geschieht häufig auf der Ebene des Managements und der Geschäftsführung, jedoch auch Abteilungen wie Marketing und vor allem Controlling greifen auf diese Datenhistorie zurück.

Hauptsächlich wird in diesem Abschnitt beschrieben was Data Mining ist, das neben OLAP die wichtigste BI-Anwendung darstellt. Anschließend werden noch Reporting-Technologien vorgestellt. Zuletzt werden Dashboard- und Portal-Technologien thematisiert.

3.1 Data Mining und Knowledge Discovery

Data Mining ist eine Anwendung auf die Daten des DWH. Hierbei wird versucht bestimmte Regelmäßigkeiten in den Daten zu finden und diese dann für operationale Entscheidungen des zu leitenden Unternehmens einzusetzen. Für diese Anwendung ist es natürlich auch essentiell, dass die Daten des DWH akribisch gepflegt werden und vor allem, dass die Datenhistorie korrekt und vollständig ist.

Es ist leicht vorstellbar, dass das Herausfinden von Regelmäßigkeiten oder Mustern in den Daten vor allem wegen der Komplexität und der Quantität der Daten nicht von humanen Kräften durchgeführt werden kann. Deshalb gibt es verschiedene Techniken und Methoden, die aus den Bereichen der Statistik, der Künstlichen Intelligenz, des Maschinellen Lernens und der klassischen Mustererkennung (pattern recognition) kommen bzw. sich aus diesen Bereichen ableiten lassen. Je nach Aufgabenstellung können eine oder mehrere Methoden zum Einsatz kommen (Kemper et al., 2006, S. 107f).

Definition und Abgrenzung

Datenanalysen zur Mustererkennung hat es bereits in den 60er Jahren gegeben. Doch erst die zentrale Datenhaltung in Data Warehouses (Kemper et al., 2006, S. 106) und das schnelle Wachstum der gespeicherten Datenmengen (Bissantz et al., 2000, S. 379) ermöglichte einen Durchbruch auf breiter Basis in den vergangen Jahren. In diesem Kontext hat sich der Begriff Data Mining etabliert, der das Fördern von wertvollen verschütteten Informationen aus großen Datenbeständen umschreibt (Gluchowski et al., 2008, S. 191). Des Weiteren taucht im Zusammenhang mit Data Mining der Begriff Knowledge Discovery in Databases (KDD) auf und wird häufig fälschlicher Weise synonym mit Data Mining verwendet (Kemper et al., 2006, S. 106). Richtigerweise ist Data Mining ein untergeordneter

Aspekt von KDD (Gluchowski et al., 2008, S. 191) wie Fayyad et al. definieren: „Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data (Fayyad et al., 1996, S. 40).“

KDD-Prozess

Weiterhin beschreiben Fayyad et al. KDD als Prozess, der Ähnlichkeiten mit dem BI-Prozess aufweist, der in Kapitel 2.2 dieser Arbeit beschrieben wird, aufweist (vgl. Abb. 19). Jedoch haben die einzelnen Schritte im KKD-Prozess andere Aufgaben und die Quelldaten für KDD sind bereits aus den transaktionalen Systemen innerhalb des ETL-Prozesses kreiert worden. Die Grundlage bzw. die Datenbasis für KDD ist somit das DWH.

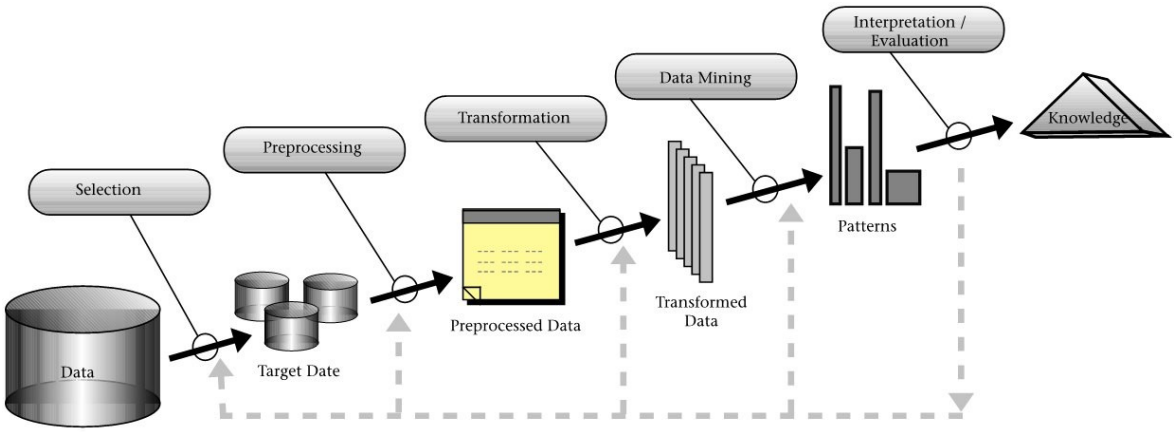

Abb. 19: Übersicht der Schritte des KDD Prozesses

(Quelle: Fayyad et al., 1996, S. 41)

Fayyad et al. beschreiben neun Schritte des KDD-Prozesses (Fayyad et al., 1996, S. 42):

- Im ersten Schritt muss ein Verständnis der Anwendung entwickelt werden und das Ziel des Prozesses festgelegt werden.
- Dann wird bestimmt welche Teilmenge des DWH analysiert werden soll.
- Danach werden die Daten gereinigt und Strategien für fehlende Einträge festgelegt.
- Im vierten Schritt des KDD-Prozesses wird eine möglichst repräsentative Teilmenge herausgefiltert, die die Gesamtheit abbildet.
- Nach dem werden die Methoden des Data Mining ausgewählt mit denen die Ziele aus dem ersten Schritt erreicht werden könnten.
- Diese werden dann im sechsten Schritt festgelegt.

- Der siebte Schritt ist das eigentliche Data Mining. Hierbei wird nach Mustern in den repräsentativen Daten gesucht, es werden Gruppierungen vorgenommen und Klassifikationen bestimmt. Die Qualität dieses Schrittes wird von der Sorgfältigkeit der ersten sechs Schritte beeinflusst.
- Schließlich werden die Ergebnisse aus Schritt 7 interpretiert und evtl. visualisiert. Es kann jedoch zu jedem der ersten sieben Schritte für weitere Iterationen zurückgekehrt werden.
- Beim letzten und neunten Schritt muss das entdeckte Wissen angewandt oder zumindest dokumentiert werden. Außerdem muss überprüft werden ob ggf. Konflikte oder Überschneidungen mit zuvor extrahiertem Wissen bestehen.

Der KKD-Prozess kann beliebig vielen Wiederholung oder Schleifen zwischen verschiedenen Schritten beinhalten. Häufig wird nur der siebte Schritt thematisiert und diskutiert, jedoch sind die anderen Schritte für den Erfolg des Data Mining genauso wichtig.

Data Mining

Wie bereits erwähnt ist Data Mining der siebte Schritt im KDD Prozesses und damit ein Subprozess bei der Erzeugung von Wissen. Dieser Subprozess soll jetzt genauer beschrieben werden.

Zuerst muss festgestellt werden, dass Data Mining immer eine wiederholte, iterative Anwendung von mehreren Data Mining Methoden ist. Dabei sind die Ziele vor allem die „Validierung“ und das „Entdecken“ von Regeln und Mustern. Unter Validierung versteht man das Überprüfen von Hypothesen des Benutzers bzw. des Anwenders über die Muster in den Daten, die ihm schon vorher bekannt waren bzw. von ihm vermutet wurden(Fayyad et al., 1996, S. 43). Es muss also vorher eine Hypothese bekannt sein oder aufgestellt werden. Hypothesen werden meist durch einfaches Beobachten und Überlegen, oder durch Expertenwissen des Anwenders generiert. Nun soll der gesamte KDD-Prozess begonnen werden, da unterschiedliche Hypothesen andere Vorgehensweisen und vor allem unterschiedliche Teildaten benötigen, die im KDD-Prozess ausgewählt werden. Mit statistischen Verfahren und Data Mining-Techniken wird überprüft ob die Hypothese bekräftigt oder widerlegt werden kann. Nun müssen die gewonnen Ergebnisse interpretiert werden, was ein hohen Verständnis der verwendeten Data Mining-Techniken und der geschäftlichen Zusammenhänge erfordert (Martin, 1998, S. 256).

Beim zweiten Hauptziel von Data Mining, dem Entdecken geht es vermehrt um das Erschließen und Finden von neuen, im Vorfeld nicht bekannten Mustern innerhalb der Quelldaten. Das Ziel „Entdecken" muss weiterhin aufgeteilt werden in die Bereiche „Vorhersage/ Prognose" (Vorhersage von zukünftigem Verhalten von Einheiten in den Daten) und „Beschreibung" (Beschreibung von Mustern in für Menschen verständlicher Form) aufgeteilt werden. (Fayyad et al., 1996, S. 43f). Mittels Data Mining-Techniken wird im siebten KDD-Prozessschritt neues, unbekanntes Wissen extrahiert. Dies kann direkt oder indirekt durchgeführt werden. *Direkt* bedeutet, dass der Prozess angeleitet (überwacht) wird. Beispielsweise wird die Kreditwürdigkeit als Zielvariable fest gelegt und soll nun genaue Vorhersagen mithilfe von anderen Variablen ermöglichen:

> Die Kreditwürdig ist hoch, wenn das Haushaltseinkommen größer als 2000 €/ Monat ist, es keine Kontosperre im letzten Quartal gab, ...

Indirekt bedeutet, dass der Prozess nicht angeleitet wird (unüberwacht). In den Daten wird z.B. versucht Korrelation bzw. Zusammenhänge zwischen Datensätzen zu finden. Hierbei liefert beispielsweise die Cluster-Analyse vergleichsweise gute Ergebnisse (Martin, 1998, S. 257).

Wie bereits in der Beschreibung des KDD-Prozesses erwähnt, werden beim Data Mining (Schritt 7 vom KDD-Prozess) die vorher festgelegten Methoden des Data Mining auf die Datenmenge angewandt. Im Folgenden möchte ich die wichtigsten Problemstellungen und die dazu passenden Data Mining-Methoden/ -Techniken nennen. Hierbei ist es wichtig ob es um die Vorhersage oder um die Beschreibung von Datenmustern geht. Je nachdem sind auch die Methoden zu wählen. Doch sind die Grenzen hier nicht scharf. Manche Methoden zur Vorhersage können auf für die Beschreibung geeinigt sein und umgekehrt (Fayyad et al., 1996, S. 44).

Data Mining-Methoden (vor allem mit dem Ziel des Entdeckens) können wie bereits erwähnt in Vorhersage und Beschreibung untergliedert werden. Diese beiden Bereiche können dann weiter wie in Abb. 20 gezeigt, in folgende Punkte eingeteilt werden (Kemper et al., 2006, S. 108f):

Das Ziel der **Deskription** ist das Beschreiben von interessanten Strukturen auf der Basis von deskriptiven statistischen Methoden. Sie kommt vor allem bei der entdeckenden Datananalyse zum Einsatz.

Die **Abweichungsanalyse** wird eingesetzt um untypische oder fehlerhafte Werte zu erkennen. Dabei können z.B. Kreditkartenmissbräuche erkannt werden, wenn außergewöhnlich hohe Summen oder untypische Zahlungsorte herangezogen werden.

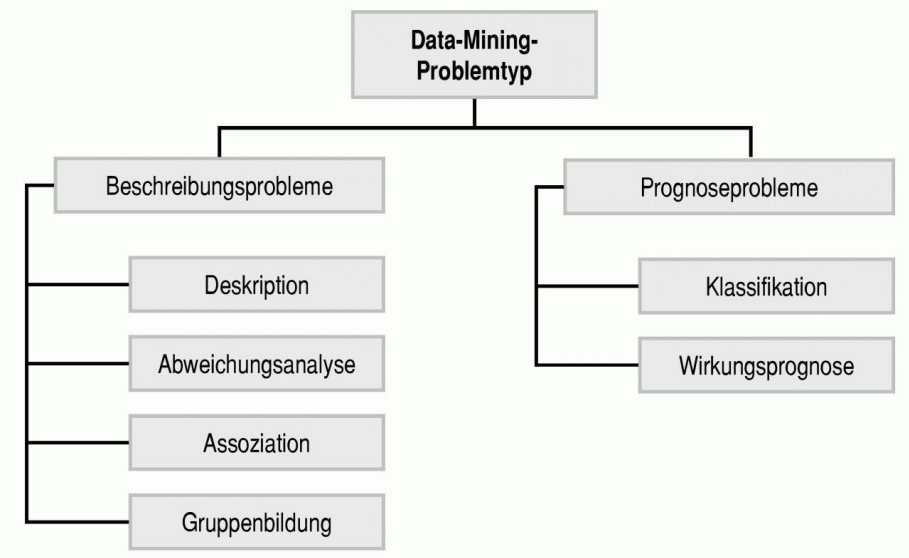

Abb. 20: Gliederung von Data Mining-Methoden

(Quelle: Kemper et al., 2006, S. 108)

Mit Hilfe der **Assoziation** werden Abhängigkeiten zwischen Objekten und Attributen identifiziert. Ein klassisches Beispiel ist die Warenkorbanalyse bei der häufig gemeinsam gekaufte Waren identifiziert werden. Als Data Mining Methoden können Korrelationsanalyse und Assoziationsanalyse eingesetzt werden.

Die **Gruppenbildung** wird oft auch als Segmentierung bezeichnet und soll Cluster gleichartiger Objekte (z.B. Kunden), die ähnliche Merkmale haben, identifizieren .Die Objekte sollen innerhalb des Clusters möglichst Homogen und die Objekte unterschiedlicher Cluster möglichst heterogen sein. Die Merkmale eines Clusters stehen im Vorfeld meist nicht fest. Methoden die zur Gruppenbildung angewandt werden können sind die Clusteranalyse und künstliche neuronale Netze.

Bei der **Klassifikation** stehen Klassen mit bestimmten Eigenschaften im Vorfeld fest. Nun werden weitere Daten zu den bestehenden Klassen zugeordnet indem unabhängige Merkmale ausgewertet werden (z.B. Kreditwürdigkeit im Bezug zu Alter, Einkommen, Berufsgruppe,...). Mögliche Methoden sind Reggresionsanalysen, Klassifikationsbäume, künstliche neuronale Netze und generische Algorithmen.

Die Wirkungsprognose dient ähnlich wie die Klassifikation dazu um von existierenden Daten auf unbekannte Merkmale zu schließen. Passende Methoden sind Reggresionsana-

lysen, Klassifikationsbäume, künstliche neuronale Netze, Box-Jenkins-Analysen und generische Algorithmen.

Um das Thema Data Mining zu erweitern wäre es nun sinnvoll die Methoden und Techniken des Data Mining zu beschreiben und zu analysieren. Dies erfolgt in dieser Arbeit nicht. Ich möchte lediglich auf die Literatur: Gluchowski et al., 2008; Martin, 1998; Fayyad et al., 1996 verweisen. Außerdem ist bisher nicht erfolgt wie man Data Mining konkret auf die Daten eines Data Warehouse anwendet. Auch dies ist nicht Teil dieser Ausarbeitung. Die Darstellung des KDD-Prozesses und der Data Mining-Problemfälle sollen als Veranschaulichung dieser Anwendung eines BI-Systems genügen.

3.2 Reporting

Bisher galt die Aufmerksamkeit immer der Aufbereitung, der Generierung und dem Herausfiltern von Daten, Information und Wissen. Dagegen geht es in diesem Abschnitt vor allem darum, die gewonnen Informationen zu präsentieren und benutzerfreundlich darzustellen.

Berichte werden in allen Unternehmensbereichen und in allen Hierarchieebenen benutzt um schnelle Einsicht in relevante Zusammenhänge zu erhalten. Gluchowski et al. definieren Berichte als Dokumente „die unterschiedliche Informationen für einen bestimmten Untersuchungszweck miteinander kombinieren und in aufbereiteter Form vorhalten“ (Gluchowski et al., 2008, S. 206). Die erforderlichen Berichte werden aus Anwendersicht automatisch generiert. Somit kommt dem Anwender eine passive Rolle zu in der er die Information des Berichts lediglich interpretieren muss. Das innerbetriebliche Berichtswesen ist meist relativ aufwändig und reicht über das vom Gesetzgeber geforderte Mindestanforderung an die Rechnungslegung hinaus. Der Grund dafür unmittelbar ersichtlich, da die Geschäftsleitung bzw. das Management in periodischen Abständen und im Bedarfsfall bestimmte Fakten benötigt um Trends und Entwicklungen zu erkennen und entsprechend auf diese zu reagieren (Gluchowski et al., 2008, S. 206). Diese Berichte sind verständlicherweise nur von Nöten, wenn der Empfänger kein Zugriff auf das OLAP-System hat – dort werden die Informationen schließlich in Echtzeit[14] dargestellt und sind somit

[14] “Echtzeit” ist hier relativ zu betrachten, da das DWH die Daten meist „nur“ periodisch aus den operativen Systemen extrahieren. Jedoch ist OLAP immer genau so aktuell wie das DWH – was bei statischen Berichten (PDF, Papier) nicht der Fall ist.

qualitativ hochwertiger. Aber auch mit OLAP-Zugriff können Berichte durchaus nützlich sein und bieten eine einfache und unkomplizierte Form der Präsentation.

Berichte unterscheiden sich vor allem durch ihren Zweck und Inhalt aber auch die Darstellungsform ist eine wichtige Unterscheidungsform. So können grafische und textliche Darstellungsformen einzeln oder gemischt verwendet werden. Grafische Berichte beinhalten die üblichen Geschäftsgrafiken wie Kreis-, Balken- und Liniendiagramme und begrenzen sich meinst auf verhältnismäßige Angaben zur Geschäftslage. Textliche Varianten haben zumeist tabellarisch angeordnete Information zur Abbildung von absoluten oder relativen Zahlen und Inhalten aber auch ausformulierte Texte können vorkommen. Die strukturelle Anordnung der eingesetzten Elemente trägt zur Übersichtlichkeit und Verständlichkeit der Berichte bei. Auch die Abstimmung auf den Adressaten ist ein Erfolgsfaktor des Reporting. (Gluchowski et al., 2008, S. 208).

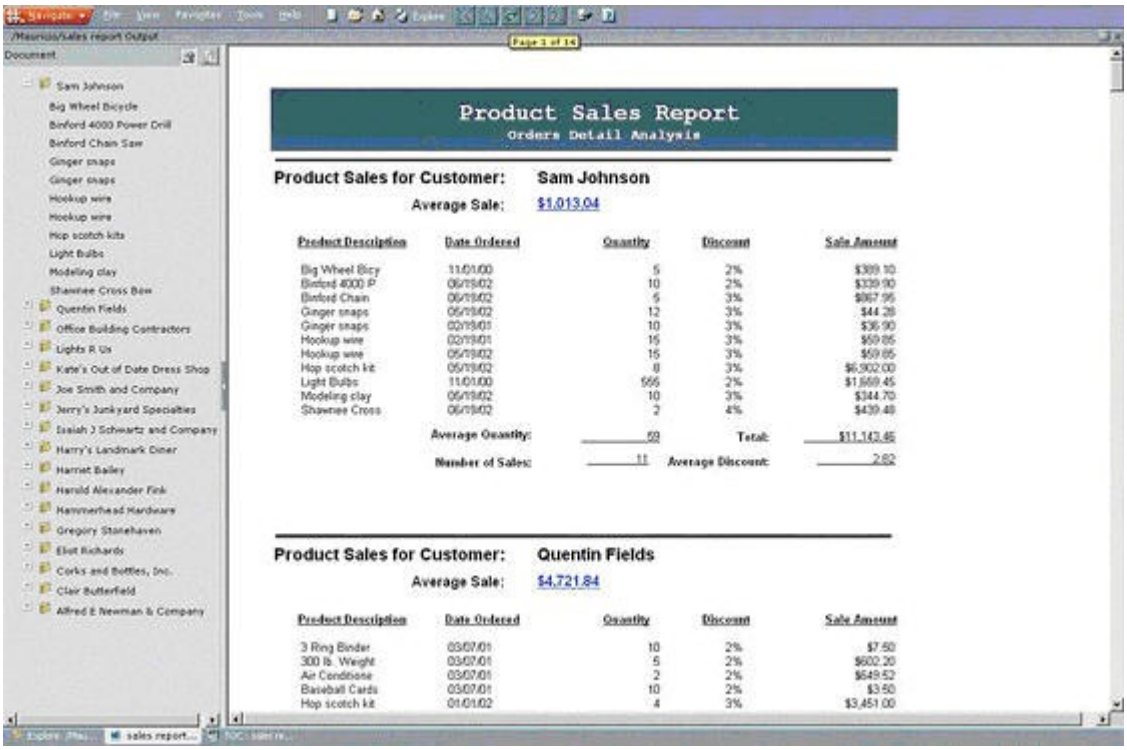

Abb. 21: Beispiel eines Berichts in tabellarischer Form

(Quelle: http://oraclebizint.wordpress.com/2007/07/19/hyperion-system-9-bi-overview - 06.09.2008)

Berichte oder Reports haben zwei Ebenen, einerseits die abstrakte Schablone mit Formatierungsmöglichkeiten und andererseits das konkrete Berichtsergebnis. Die Reportschablone wird mit einer Programmiersprache, die auf die Datenaufbereitung zugeschnitten ist, erstellt. Die Inhalte werden aus den Datenbanken bzw. dem DWH extrahiert. Modere Berichtssysteme unterstützen jedoch auch die Berichtserstellung mittels Drag&Drop, so-

dass keine Programmierkenntnisse nötig sind. Mittels eines Werkzeugkastens lassen sich die gewünschten Daten, in der gewünschten Granularität auf den Bericht ziehen. So kann schon während der Gestaltung ein optischer Eindruck vom Berichtsergebnis entstehen. Außerdem ist bei derartigen Berichtssystemen die Einbindung von grafischen Elementen recht simpel. Einmal erstellte Reportschablonen lassen sich speichern und wiederverwenden. Lediglich die Daten selbst werden aktualisiert. Weiterhin gibt es Möglichkeiten Berichtssummen und Anteile an den Summen zu errechnen, die nicht aus dem DWH extrahiert werden können.

Berichte werden meist optimiert um gedruckt zu werden. Doch gibt es aus Reporting-Tools mit der Möglichkeit des Web-Reporting d.h. die Berichte werden für das Internet optimiert (Gluchowski et al., 2008, S. 211f). Dieser Bereich grenzt jedoch schon an die Funktionalitäten von Dashboards und Portaltechnologien die im nächsten Abschnitt behandelt werden.

3.3 Dashboards und Portale

Im Gegensatz zu Reporting-Lösungen bieten Dashboards[15] und BI-Portale die Informationen aus dem DWH nicht in druckoptimierter Form an, sondern werden für die Interaktive Verwendung am Bildschirm konzipiert. Dabei werden die Daten so präsentiert und angeordnet, dass sie auf einer oder einigen wenigen Bildschirmseiten Platz finden(Gluchowski et al., 2008, S. 214). Es geht in diesem Abschnitt also wiederum um die Präsentation und nicht um die Generierung und Aufbereitung der Daten.

Dashboards

Dashboards sind in ihrem Erscheinungsbild sehr unterschiedlich. Jedoch haben fast alle Dashboards die Gemeinsamkeit, dass sie grafische Elemente zur Veranschaulichung der Informationen beinhalten. Stephen Few definiert Dashboards folgendermaßen: „A Dashboard is a visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the information can be monitored at a glance (Few, 2006, S. 34).“

Das Dashboards eine visuelle Orientierung haben liegt nicht daran, dass der Arbeitsbereich von Anwendern verschönert werden soll, sondern weil Grafiken die Information effizienter und schneller kommunizieren als rein textliche Berichte. Um lediglich eine be-

[15] engl. für Armaturenbrett

stimmtes Zielinformation aus dem DWH herauszubekommen, ist es oft nötig in eine Sammlung von Daten einzusehen, die sonst keine Beziehungen untereinander haben. Oft kommen diese Informationen aus verschiedenen Quellen und beziehen sich auf unterschiedliche Unternehmensbereiche. Die nötigen Informationen sind so auf einer Bildschirmansicht zu platzieren, dass der Anwender (nicht zwingend das Management oder die Geschäftsleitung) seine Aufgabe möglichst schnell erledigen kann. Die bereits erwähnte Forderung an ein Dashboard, die nötigen Informationen auf einer Bildschirmansicht zu platzieren, wird dadurch gerechtfertigt, dass der Anwender die Informationen auf einen Blick sehen können soll. Bevor man ein Dashboard entwirft bei dem man scrollen muss um alle Informationen zu sehen, sollte man lieber mehrere Dashboards entwerfen und die Informationen sinnvoll aufteilen. Das Ziel dabei ist es alle benötigten Angaben möglichst schnell aufzunehmen und zu verinnerlichen(Few, 2006, S. 35).

Abb. 22: Beispiel eines Dashboards von Oracle Corp. - angezeigt im Webbrowser

Hier werden Verkaufs-zahlen dargestellt. Dabei sind alle Einheiten von visueller Natur.

(Quelle: Few, 2006, S. 13)

Heutzutage ist die beste Möglichkeit Dashboards anzuzeigen wohl noch ein klassischer Webbrowser (siehe Abb. 22). Jedoch ist davon auszugehen, dass im Zuge von Mobile Enterprise Solutions immer mehr mobile BI-Solutions eingesetzt werden(z.B. die Plattform

MoBi von Business Objects[16]), bei welchen Dashboards auf Mobiltelefonen und Organizern angezeigt werden.

Eine weitere Forderung an Dashboards ist die Zugänglichkeit an Detailinformationen. Während ein Dashboard an sich nur die wichtigsten Informationen auf einen Blick liefert, sind oft die damit zusammenhängenden Details vonnöten. So sollen Dashboards diese Information schnell und einfach zugänglich machen. Dies kann in Form von einem Wechsel zu Detailansichten die dem Dashboard hinterlegt sind geschehen (Few, 2006, S. 36). Oder auch eine Verlinkung zum OLAP-System erscheint als sinnvoll.

Zusammenfassend ist zu sagen, dass Dashboards Informationen auf einen Blick liefern sollen und den Weg zu den Details dieser Informationen verfügbar macht.

BI-Portale

Eine Abgrenzung von Dashboards und Business Intelligence-Portalen lässt sich nicht immer trennscharf vornehmen (Gluchowski et al., 2008, S. 216). Hinzu kommt, dass in der Wirtschaft beide Begriffe häufig nahezu synonym verwendet werden.

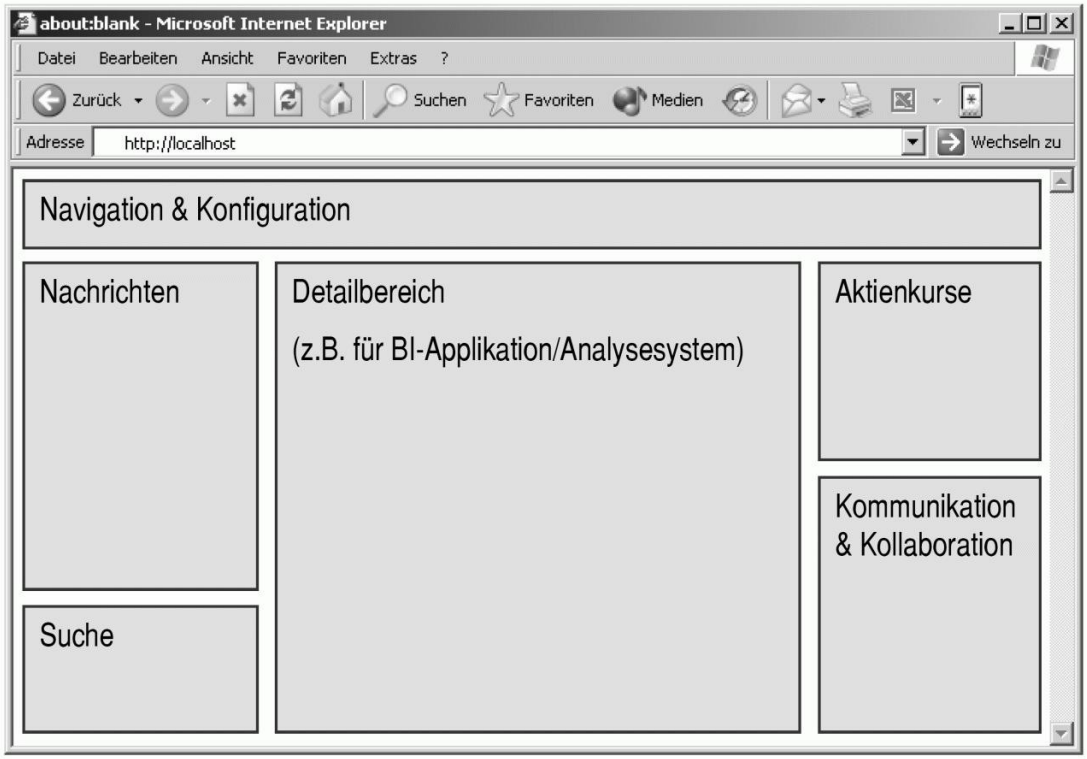

Abb. 23: Mögliche Gliederung eines BI-Portals

(Quelle: Kemper et al., 2006, S. 135)

In meinen Untersuchen habe ich festgestellt, dass BI-Portale lediglich eine Erweiterung von Dashboards sind. So bieten die Portale verschieden interne und externe Informatio-

[16] http://www.enterprise-dashboard.com/2008/02/27/mobile-bi-dashboards-from-business-objects/ (08.09.08)

nen an, die dann auf einer Bildschirmansicht dargestellt werden (siehe Abb. 23). Interne Inhalte werden aus unternehmenseigenen Informationssystemen bezogen. Dies können eigene Pressemitteilungen, Managementberichte oder bestimmte Memos sein. Ganz zentrale interne Informationen sind jedoch Unternehmensdaten die aus dem DWH extrahiert, aufbereitet und dann ins Portal integriert werden. Und dabei kann es auch zur Verschmelzung mit Dashboards. kommen. Diese können in das Portal integriert werden. Z.B. bietet die „it innovations GmbH“ Portal-Lösungen an, bei denen Dashboards integriert werden[17]. Externe Informationen stammen von Drittanbietern und umfassen wettbewerbsrelevante Daten wie Aktienkurse, Analysen oder Nachrichten (Kemper et al., 2006, S. 135).

Klassische Portalinhalte sind in der Regel statisch und werden erst durch die Integration von Anwendungen wie webbasierten Analysesystemen oder interaktiven Dashboards zu einem dynamischen Präsentationstool. Weiterhin werden Portale oft durch Kommunikationsmöglichleiten ergänzt. Diese reichen von klassischen Anwendungen wie E-Mail und Kalender bis hin zu Voice-over-IP, Chats und Foren (Kemper et al., 2006, S. 135f).
Ähnlich wie bereits im Kontext der Dashboards beschrieben, wird auch bei BI-Portalen der Webbrowser zur Darstellung verwendet und auch hierbei sind mobile Lösungen bereits vorhanden. Außerdem ist eine individuelle Anpassung an den Benutzer nötig. Hierbei sind gruppenbezogene und individuelle Personalisierungen denkbar. Dadurch können die Benutzer die Darstellung (Farben, Layouts) und die Inhalte bestimmen und an die Bedürfnisse anpassen (Kemper et al., 2006, S. 136f).

[17] http://www.it-innovations.de/newsletter/newsletter/business+intelligence-portale+mit+dem+sharepoint+server+2.htm (09.09.08)

4 Optimierung der Datenhistorie

In den zwei vorangegangen Kapiteln wurde zum einen die Bereitstellung einer Datenhistorie und zum anderen die Anwendung derselben thematisiert. In diesem Kapitel soll es nun darum gehen, wie eine Datenhistorie durch die Modellierung von Slowly Changing Dimensions optimiert werden kann. Dazu sollen zuerst einige Grundlagen zur Datenhaltung geliefert werden. Dann wird auf die verschiedenen Typen von SCD eingegangen und wie diese modelliert werden können. Abschließend wird an einem Beispiel gezeigt, wie Microsoft® SQL Server 2005® für Anwender eine Unterstützung liefert um SCD zu modellieren. Dabei werden die SCD Typen 1 und 2 nochmals veranschaulicht.

4.1 Grundlagen zur Datenhaltung

Im Kontext der Datenhaltung kann man verschiedene Ansätze der Sicherung von Daten unterscheiden. Zu erst werden an dieser Stelle einige Herangehensweisen von Backups beschrieben. Dann folgen eine Begriffsbestimmung zu Archivierungen und eine Abgrenzung der beiden eben genannten Begrifflichkeiten. Schließlich werden kurz die Grundlagen der Historisierung erörtert.

4.1.1 Backup

Ein Backup ist eine Sicherung von Datenbeständen auf speziellen Sicherungsmedien, um bei technischen Systemfehlern Datenbereiche wiederherstellen zu können (Kemper et al., 2006, S. 66). Sollte es vorkommen, dass ein Speicherfehler auftritt, wird die jüngste, also die aktuellste Sicherungskopie wieder eingespielt. Heutzutage ist das Sichern der kompletten Datenbank oft nicht mehr möglich. Dies liegt daran, dass zum einen die Datenmengen steigen. Daraus folgt, dass zum Sichern immer größere Offline-Phasen[18] benötigt werden. Zum Anderen wachsen die Verfügbarkeitsanforderungen welche im Interessenkonflikt mit Offline-Phasen stehen. Deshalb sind neue Techniken für die Sicherung und die Wiederherstellung notwendig (Störl et al., 1998, S. 113). Solche neuen Sicherungsmöglichkeiten sind unter anderem

- das partielle Backup,

  Unter partiellem Backup versteht man die Sicherung von Teilen der Datenbanken. Teilsicherungen laufen deutlich schneller ab und dadurch ist die Datenbank schneller wieder verfügbar. Außerdem kann die Sicherungsstrategie variabler ge-

[18] Im Offline-Modus kann auf die Datenbank nicht zugegriffen werden. Alle Benutzer und vor allem alle Anwendungen müssen von der Datenbank abgemeldet werden.

staltet werden. D.h. Datenbereiche die sich häufiger ändern, können auch häufiger gesichert werden (Störl et al., 1998, S. 114).

- das inkrementelle Backup,

  Inkrementelles Backup sichert lediglich die seit einem bestimmten Zeitpunkt veränderten Daten. Solche Zeitpunkte sind z.B. das letzte komplette oder inkrementelle Backup. Wie offensichtlich erkennbar ist, wird dadurch der Platzbedarf reduziert und die Offline-Phasen werden verkürzt, besonders wenn die Änderungen nur einen kleinen Teil der Gesamtdaten betreffen (Störl, 2001, S. 26).

- das parallele Backup,

  Bei der Erstellung von Sicherungen ist oft nicht das Lesen der Datenbank sondern das Speicher der Sicherheitskopie der Engpass. Deswegen wird beim parallelem Backup während einer Sicherung gleichzeitig auf mehrere Sicherheitsmedien geschrieben (Störl, 2001, S. 26).

- und das Online-Backup.

  Unter Online-Backup wird die Sicherung der Datenbank oder von Teilen der Datenbank im laufenden Betrieb verstanden. Der große Vorteil besteht darin, dass der Betrieb auf der Datenbank ganz oder teilweise weitergehen kann (Störl et al., 1998, S. 117)

Um die Sicherheitskopien wieder in die Datenbank einzuspielen, gibt es analoge Recovery-Mechanismen zu den genannten Backup-Verfahren. Auch hier spielt bei der Offline-Recovery die Datenmenge und die Offline-Dauer eine wichtige Rolle. Bei der Online-Recovery kann auf nicht von der Wiederherstellung betroffene oder bereits wiederhergestellte Daten zugegriffen werden (Störl, 2001, S. 28f).

4.1.2 Archivierung

Bei Archivierungen geht es darum, Datenbereiche mit Hilfe von Sicherheitskopien bei fachlichem Bedarf wiederherstellen zu können. Solche fachlichen Bedarfe können beispielsweise Rechtsstreitigkeiten sein. (Kemper et al., 2006, S. 66). Aber auch für Dokumentationszwecke, Wiederverwendung oder zur Systementlastung werden Archivierungen durchgeführt (Herbst, 1997, S. 7f).

Eine Archivierung wird durch folgende Eigenschaften gekennzeichnet (Schaarschmidt, 2001, S. 32):

- Archivierung logischer Datengranulate: Die Daten die archiviert werden sollen, können keine physischen Granulate (d.h. zusammenhängende Speicherbereiche) wie Dateien oder Seiten sein. Es sind immer logische Datengranulate wie Tupel[19], Relationen[20] oder Views[21].
- Benutzerveranlassung: Ein Archivierungsvorgang wird durch einen Benutzer entweder manuell oder ein ereignisgesteuert ausgelöst. Dabei ist es wichtig die zu archivierenden Daten zu kennen.
- Datenauslagerung: Die archivierten Daten sind logisch und physisch von den operativen Daten (in diesem Fall: die aktuellen Daten im Data Warehouse oder im operativen, transaktionalen System) getrennt. Beim Archivieren werden Daten der Datenbank in ein Archiv verschoben.
- Archivzugriff: Die Zugriffsmöglichkeit auf das Archiv ist gewährleistet. Außerdem können die Daten bei Bedarf in die Datenbank zurückgeführt werden.

Häufig wird die Archivierung im Datenbankumfeld mit dem Begriff Backup gleichgesetzt. Bei einem Backup wird jedoch eine Kopie der Datenbank erzeugt, mit dem Ziel der Datensicherung. Kommt es zu Speicherfehlern in der Datenbank, kann die Kopie zur Sicherung herangezogen werden. Im Gegensatz zur Archivierung wird beim Backup die Datenbank nicht entlastet, d.h. Daten werden nicht ausgelagert und die Datenbank nicht verändert. Dabei ist die langfristige Aufbewahrung und der inhaltsbezogene Zugriff nicht wichtig. Vielmehr ist die Aktualität der Datenbankkopie von Bedeutung (Schaarschmidt, 2001, S. 33).

4.1.3 Historisierung

Mit Hilfe von Konzepten der Historisierung, können Änderungen von Attributsausprägungen, Beziehungen und Entitäten im Zeitlauf dokumentiert werden und deren unterschiedliche fachlichen Zuständen können ausgewertet werden (Kemper et al., 2006, S. 66).

Wie bereits in Abschnitt 2.2.2 erwähnt und von Inmon gefordert, sind Elemente zum Nachvollziehen der Chronologie der „Schnappschüsse“ im DWH nötig. Lehner verstärkt in seinen Anforderungen an ein DWH die Notwendigkeit einer Historisierung. Er be-

[19] Datensätze

[20] Beziehungen, Zusammenhänge

[21] Aliase (Ersatzname für Kopien von Variablen) für Abfragen der Datenbank

schreibt diese Forderung so, dass Daten, die in ein Data Warehouse-Systeme einmal eingebracht wurden, nicht überschrieben, sondern ergänzt werden, sodass im Laufe der Zeit eine Historisierung der extrahierten Zustände der Quellsysteme erreicht wird. Weiter charakterisiert Lehner diese Forderung an ein DWH als optional, da die Realisierung meinst durch zu wenigen Speicherplatz bzw. durch die nötige Aufrechterhaltung der Performance beschränkt wird (Lehner, 2003, S. 10). Wie notwendig Historisierungen jedoch sein können, wird im nächsten Abschnitt deutlich.

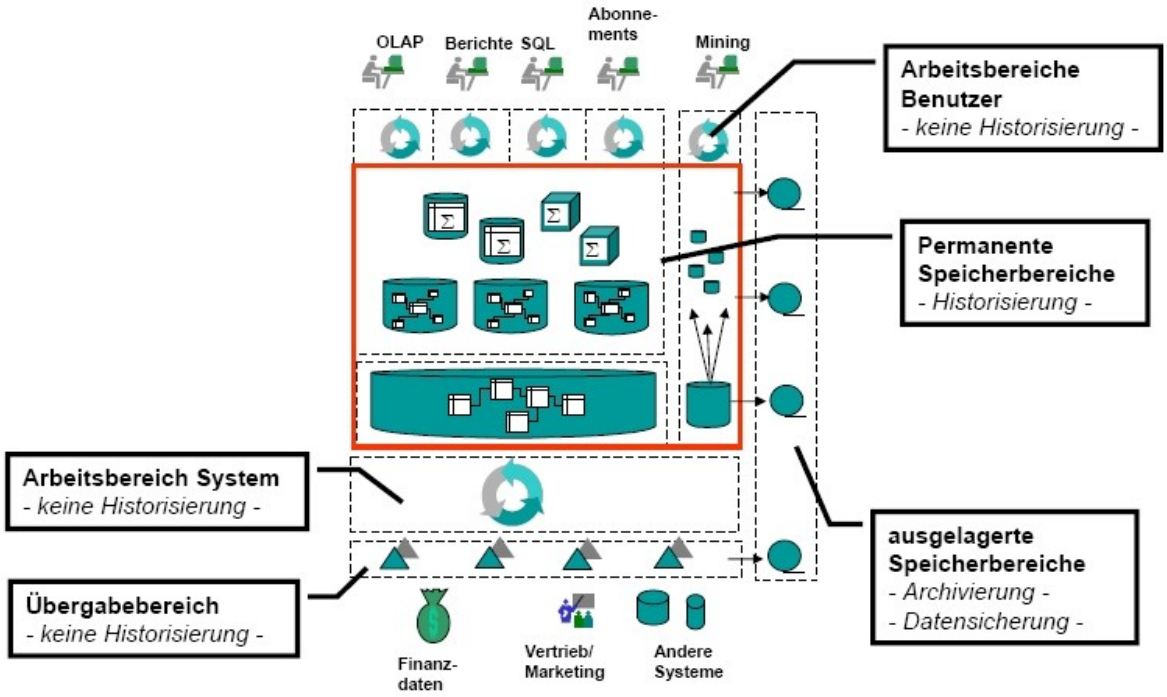

Abb. 24: Hisorisierungsbereiche im BIS

(Quelle: Finger, 2002, S. 5)

Abb. 24 Zeig zusammenfassend was in einem BI-System historisiert wird. Deutlich zu sehen ist, dass Datensicherung (Backup) und Archivierung aus dem BI-System ausgelagert sind und nur für technische Notfälle oder fachliche Bedarfe benötigt werden.

4.2 Slowly Changing Dimensions

Dieser Abschnitt bildet den Kern dieser Ausarbeitung und stellt die Modellierung von Slowly Changing Dimensions dar. Erst durch die Behandlung von SCD kann ein DWH eine Optimierung der Datenhistorie erfahren. Obwohl dieser Teil der Arbeit verhältnismäßig kurz ausfällt, ist dieser Kerninhalt ohne die nötige Grundlage der anderen Kapitel nicht aufbereitbar.

In Kapitel 2.2.2 wurde bei der Modellierung des DWH davon ausgegangen, dass die jede Dimension unabhängig von allen anderen ist (vgl. Abb. 10 die Dimensionstabellen sind miteinander in keiner Weise verknüpft). Besonders wurde die Annahme getroffen, dass Dimensionen wie 'Produkt' und 'Kunde', unabhängig von der Dimension 'Zeit' sind. In der Realität ist diese Annahme nicht immer korrekt, denn mit der Zeit verändern sich Beschreibungen und Formulierungen von reellen Produkten gelegentlich. Kunden verändern sich sogar mit gewisser Regelmäßigkeit. Sie wechseln ihre Namen, heiraten, lassen sich scheiden und wechseln den Wohnort. Ein weiteres Beispiel sind die Vertriebsnetze der Unternehmen, die ihre Zuständigkeitsbereiche ändern. Dabei werden die Regionen und Gebiete neu gegliedert, sodass bestimmte Regionen von anderen Verantwortlichen betreut werden. Diese Veränderungen der Dimensionen müssen im Data Warehouse-System berücksichtigt und dargestellt werden (Kimball, 1996, S. 100).
Die Forderung nach einer korrekten Historie wurden in dieser Arbeit unter anderem schon im Abschnitt 3.1 thematisiert. Diese korrekte Historie soll nicht nur den aktuellen Zustand eines Unternehmens im DWH darstellen, sondern auch die verschiedenen Zustände im Laufe der Zeit.

Das Lösungskonzept für die sich verändernden Dimensionen ist nicht, all diese Veränderungen in die Faktentabelle zu schreiben oder alle Dimensionen zeitabhängig zu modellieren. Solche Vorgehensweise würde das DWH aufblähen, die Verständlichkeit bzw. die Einfachheit des DWH vernichten und die Performance also die Antwortzeiten des DWH erheblich steigen lassen. Stattdessen wird an der Tatsache, dass die meisten Dimensionen weiterhin 'fast' konstant (sich im Laufe der Zeit nicht verändern) sind, festgehalten. Mit einem relativ kleinen Zusatz kann die unabhängige Dimensionsstruktur beibehalten und gleichzeitig die Veränderung an den Dimensionen dargestellt werden. Diese fast konstanten Dimensionen werden 'sich langsam verändernde Dimensionen' oder im englischsprachigen Raum 'Slowly Changing Dimensions' (SCD) genannt (Kimball, 1996, S. 100).

Sobald eine Dimension auftritt die sich verändert, müssen folgen Entscheidungen getroffen werden. Jede dieser Entscheidungen hat einen anderen Grad bzw. eine andere Qualität der Historisierung zufolge (Kimball, 1996, S. 100f):

- Die alten Dimensionen werden überschrieben, der Inhalt geht verloren und die Historie wird nicht gespeichert.

- Beim Zeitpunkt der Veränderung wird ein neuer, zusätzlicher Dimensionseintrag mit den neuen Werten kreiert. Dabei muss die Historie zwischen der neuen und alten Dimension sehr akkurat aufgeführt werden.
- Die originale Dimension wird mit einer neuen „Aktuell"-Spalte bestückt, die die neuen Werte enthält. Die alten Felder mit den alten Werten bleiben dabei erhalten. Daraus ist es möglich die Historie vorwärts vom neuen Eintrag aus und rückwärts vom Originaleintrag aus zu beschreiben. Jedoch wird nur der aktuelle und der originale Eintrag gesichert.

Diese drei Wahl- bzw. Entscheidungsmöglichkeiten werden Type 1 SCD, Type 2 SCD und Type 3 SCD genannt. Im Folgenden werden diese drei Typen genauer spezifiziert und am Beispiel „Mary Jones", die als Kunde geführt wird, verdeutlicht. Bis zum 07.07.2007 war Mary Jones nicht verheiratet und ihr Familienstand hatte in der Kundendatenbank den Wert „S" (Single). Am 07.07.2007 heiratete sie. Der Zustand verändert sich in „M" (married). An diesem Beispiel wird nun gezeigt, wie die Veränderungen im DWH gehandhabt werden (aus: Kimball, 1996, S. 100f). Zur weiteren Veranschaulichung wird noch gezeigt was passieren würde, wenn Mary Jones am 20.09.2008 verwitwen würde und der Familienstand den Zustand „W" (widowed) erhalten würde. Die letzte Veränderung wird nur an den Tabellenbeispielen gezeigt und im Text nicht berücksichtigt.

4.2.1 Type 1 SCD

Die erste Entscheidungsmöglichkeit beim Auftritt einer Veränderung ist am einfachsten zu realisieren und stellt im Grunde auch keine Historisierung dar. Bei Type 1 SCD wird der Familienstand einfach mit dem Wert „M" überschrieben. Es gibt also keinen Eintrag, der den alten Wert speichert. Weitere Änderungen werden hierbei nicht benötigt. Auch werden keine Schlüssel in der Datenbank verändert. Jedoch ist dieser Typ von SCD nicht unsinnig, wie es vielleicht auf den ersten Blick erscheint. Zum Beispiel könnte sich herausstellen, dass Mary Jones schon immer verheiratet war und der alte Eintrag schlichtweg ein Fehler war. Um diesen Fehler zu beseitigen, wird der Familienstand von Mary Jones einfach upgedatet (siehe Abb. 25).

1.

Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1	Thompson	Joe	NY	S	M	31Dec1999:23:59
2	Webster	Lisa	LA	M	F	31Dec1999:23:59
3	**Jones**	**Mary**	**Dallas**	**S**	**F**	**31Dec1999:23:59**
4	Smith	Tom	Miami	S	M	31Dec1999:23:59

2.

Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1	Thompson	Joe	NY	S	M	31Dec1999:23:59
2	Webster	Lisa	LA	M	F	31Dec1999:23:59
3	**Jones**	**Mary**	**Dallas**	**M**	**F**	**07Jul2007:11:59**
4	Smith	Tom	Miami	S	M	31Dec1999:23:59

3.

Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1	Thompson	Joe	NY	S	M	31Dec1999:23:59
2	Webster	Lisa	LA	M	F	31Dec1999:23:59
3	**Jones**	**Mary**	**Dallas**	**W**	**F**	**20Sep2008:11:59**
4	Smith	Tom	Miami	S	M	31Dec1999:23:59

Abb. 25: Type 1 SCD – Beispiel

(Quelle: eigene Darstellung)

4.2.2 Type 2 SCD

Die weitaus öfters benutzte Entscheidungsmöglichkeit ist Type 2 SCD in der zwei (oder mehrere) korrekte Eintragungen für Mary Jones existieren. Der originale Eintrag mit dem Familienstand „Single“ hat dann eine Gültigkeit für alle vor dem 07.07.2007 gemachten Einträge und der neue Eintrag hat eine Gültigkeit für alle ab dem 07.07.2007 gemachten Einträge in der Datenbank. Dies wird erreicht indem ein zweiter Dimensionseintrag angelegt wird in der Mary Jones den Familienstand „Verheiratet“ hat. Der zweite Eintrag muss jedoch auch einen neuen Schlüssel bekommen damit die Dimension Kunde, unter welcher Mary Jones geführt wird, ein eindeutiges Schüssel hat (siehe Abb. 26). Die führt zu einem wichtigen Prinzip beim Modellieren von SCD:

> ***Beim Einsatz von Type 2 SCD ist es notwendig, dass die Dimensions-Schlüssel weiterhin eindeutig sind. Meist ist dabei ausreichend dem Original-Schlüssel zwei oder drei Dezimalstellen hinzuzufügen um den Prozess der Schlüssel-Generierung möglichst simpel zu halten.***

Im Mary-Jones-Beispiel kann die Kundennummer nicht mehr als Schlüssel in der Kunden-Dimension verwendet werden, denn bei mehrere Einträgen die Mary Jones als Kunde beschreiben, wird eine eindeutiger Schlüssel benötigt. Wenn an die Kundennummer le-

diglich zwei Dezimalstellen angehängt werden, sind bereits 100 Schnappschüsse von Mary Jones' Familienstand möglich. Bei drei Dezimalstellen sind sogar 1000 Schnappschüsse realisierbar. An dieser Stelle ist anzumerken, dass die Erweiterung der Dimensions-Schüssel bereits während dem, in Abschnitt 2.2.1 beschrieben, ETL-Prozess generiert werden müssen. Daraus ergibt sich ein weiteres Prinzip der SCD-Modellierung:

> ***Das Erstellen von eindeutigen Dimensions-Schlüssel ist Aufgabe des Data Warehouse-Teams im Unternehmen. Außerdem sind beim Erstellen der Dimensions-Schlüssel Metadaten nötig, um nachvollziehen zu können welche Schlüssel bereits verwendet wurden.***

Abb. 26: Type 2 SCD – Beispiel

1. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1.01	Thompson	Joe	NY	S	M	31Dec1999:23:59
2.01	Webster	Lisa	LA	M	F	31Dec1999:23:59
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	**31Dec1999:23:59**
4.01	Smith	Tom	Miami	S	M	31Dec1999:23:59

2. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1.01	Thompson	Joe	NY	S	M	31Dec1999:23:59
2.01	Webster	Lisa	LA	M	F	31Dec1999:23:59
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	**31Dec1999:23:59**
3.02	**Jones**	**Mary**	**Dallas**	**M**	**F**	**07Jul2007:23:59**
4.01	Smith	Tom	Miami	S	M	31Dec1999:23:59

3. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1.01	Thompson	Joe	NY	S	M	31Dec1999:23:59
2.01	Webster	Lisa	LA	M	F	31Dec1999:23:59
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	**31Dec1999:23:59**
3.02	**Jones**	**Mary**	**Dallas**	**M**	**F**	**07Jul2007:23:59**
3.03	**Jones**	**Mary**	**Dallas**	**W**	**F**	**20Sep2008:23:59**
4.01	Smith	Tom	Miami	S	M	31Dec1999:23:59

(Quelle: eigene Darstellung)

Der Einsatz von Type 2 SCD partitioniert die Historie zwangsläufig in Bereiche. Im Mary-Jones-Beispiel wird die „Single"-Beschreibung für alle Faktentabelleneinträge benutzt die vor dem 07.07.2007 kreiert wurden. Ab dem 07.07.2007 wird der „Verheiratet"-Dimensionstabelleneintrag für die Einträge in der Faktentabelle benutzt. In diesem Fall ist der Übergang vom „Single-" in den „Verheiratet-Zustand" nahtlos am Hochzeitstag geschehen. Die Bereiche, in die die Historie in diesem Fall geteilt wurde, sind die Bereiche vor und nach der Hochzeit, dem 07.07.2007. Sollte bei einer Auswertung, die auf dem DWH

geschieht, die Beschränkung nur auf dem Kundennamen „Mary Jones“ liegen, werden beide Dimensionseinträge vereint und der Faktentabelle zugeordnet. Der erste Dimensionseintrag vereint nur die Fakten vor der Hochzeit und der zweite Eintrag vereint nur die Fakten ab der Hochzeit. So kann eine Abfrage für alle Fakten von Mary Jones realisiert werden. Dabei ist es nicht wesentlich notwendig ein Gültigkeitsdatum[22] für die Veränderung bei Mary Jones zusetzen. Außerdem ist es unnötig die Dimensionstabelleneinträge mit Zeitwerten zu beschränken, um die richtige Antwort zu bekommen. Dieser Punkt führt bei der Modellierung von Type 2 SCD oftmals zu Verwirrungen. Jedoch lässt sich daraus ein weiteres Modellierungsprinzip ableiten:

> ***Type 2 SCD teilt die Historie automatisch auf. Anwendungen, die das DWH als Grundlage verwenden, müssen +++ keine Zeitattribute als Beschränkung fordern, um brauchbare Daten zu erhalten.***

Die Vermeidung von fremden Zeitattributen bei Modellieren von Type 2 SCD soll unnötige Komplexität und redundante Beschränkungen verringern. Weiterhin ist in manchen Fällen ein Gültigkeitsdatum bedeutungslos und der Versuch Beschränkungen auf dieses Gültigkeitsdatum zu setzten, kann falsche Analyseergebnisse liefern. Dies kann in Fällen passieren, in denen der Dimensionseintrag nicht etwas Einziges wie Mary Jones repräsentiert, sondern ein Klasse von Dingen wie zum Beispiel Dosensuppen. Im Fall von Dosensuppen repräsentiert der Dimensionseintrag beispielsweise eine bestimmte EAN[23]. An einem gewissen Zeitpunkt, beispielsweise der 07.07.2007 wird der Inhalt der Suppe geändert. Statt salzhaltigen Suppen werden ab dem Zeitpunkt Suppen ohne Salz verkauft. Dies stelle eine Veränderung der Inhaltsstoffe der Suppe dar. In Lebensmittelgeschäften kommt dies häufig vor und in diesem Fall wird eine neue EAN vergeben um das veränderte Produkt zu identifizieren. In diesem Fall ist leicht ersichtlich, dass die EAN nichts anderes als ein eindeutiger Schlüssel ist, der vom Suppenhersteller verwaltet wird.
Beim betrachten der Datenbank des Lebensmittelgeschäftes, ist keine deutliche Teilung der Historie erkennbar. Die alten Dosensuppen werden auch nach dem 07.07.2007, so lang der Vorrat reicht, verkauft. Das Verkaufsende ist sogar je nach Filiale unterschiedlich. Die neuen Dosensuppen werden ab dem 07.07.2007 auf den Verkaufsregalen erscheinen und die alten Dosensuppen allmählich ersetzten. Somit entsteht in jeder Filiale eine Übergangszeit in der beide Suppenarten verkauft werden und damit beide EAN registriert

[22] Gültigkeitsdatum - Effective Date

[23] European Article Number – meist 13-stellige, eindeutige Identifizierungsnummer für Artikel in Europa

werden. Auch sollen an der Kasse keine besonderen Verkaufsprozesse für irgendeine der beiden Arten gestartet werden. Noch mal: die Verwendung von zwei Dimensionseinträgen teilt die Faktentabelleneinträge automatisch auf. Der Benutzer braucht sich also auch nicht um die Aufteilung der Historie kümmern, vorausgesetzt er das benutzt das „Salz-Feld" in der Faktentabelle.
Der wichtige Punkt hier bei ist folgender: Obwohl ein Gültigkeitsdatum für die aktuelle Produktion in die Produktdimension eingetragen werden könnte, sollten keine Beschränkungen der Analyse auf dieses Datum gelegt werden um die Verkäufe einzuteilen. Da dies Gültigkeitsdatum der aktuellen Produktion keine Relevanz für den Verkauf hat, würde es falsche Analyseergebnisse liefern.

Mit der Tatsache, dass Type 2 SCD die Historie partitioniert, kann der neue Wert eines veränderten Attributs nicht für die alte Historie und der alte Wert nicht für die Historie des neuen Werts verwendet werden.
Am Mary-Jones-Beispiel heißt das, dass mit der Beschränkung des Familienstandes auf „Verheiratet", Mary Jones vor dem 07.07.2007 nicht auftauchen wird. In den meisten Fällen wird dies auch so erwünscht. In einigen wenigen Fällen jedoch, ist es erwünscht herauszufinden wie die Historie sich entwickelt hätte, wenn der neue Zustand die ganze Zeit der Zustand der Dimension gewesen wäre. Dazu müsste eine andere Variante der Historisierung gewählt werden.

Die eben beschriebene Ausführung von Type 2 SCD wie sie von Kimball (Kimball, 1996) spezifiziert wurde ist laut Finger unvollkommen, wenn das Quellsystem nicht weitere genauen Gültigkeitsbereiche zur Verfügung stellt (Finger, 2002, S. 18). Deswegen gibt es zwei Erweiterungen dieser Version. Einmal gibt es eine Erweiterung mit „Current Flags" und einmal eine Erweiterung mit Gültigkeitsfeldern. Die zuletzt genannte Variante wird auch wie in Abschnitt 4.3 von Microsoft® SQL Server 2005® verwendet. Dort wird jedoch statt der Schlüsselerweiterung ein Ersatzschlüssel in ein neues Feld eingefügt und der alte Schlüssel bleibt als Attribut erhalten.

Type 2 SCD mit „Current Flags"
Der Vorteil dieser Methode liegt darin, dass die Abfragen über dieses „Current Flag"-Feld perfomanter (leistungsfähiger) ist als Abfragen über die Ladezeit[24]. Jedoch ist eine zusätz-

[24] MAX(loadtime)

liche Tabellenaktualisierung nötig. Diese Version wird meist dann eingesetzt, wenn flexible Abfragen bei besonderem Fokus auf aktuelle Attributsausprägungen erfolgen sollen (Finger, 2002, S. 21) (siehe Abb. 27).

1. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Curr	Loadtime
1.01	Thompson	Joe	NY	S	M	1	31Dec1999:23:59
2.01	Webster	Lisa	LA	M	F	1	31Dec1999:23:59
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	**1**	**31Dec1999:23:59**
4.01	Smith	Tom	Miami	S	M	1	31Dec1999:23:59

2. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Curr	Loadtime
1.01	Thompson	Joe	NY	S	M	1	31Dec1999:23:59
2.01	Webster	Lisa	LA	M	F	1	31Dec1999:23:59
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	**0**	**31Dec1999:23:59**
3.02	**Jones**	**Mary**	**Dallas**	**M**	**F**	**1**	**07Jul2007:23:59**
4.01	Smith	Tom	Miami	S	M	1	31Dec1999:23:59

3. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Curr	Loadtime
1.01	Thompson	Joe	NY	S	M	1	31Dec1999:23:59
2.01	Webster	Lisa	LA	M	F	1	31Dec1999:23:59
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	**0**	**31Dec1999:23:59**
3.02	**Jones**	**Mary**	**Dallas**	**M**	**F**	**0**	**07Jul2007:23:59**
3.03	**Jones**	**Mary**	**Dallas**	**W**	**F**	**1**	**20Sep2008:23:59**
4.01	Smith	Tom	Miami	S	M	1	31Dec1999:23:59

Abb. 27: Type 2 SCD mit „Current Flags“

(Quelle: eigene Darstellung)

Type 2 SCD mit Gültigkeitsfeldern

Der große Vorteil von Gültigkeitsfeldern ist die genaue fachliche Historisierung. Alle Datensätze erhalten ein Startdatum und Endedatum der Gültigkeit. Das Endedatum ist ein Datum dass in der Realität nicht erreicht wird (z.B. das Jahr 9999). Beim Prozess der Historisierung (innerhalb des ETL-Prozesses) wird dem alten Datensatz das Datum der Änderung als Endedatum eingefügt (im Mary Jones Beispiel der Tag der Hochzeit) und dem neuen Datensatz das Datum der Änderung als Startdatum eingefügt (siehe Abb. 28). Diese Variation wird wie bereits erwähnt von dem SCD-Wizard in Microsoft® SQL Server 2005® erzeugt. Laut Finger wird diese Variante immer dann eingesetzt, wenn eine genaue Historisierung erforderlich ist und wenn operative Systeme das Erkennen des genauen Änderungszeitpunktes ermöglichen. Jedoch kann auch einfach die Ladezeit als genauer

Änderungszeitpunkt verwendet werden. Dazu ist es jedoch erforderlich, dass das DWH recht häufig vom ETL-Prozess beladen wird (Finger, 2002, S. 21).

1. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	StartDate	EndDate	Curr
1.01	Thompson	Joe	NY	S	M	31Dec1999	31Dec1999	1
2.01	Webster	Lisa	LA	M	F	31Dec1999	31Dec1999	1
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	31Dec1999	31Dec1999	**1**
4.01	Smith	Tom	Miami	S	M	31Dec1999	31Dec1999	1

2. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	StartDate	EndDate	Curr
1.01	Thompson	Joe	NY	S	M	31Dec1999	31Dec1999	1
2.01	Webster	Lisa	LA	M	F	31Dec1999	31Dec1999	1
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	31Dec1999	07Jul2007	**0**
3.02	**Jones**	**Mary**	**Dallas**	**M**	**F**	08Jul2007	31Dec1999	**1**
4.01	Smith	Tom	Miami	S	M	31Dec1999	31Dec1999	1

3. Dimensionstabelle

Cust_ID	LastName	FirstName	City	MaritalStat	Gender	StartDate	EndDate	Curr
1.01	Thompson	Joe	NY	S	M	31Dec1999	31Dec1999	1
2.01	Webster	Lisa	LA	M	F	31Dec1999	31Dec1999	1
3.01	**Jones**	**Mary**	**Dallas**	**S**	**F**	31Dec1999	31Dec1999	**0**
3.02	**Jones**	**Mary**	**Dallas**	**M**	**F**	08Jul2007	20Sep2008	**0**
3.03	**Jones**	**Mary**	**Dallas**	**W**	**F**	21Sep2008	31Dec1999	**1**
4.01	Smith	Tom	Miami	S	M	31Dec1999	31Dec1999	1

Abb. 28: Type 2 SCD mit Gültigkeitsfeldern

(Quelle: eigene Darstellung)

4.2.3 Type 3 SCD

In manchen Fällen ist es notwendig beide, den neuen und den alten Wert des veränderten Eintrags beizubehalten, um über den Stichtag der Veränderung hinaus (vorwärts und rückwärts) Analyseergebnisse zu kreieren. Diese Entscheidungsmöglichkeit stellt Type 3 SCD dar. In diesem Fall wird kein neuer Dimensionseintrag benötigt. Dem originalen Dimensionseintrag wird einfach ein neues Feld für das veränderte Attribut angefügt. Die Tabelle wird verbreitert. Im Mary-Jones-Beispiel wird eine Spalte mit dem Namen „Aktueller Familienstand“ angefügt. Sinnvollerweise sollte das originale Feld auch umbenannt werden. Im Beispiel bietet sich der Name „Ursprünglicher Familienstand“ an (siehe Abb. 29). In diesem Fall macht es außerdem Sinn ein Gültigkeitsdatum[25] anzufügen. Jedes Mal wenn der Familienstand von Mary Jones sich jetzt verändert, wird der Wert in „Aktueller Familienstand“ überschrieben und das Gültigkeitsdatum verändert. Das Feld „Ursprünglicher Familienstand“ wird dabei nie verändert. Nun ist es möglich die Historie des Dimensionseintrags von Mary Jones mit dem „Ursprünglichen Familienstand“ und dem

[25] Gültigkeitsdatum - Effective Date

„Aktuellen Familienstand“ zu verfolgen bzw. zu analysieren. Der einzige Weg um die Historie zu partitionieren, ist in diesem Fall das Gültigkeitsdatum zu benutzen, da die Faktentabelle nur auf einen Dimensionseintrag zugreift.

Type 3 SCD kann nicht benutzt werden, um die Veränderung einer Klasse von Objekten akkurat zu beschrieben wie z.B. die Dosensuppen. Im Nachhinein würde nicht mehr erkennbar sein, wann die alten Dosensuppen endgültig ausverkauft wären. Solche Situation kann vorkommen, wenn Artikeländerungen (hier: Salzfreie Dosensuppen) vorgenommen werden, jedoch mit der alten EAN weiter gearbeitet wird. Analysen, die die Historie der Verkäufe von Dosensuppen betrachten, können die beiden verschiedenen Inhalte nicht unterscheiden.

1. Dimensionstabelle

Cust_ID	LastName	FirstName	City	Current MaritalStat	Original MaritalStat	Gender	EffectiveDate
1	Thompson	Joe	NY	S	S	M	31Dec1999:23:59
2	Webster	Lisa	LA	M	M	F	31Dec1999:23:59
3	**Jones**	**Mary**	**Dallas**	**S**	**S**	**F**	**31Dec1999:23:59**
4	Smith	Tom	Miami	S	S	M	31Dec1999:23:59

2. Dimensionstabelle

Cust_ID	LastName	FirstName	City	Current MaritalStat	Original MaritalStat	Gender	EffectiveDate
1	Thompson	Joe	NY	S	S	M	31Dec1999:23:59
2	Webster	Lisa	LA	M	M	F	31Dec1999:23:59
3	**Jones**	**Mary**	**Dallas**	**M**	**S**	**F**	**07Jul2007:11:59**
4	Smith	Tom	Miami	S	S	M	31Dec1999:23:59

3. Dimensionstabelle

Cust_ID	LastName	FirstName	City	Current MaritalStat	Original MaritalStat	Gender	EffectiveDate
1	Thompson	Joe	NY	S	S	M	31Dec1999:23:59
2	Webster	Lisa	LA	M	M	F	31Dec1999:23:59
3	**Jones**	**Mary**	**Dallas**	**W**	**S**	**F**	**20Sep2008:11:59**
4	Smith	Tom	Miami	S	S	M	31Dec1999:23:59

Abb. 29: Type 3 SCD – Beispiel

(Quelle: eigene Darstellung)

Mit Type 3 SCD können lediglich die originalen und die aktuellen Zustände gehandhabt werden. Zwischenzustände gehen verloren. Sollte jedoch eine akribische Historie von Nöten sein, dann sollte Type 2 SCD benutzt werden. Als noch bessere Lösung schlägt Kimball eine Mischung aus Type 2 SCD und Type 3 SCD vor, die jedoch Komplexität des DWH steigern würde. Außerdem gibt es noch erweiterte Formen von SCD.

4.2.4 Weitere Formen von SCD

Weitere Formen und vor allem Mischformen erweisen sich vor allem für betriebswirtschaftlich orientierte Anwender von BI-Systemen als sehr komplex. In dieser Arbeit möchte ich nur ein Beispiel für eine Erweiterung von Type 3 SCD geben und weiterhin auf die Fachliteratur (Kimball et al., 2000) verweisen.

Serie von Type 3 SCD

Im Falle von Slowly Changing Dimensions die sich in einem vorhersagbaren Rhythmus ändern (z.B. jährlich) und alle Zwischenstände benötigt werden, kann diese Variation verwendet werden. Zur Veranschaulichung dient in Beispiel, indem nach jedem Geschäftsjahr die Produktkategorien neu geordnet werden, aber eine Analyse über die kompletten Jahre möglich sein soll. Diese Situation kann am bestem mit einer Serie Type 3 SCD abgebildet werden. Jede Dimensionszeile der Produkte erhält eine „aktuelle Produktkategorie" Attribut (die wie in Type 3 SCD überschrieben wird) und je ein Attribut für jede Änderung (z.B. Produktkategorie 2003, Produktkategorie 2004, ...) (siehe Abb. 30).

Dimensionstabelle

	Prod:ID	ProdKat	Name	Eff.Date	LoadTime
1.	0001	Eisenwaren	Schraube 3x40	31Dec2000	31Dec2000:23:59
	0002	Eisenwaren	Schraube 6x80	31Dec2000	31Dec2000:23:59
	0003	**Eisenwaren**	**Dachpappstift**	**31Dec2000**	**31Dec2000:23:59**
	0004	Eisenwaren	Nagel 80	31Dec2000	31Dec2000:23:59

Dimensionstabelle

	Prod:ID	ProdKat2001	ProdKat2002	Name	Eff.Date	LoadTime
1.	0001	Eisenwaren	Eisenwaren	Schraube 3x40	31Dec2000	31Dec2000:23:59
	0002	Eisenwaren	Eisenwaren	Schraube 6x80	31Dec2000	31Dec2000:23:59
	0003	**Eisenwaren**	**Dachdeckerzubehör**	**Dachpappstifft**	**31Dec2001**	**31Dec2001:23:59**
	0004	Eisenwaren	Eisenwaren	Nagel 80	31Dec2000	31Dec2000:23:59

Dimensionstabelle

	Prod:ID	ProdKat2001	ProdKat2002	ProdKat2003	Name	Eff.Date	LoadTime
1.	0001	Eisenwaren	Eisenwaren	Eisenwaren	Schraube 3x40	31Dec2000	31Dec2000:23:59
	0002	Eisenwaren	Eisenwaren	Eisenwaren	Schraube 6x80	31Dec2000	31Dec2000:23:59
	0003	**Eisenwaren**	**Dachdeckerzubehör**	**Befestigungsmaterial**	**Dachpappstifft**	**31Dec2002**	**31Dec2002:23:59**
	0004	Eisenwaren	Eisenwaren	Eisenwaren	Nagel 80	31Dec2000	31Dec2000:23:59

Abb. 30: Serie von Type 3 SCD

(Quelle: eigene Darstellung)

Diese scheinbar unkomplizierte Technik ist jedoch unbrauchbar für unvorhersehbare Änderungen. Beispielsweise verändern sich Kundenattribute einzigartig. Eine Serie von Type 3 SCD kann nicht verwendet werden um die alten Zustände zu historisieren (z.B. alter_Zustand-1, alter_Zustand-2, ...) wenn die Zustände sich unregelmäßig ändern. Jedes

Attribut stünde dann in Verbindung mit einem einzigartigen Zeitpunkt und hätte dann nur eine einzige Zuordnung pro Dimensionszeile.

Die Verwendung von gemischten oder erweiterten SCD Formen kann kompliziert und verwirrend sein. Außerdem sollte stets zwischen Komplexität und Flexibilität abgewogen werden, da dies konträre Ziele sind. Weiterhin wächst die Anzahl der Fehlerquellen bei steigender Komplexität. Deshalb sollte von unnötiger Komplexität abgesehen werden (Kimball et al., 2000).

Als Nachtrag und zur Klärung muss hinzugefügt werden, dass ich bei meinen Recherchen im World Wide Web auf Type 0 und Type 4 SCD gestoßen bin[26]. In der Praxis kommen zwar vor, sind jedoch nicht empfehlenswert. Typ 0 SCD ist der Zustand wenn kein Aufwand betrieben wird um SCD zu modellieren – also eine nicht optimierte Datenhistorie. Type 4 SCD ist im eigentlichen Sinnen auch keine Optimierung sondern ein Aufblähen des DWH. Dabei wird, wenn sich eine Dimension verändert, die komplette Dimensionstabelle mit der Änderung an die Dimensionstabelle angefügt (siehe Abb. 31). Im Laufe der Zeit kann Dimensionstabelle sehr groß werden, was zu unnötigen Datenmengen führt.

Dimensionstabelle

	Cust_ID	LastName	FirstName	City	MaritalStat	Gender	Loadtime
1.	1	Thompson	Joe	NY	S	M	31Dec1999:23:59
	2	Webster	Lisa	LA	M	F	31Dec1999:23:59
	3	**Jones**	**Mary**	**Dallas**	**S**	**F**	**31Dec1999:23:59**
	4	Smith	Tom	Miami	S	M	31Dec1999:23:59
	1	Thompson	Joe	NY	S	M	**07Jul2007:11:59**
	2	Webster	Lisa	LA	M	F	**07Jul2007:11:59**
	3	**Jones**	**Mary**	**Dallas**	**M**	**F**	**07Jul2007:11:59**
	4	Smith	Tom	Miami	S	M	**07Jul2007:11:59**
	1	Thompson	Joe	NY	S	M	**20Sep2008:11:59**
	2	Webster	Lisa	LA	M	F	**20Sep2008:11:59**
	3	**Jones**	**Mary**	**Dallas**	**W**	**F**	**20Sep2008:11:59**

Abb. 31: Type 4 SCD – Beispiel

(Quelle: eigene Darstellung)

4.3 Slowly Changing Dimensions mit Microsoft® SQL Server 2005®

Während meiner Untersuchung von Literatur und Internetseiten zu diesem Thema konnte ich feststellen, dass die konkrete Umsetzung von Slowly Changing Dimensions ist bei

[26] http://etl-tools.info/en/scd.html (01.10.2008)

DWH-Entwicklern nicht sehr beliebt ist. Deswegen bieten Hersteller von BI-Systemen Hilfsmittel an um diese Hürde zu meistern. So bietet Microsoft seit der 2005er Version von SQL Server bietet Unterstützung von zumindest Type 1 und 2 SCD in Form von einem Wizard an.

In diesem Abschnitt wird ein Beispiel zum Umgang mit SCD mithilfe von Microsoft® SQL Server 2005® gezeigt. Da SCD bereits im ETL-Prozess erstellt werden müssen, wird dazu SQL Server Integration Services® (SSIS) verwendet. Zur Erstellung dieser Demo der Darstellung von SCD mit Microsoft® SQL Server 2005® wurde (Stropek, 2005) und vor allem (Barclay et al., 2005) verwendet.

Einrichtung des Szenarios

Um das Beispiel nur in einer kleinen Umgebung zu zeigen, wird die Datenbank SCDTEST kreiert. In diese Datenbank wird nun eine Quelltabelle (MySourceDimCustomer) mit den Dimensionen CustomersKey (Kundennummer), FirstName, LastName, BirthDate, MaritalStatus, Gender und NumberCarsOwned und eine Zieltabelle (MyTargetDimCustomer) mit den Dimensionen CustomerWK (Primärschlüssel), CustomersKey, FirstName, LastName, BirthDate, MaritalStatus, Gender, NumberCarsOwned, StartDate und EndDate erzeugt. Vorher wird jedoch zuerst geprüft ob die Tabellen schon existieren und im positiven Fall gelöscht um eine neue Umgebung vorliegen zu haben. Um in die Quelltabelle Daten zu laden, werden die 100 ersten Datensätze aus der Tabelle DimCustomer aus Microsofts Beispieldatenbank „AdventureWorksDW" kopiert. Der dazu erforderliche SQL-Code, der im SQL Server Management Studio® (SSMS) ausgeführt werden muss, sieht folgendermaßen aus:

```
USE SCDTEST
GO

SET NOCOUNT ON

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[MyTargetDimcustomer]') AND OBJECTPROPERTY(id,
N'IsUserTable') = 1)
DROP TABLE [MyTargetDimcustomer]

CREATE TABLE MyTargetDimcustomer(
      CustomerWK INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
      CustomerKey INT NOT NULL,
      FirstName NVARCHAR(50)  NULL,
      LastName NVARCHAR(50)  NULL,
      BirthDate DATETIME NULL,
      MaritalStatus NCHAR(1)  NULL,
      Gender NVARCHAR(1)  NULL,
      NumberCarsOwned TINYINT NULL,
      StartDate DATETIME NULL,
      EndDate DATETIME NULL
```

```
)

IF      EXISTS    (SELECT    *    FROM    dbo.sysobjects    WHERE    id    =
OBJECT_ID(N'[MySourceDimCustomer]')       AND       OBJECTPROPERTY(id,
N'IsUserTable') = 1)
DROP TABLE [MySourceDimCustomer]

CREATE TABLE [MySourceDimCustomer](
      CustomerKey INT NOT NULL PRIMARY KEY,
      FirstName NVARCHAR(50)NULL,
      LastName NVARCHAR(50) NULL,
      BirthDate DATETIME NULL,
      MaritalStatus nchar(1) NULL,
      Gender NVARCHAR(1) NULL,
      NumberCarsOwned tinyINT NULL
)

-- insert some sample data
INSERT INTO MySourceDimCustomer
SELECT TOP 100
      CustomerKey
      ,FirstName
      ,LastName
      ,BirthDate
      ,MaritalStatus
      ,Gender
      ,NumberCarsOwned
FROM AdventureWorksDW..DimCustomer
```

Als nächstes wird ein Datenfluss-Task erzeugt in dem man das Symbol in den SSIS Flow

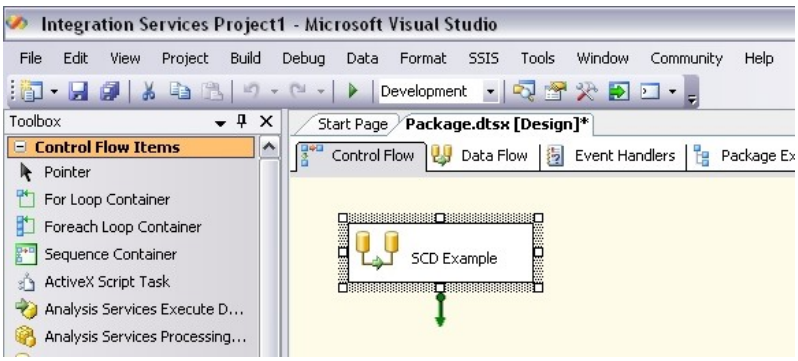

Abb. 32: Flusskontrolle in SSIS

(Quelle: eigener Screenshot)

Control zieht (siehe Abb. 32). Beim Doppelclick auf das entstandene Rechteck, gelangt man in den SSIS Data Flow. Dort wird zunächst eine Quelle hinzugefügt aus der die Daten extrahiert, transformiert und geladen werden sollen. Durch den Connection Manager kann nun die Verbindung zur "SCDTEST" Datenbank hergestellt werden. Außerdem muss in dem Quelldatenbanksymbol unter "Edit" die Quelldimensionstabelle "MySourceDimCustomer" angesprochen werden. Diese wird in diesem Fall mit dem Data Access Mode "SQL Command" gemacht. Der SQL Befehl sieht wie folgt aus:

```
SELECT CustomerKey, FirstName, LastName, BirthDate, MaritalStatus
,Gender, NumberCarsOwned
FROM MySourceDimCustomer
```

Dann kann das SCD Symbol dem Data Flow hinzugefügt werden. Anschließend muss die Datenquelle mit dem SCD-Symbol verbunden werden. Mittels Rechtscklick gelangt man nun unter "Edit" zum Slowly Changing Dimensions Wizard. Dieser fürt den Benutzer jetzt Schritt für Schritt durch alle nötigen Eistellungen und Optionen. Hier werden Screenshots von den elementaren Einstllungsmöglichkeiten gezeigt.

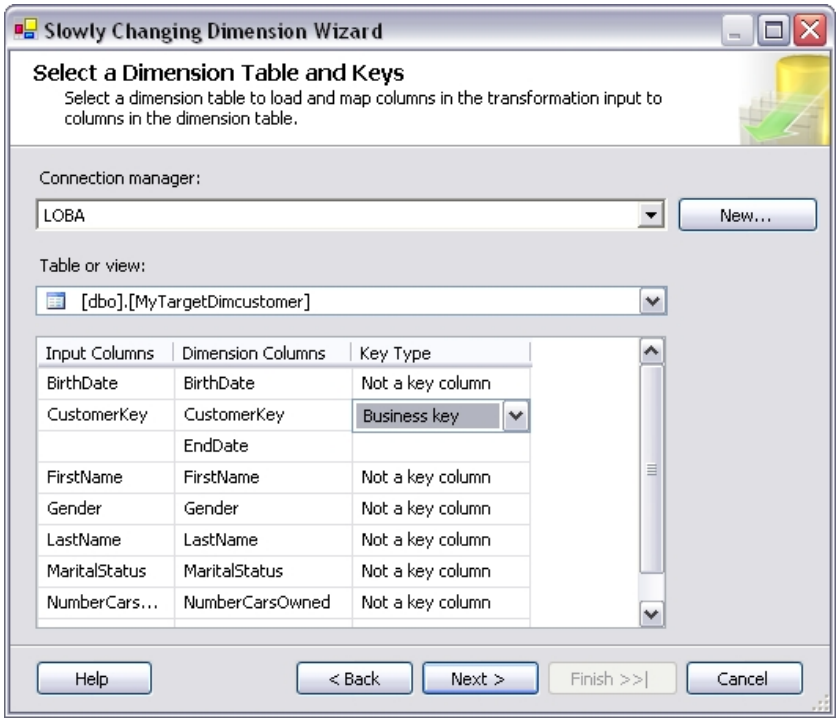

Abb. 33: SCD Wizard – Fenster 2

(Quelle: eigener Screenshot)

In Abb. 33 wird die Zieldimensionstabelle ausgewählt in welche die Daten hineinkopiert werden sollen. Außerdem muss ein e Dimension als Business Key ausgewählt werden.

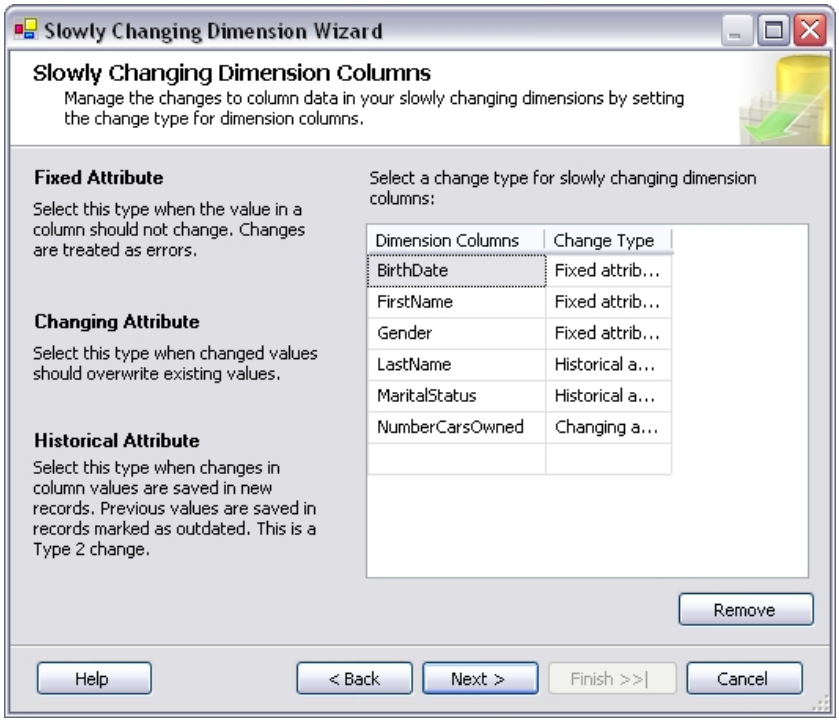

Abb. 34: SCD Wizard - Fenster 3

(Quelle: eigener Screenshot)

In Abb. 34 werden die Dimensionsausprägungen nach Veränderlichkeit definiert. BirthDate, FirstName und Gender verändern sich nicht und werden als "Fixed attribute" deklariert. Dadurch sind diese Attribute auch vor fehlerhaften Änderungen geschützt. LastName und MaritalStatus verändern sich im weiteren Verlauf des Beispiels. Diese Attribute sollen historisiert werden und werden deswegen als "Historical attribute" deklariert. Bei Änderung dieses Attributes wird ein Prozessausgelöst, der die Dimension nach Type 2 SCD historisiert. NumberCarsOwned ist ein "Changing attribute" und wird im Änderungsfall einfach überschrieben.

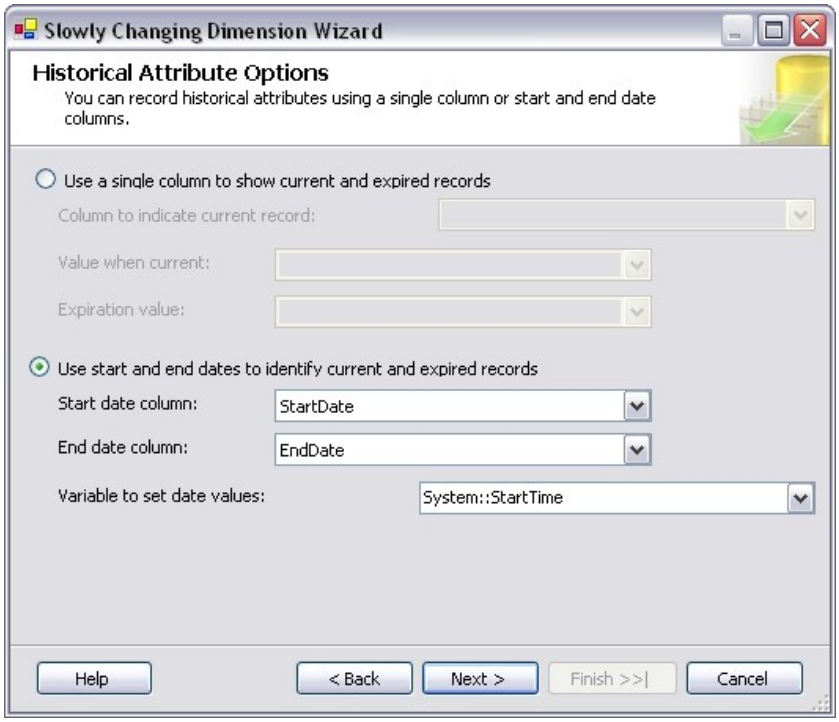

Abb. 35: SCD Wizard - Fenster 5

(Quelle: eigener Screenshot)

In dem Fenster das Abb. 35 zeigt, kann eingestellt werden ob eine einzige Spalte die Historisierung anzeigt, oder ob eine Start Datum-Spalte und ein Ende Datum-Spalte angelegt werden sollen um die Historisierung zu realisieren. Im Beispiel wird ein Start und Ende Datum verwendet.

Sobald der Wizard abgeschlossen wird, erzeugt SSIS drei Data Flow Pfade in denen er die Dimensinoen je nach angegebenem Attribut behandelt (vgl. Abb. 36). Der linke Pfad wird für Dimensionsattribute mit dem Change Type[27] "Changing attribute" erstellt. Die so deklarierten Dimensionsattribute werden einfach aktualisiert und nicht historisiert. Der mittlere Pfad wird für neue Einträge erstellt. Neu hinzugefügte Daten müssen nicht historisiert werden, jedoch mit einem Datum versehen werden, ab wann sie gültig sind. Der rechte Pfand ist für Dimensionsattribute die den Change Type "Historical attribute" haben. Sind Dimensionen mit diesem Change Type betroffen, Wird zunächst das Ende von der Gültigkeit dieses Attributs ermittelt und anschließend beim alten Attribut eingetragen. Auch hierbei wird das Startdatum ermitteln und beim neuen Attribut eingetragen. Nun kann der Data Flow ausgeführt. Mit "F5" startet man das Debugging. In aktuellen Beispiel mit der SCDTEST-Datenbank werden beim ausführen alle 100

[27] siehe Abb. 34

Datensätze in die Zieltabelle "MyTargetDimCustomer" eingetragen, da diese vorher leer war. SSIS zeigt im Debug-Modus auch an, dass "100 rows" den mitteren Datenfluss-Pfad durchschritten haben (siehe Abb. 36). Die Zieltabelle ist nun identisch mit der Quelltabelle.

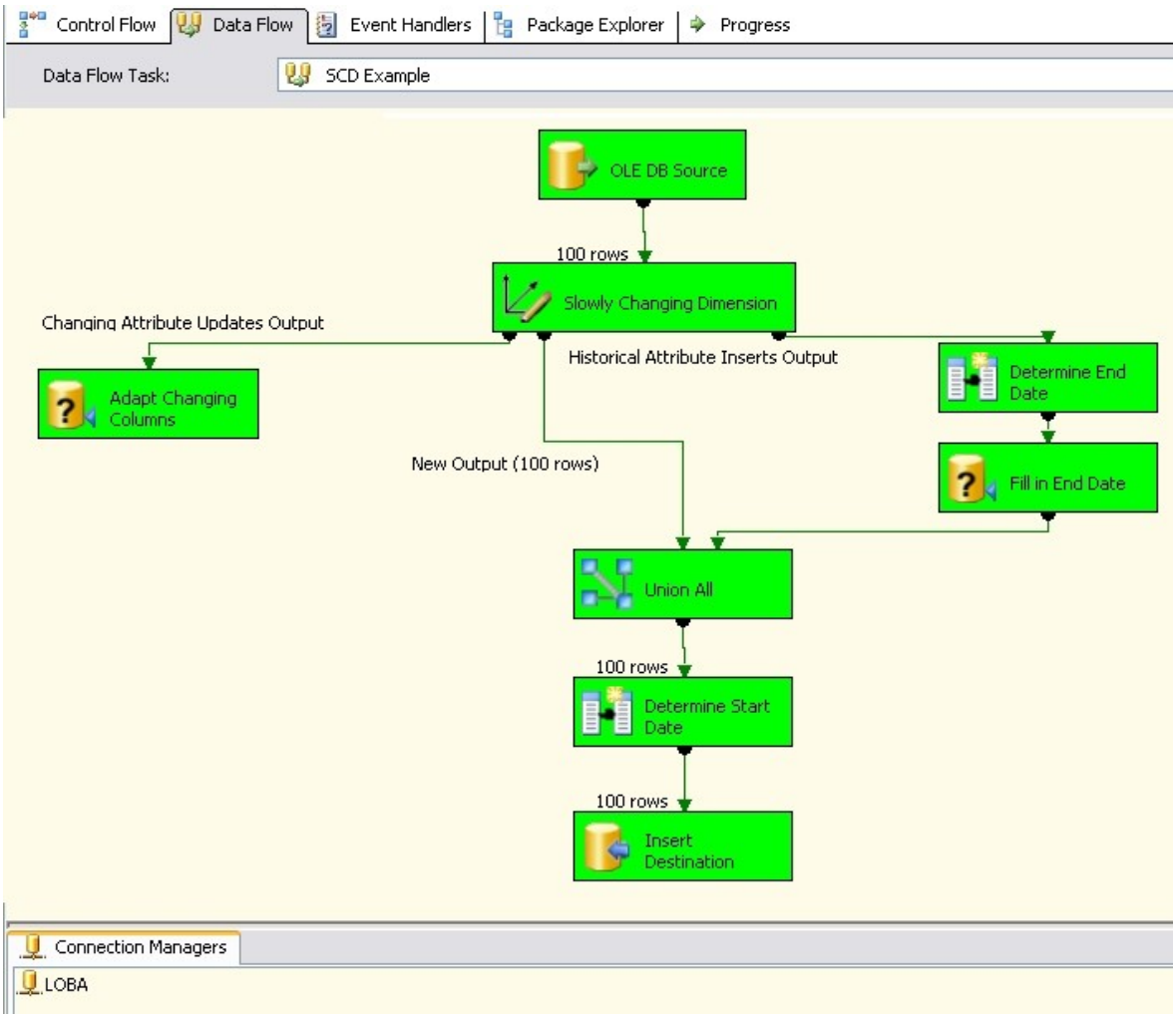

Abb. 36: SCD-Beispiel in der Datenfluss-Ansicht in SSIS

(Quelle: eigener Screenshot)

Veränderung von Dimensionen:

Nun sollen einige Änderungen vorgenommen werden, die dann historisiert werden. Zuerst verkauft Elizabeth Johnson zwei ihrer Autos. Da das Dimensionsattribut NumbersCarsOwned als "Changing attribute" deklariert wurde, wird die Anzahl der Autos einfach mit "2" überschrieben. Dies ist ein einfacher Type 1 SCD-Fall.

Als nächtes heiratet Christy Zhu und verändert damit ihren Familienstand und ihren Nachnamen. Beides sind "Hisorical attributes" und werden nicht überschrieben sondern historisiert.

Dann wird ein neuer Datensatz eingefügt. Der Kunde John Doe, der männlich und verheiratet ist, hat zwei Autos.

Zuletzt wird bei einem Kunden der Vornahme geändert. Da damit jedoch ein "Fixed attribute" betroffen ist, wird keine Änderung in der Zieltabelle vorgenommen.

Der zu den Änderungen gehörige SQL-Code wird in SSMS ausgeführt. Von den Änderungen ist zunächst nur die Quelltabelle betroffen. Dort werden die Originalen Daten einfach überschrieben, denn die Quelltabelle sysmbolisiert im Beispiel ein operatives System.

```
USE SCDTEST
GO

SET NOCOUNT ON

-- type 1 change (Changing attribute)
-- Elizabeth Johnson sells two of her cars
UPDATE MySourceDimCustomer
SET
      NumberCarsOwned = 2
WHERE CustomerKey = 11004

-- type 2 change (Historical attribute)
-- Christy Zhu gets married
UPDATE MySourceDimCustomer
SET
      LastName = 'Smith',
      MaritalStatus = 'M'
WHERE CustomerKey = 11003

-- New record
INSERT INTO MySourceDimCustomer
VALUES(
      11100
      ,'John'
      ,'Doe'
      ,'12-Aug-1974'
      ,'M'
      ,'M'
      ,2
)

-- change to a fixed attribute
-- Ian Jenkins changes his name to John
-- will not be changed by the transform
UPDATE MySourceDimCustomer
SET
      FirstName = 'John'
WHERE CustomerKey = 11013
```

Nachdem der SQL-Code ausgeführt wurde, müssen die Änderungen nun in die Zieltabelle, die ein Data Warehouse sysmbolisiert, übertragen werden. Jedoch darf hier nichts überschrieben werden, sondern muss historisiert werden. Dazu wird der eben erstellte Data Flow (siehe Abb. 36) im SSIS nochmals ausgeführt. Im Debug-Modus wird

angezeigt, dass jeweils eine Reihe im linken "Changing Attribute"-Pfad, eine im mitterlern "New Output"-Pfad und eine im rechten "Historical Attribute"-Pfad verarbeitet wurden.

Ergebnisse:

Die Ergebnisse sollen nun angezeigt werden. Dazu wird im SSMS der folgende SQL-Code ausgeführt um nur die betroffenen Datensätze angezeigt zubekommen.

```
USE SCDTEST
GO

SELECT *
FROM MyTargetDimcustomer
WHERE CustomerKey in (11003, 11004, 11100, 11013)
ORDER BY CustomerKey
```

Als Ergebniss bekommt man den Ausschnitt aus der Zieltabelle, der das DWH symbolisiert (siehe Abb. 37).

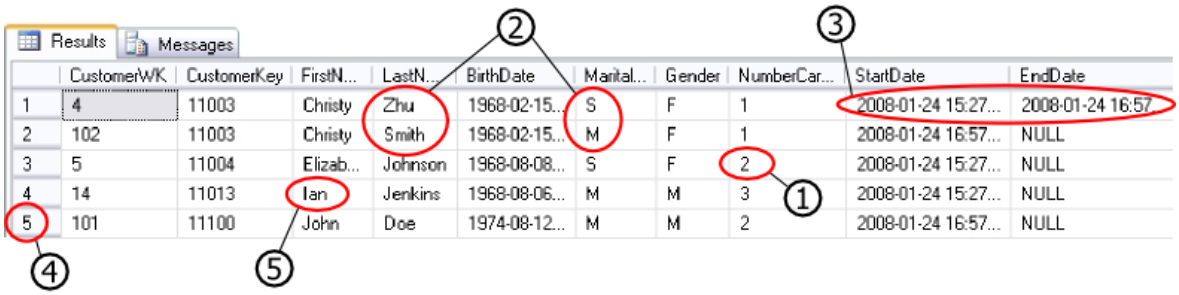

Abb. 37: Ergebnisse des SCD-Beispiels

(Quelle: eigener Screenshot)

Zuerst ist auffällig, dass die Zieltabelle eine Spallte mehr hat (CustomerWK) als die Quelltabelle. Dies ist darauf zurückzuführen, dass der Primary Key der Tabelle eindeutig sein muss. Bei SCD kommt es vor, dass Einträge verdoppelt werden und diesen Kriterium dann nicht länger erfüllt ist. In diesem Beispiel kommt der CustomerKey "11003" zweimal vor und ist damit nicht mehr eindeutig.

Nun zu den Änderungen die beim Aktualisieren der Zieltabelle gemacht wurden und dort historisiert werden sollten. Elizabeth Johnson hat jetzt statt "4" nur noch "2" Autos. Da die Dimension NumberCarsOwned als "Changing" deklariert wurde, kann die Anzahl der Autos einfach überschrieben werden. (Type 1 SCD)(siehe Abb. 37 Kreis 1).

Als nächstes wurde der Familienstand und der Nachname von Christy Zhu geändert. Da der alte Zustand für Analysezwecke eerhalten bleiben soll, wird nun mit Type 2 SCD historisiert. Der alte Eintrag bekommt ein EndDate eintrag und ist somit nicht mehr gültig. Der neue Eintrag bekommt die neuen Werte und ein StartDate. Dieses StartDate ist

identisch mit dem EndDate des alten Eintrag, sodass keine Lücke entsteht (Type 2 SCD) (siehe Abb. 37 Kreis 2 und 3).
Das StartDate des neuen Kunden John Doe ist der Zeitpunkt in der er der Zieltabelle hinzugefügt wurde. Bei Analysen die über ältere Zeitabschnitte gehen, wird dieser Kunde nicht auftauchen, da er in Wirklichkeit ja auch nicht exestierte (siehe Abb. 37 Kreis 4).
Die letzte Änderung wurde wie bereits vorhergesagt nicht übernommen, da der Vorname ein "Fixed attribute" ist und nicht verändert werden kann. Das Update der Quelltabelle wird vom SCD-Data Flow ignoriert (siehe Abb. 37 Kreis 5).

Somit ist an diesem einfachen Beispiel gezeigt worden, wie Slowly Changing Dimensions mit Microsoft® SQL Server 2005® kreiert werden können. Im SCD Wizard gibt es noch weitere Optionen, die nützlich sein können. So kann für "Fixed attributes" eine Sicherung eingebaut werden, bei der der komplette Transformationsprozess scheitert, falls ein "Fixed attribute" geändert werden soll. Ohne diese Funktion könnte würde die änderung des Attributes einfach vernachlässigt werden. Somit stellt diese Funktion sicher, dass die Änderung nicht versehntlich vernachlässigt wird. Weiterhin gibt es eine Option für "Changing attributes". Es kann aus gewählt ob nur die aktuelle Version überschrieben wird, oder aber auch alle alten, mit Type 2 SCD historisierte Versionen mit der aktuellen Änderung überschrieben (Type 1 SCD) werden sollen (Myers, 2007). Außerdem gibt es noch die Option "inferred members" (abgleitete Elemente) zu behandeln. "Inferred members" sind Dimensionselemente die noch nicht geladen sind, wenn die Faktendaten, die mit ihnen verbunden sind, bereit sind in die Faktentabelle eingefügt zu werden. Sobald die Diemensionselemente bereit stehen, können die "inferred members" aktualisiert werden. Ohne diese Option würden Einträge mit "NULL" im DWH entstehen (Jalali, 2007).
In diesem Kapitel wurde gezeigt was Slowly Changing Dimensions sind und wie sie implementiert werden. Als Beispiel wurde dazu Microsoft® SQL Server 2005® verwendet.
Die Diskussion, ob die Modellierung von SCD letztendlich eine Optimierung der Datenhistorie im DWH darstellt oder ob es eine Voraussetzung ist, müsste noch geführt werden. Meiner Meinung nach, kommt es zuerst auf den Kontext, in dem das BI-System verwendet wird, an und wie sensible etwaige Auswertung sind. Andererseits kommt es auch drauf an welche Dimensionen sich verändern und wie häufig Änderungen geschehen. Letzten Endes sind Datenhistorien ohne die Modellierung von SCD verzehrte Historien

und eine Suboptimale Lösung. Das Einrichten und Modellieren von BI-Systemen ist nicht zuletzt auch finanziell ein aufwändiges Ereignis in jedem Unternehmen. Da sollte der vergleichsweise geringe zusätzliche Aufwand für die Modellierung von SCD nicht vernachlässigt werden.

5 Zusammenfassung und Fazit

Ziel dieser Arbeit war es das nötige Wissen zur Sicherstellung einer korrekten Datenhistorie in einem Business Intelligence System zu liefern. Dabei wurde vor allem auf das Phänomen von sich langsam verändernden Dimensionen (Slowly Changing Dimensions) eingegangen, das eine Optimierung der Datenhistorie darstellt.

Dazu mussten jedoch zuvor die Grundlagen aufbereitet werden, wie BI-Systeme die Daten bereitstellen. Zuerst wurde zwischen einem analyseorientiertem BI-System und einem transaktionsorientiertem ERP-System unterschieden. Beide sind für das Managen und die Führung von größeren Unternehmen nötig. Dabei bauen BI-Systeme auf den Daten von ERP-Systemen auf und sind somit nicht unabhängig von einander.

Der in der Wirtschaft nicht ganz eindeutige Begriff „Business Intelligence“ wurde nach Kemper et al., 2006 als integrierter, unternehmensspezifischer IT-Gesamtansatz zur betrieblichen Entscheidungsunterstützung definiert. Dazu zählen im Einzelnen der ETL-Prozess, das Data Warehousing und das Online Analytic Processing. Weiterhin zählen dazu noch das Data Mining, das Reporting und die Präsentation (Dashboard und Portale), die jedoch in einem anderen Kapitel aufgeführt wurden, da sie die korrekte Historie benötigen bzw. verwenden und nicht bereitstellen, wie die ersten drei genannten Komponenten von BI.

Schließlich wurden in dieser Arbeit die Grundlagen zu Datenhistorisierung geliefert. Dazu musste zwischen Backup, Archivierung und der Historisierung unterschieden werden. Während Backup und Archivierung nur die Sicherung von Daten für zukünftig Zwecke darstellen, sind Historisierungen dafür da, die Veränderungen und den Verlauf von Datenbeständen darzustellen.

In Datenbanksysteme gibt es die Möglichkeit Slowly Changing Dimensions gezielt zu speichern. Dies geschieht im ETL-Prozess. Bevor die Daten in die Zieldatenbank geladen werden, werden sie speziell transformiert.

Type 1 SCD stellt den einfach Fall des Updates dar. Der alte Wert wird lediglich mit dem neuen Wert überschrieben. Somit wird nur der aktuelle Stand eines DWH dargestellt.

Type 3 SCD wird dagegen am seltensten verwendet. Hierbei wird eine weitere Spalte hinzugefügt, die den neuen Zustand anzeigt. In der alten spalte bleibt stets der originale Zustand gespeichert. Im Gegensatz zu Type 2 SCD kann hier *nur* der originale und der aktu-

elle Zustand angezeigt werden. Zwischenstände sind nicht möglich Diese Technik wird am seltensten verwendet (Kimball et al., 2000).

Bei Type 2 SCD wird eine Historisierung durchgeführt. Dazu wird der alte Datensatz kopiert und mit der Änderung eingefügt. Zur eindeutigen Identifikation wird ein Ersatzschlüssel eingefügt. Der alte Schlüssel bleibt den Datensätzen als Attribut erhalten. Außerdem erhalten der alte und der neue Datensatz ein Start und Endedatum. So können beliebig viele Zustände über die Zeit gespeichert werden. Type 2 SDC ist die am häufigsten verwendete Methode.

Weiterhin können zur Historisierung von Slowly Changing Dimensions Mischformen bzw. Erweiterungen dieser drei Arten verwendet werden.

Als Fazit kann einerseits angeführt werden, dass BI-Systeme zur Bereitstellung einer Datenhistorie dienen, die vor allem für die Analyse verwendet wird. Andererseits steht fest, dass BI-Systeme bei denen Slowly Changing Dimensions berücksichtigt und modelliert werden eine deutlich korrektere Datenhistorie aufweisen als BI-Systeme bei denen die sich langsam verändernden Dimensionen einfach überschrieben werden.

Literaturverzeichnis

Anandarajan et al., 2004: Anandarajan, M.; Anandarajan, A.; Srinivasan, C.A.: "Business Intelligence Techniques", Springer Verlag,Berlin, Heidelberg, 2004.

Barclay et al., 2005: Barclay, Nick; Thomson, Jamie: "SSIS: SCD Wizard demo", http://blogs.conchango.com/, 06.06.2005.

Bauer et al., 2004: Günzel, Holger; Bauer, Andreas (Hrsg.): "Data Warehouse Systeme - Architektur, Entwicklung, Anwendung", dpunkt.verlag GmbH, Heidelberg, 2004.

Bäumer, 2006: Bäumer, Jörg: "Real-Time Business Intelligence", Diplomarbeit an der Universität Koblenz-Landau, 2006.

Bissantz et al., 2000: Bissantz, Nicolas; Hagedorn, Jürgen; Mertens, Peter: "Data Mining", in: Muksch, Harry, Behme, Wolfgang (Hrsg.): Das Data Warehouse-Konzept, 4. Aufl., Gabler Verlag, Wiesbaden, 2000.

Chamoni et al., 1998: Chamoni, Peter; Gluchowski, Peter: "Analytische Informationssysteme - Einordnung und Überblick", in: Chamoni, Peter; Gluchowski, Peter (Hrsg.): Analytische Informationssysteme, Springer Verlag, Berlin, Heidelberg, New York, 1998.

Chamoni et al., 2000: Chamoni, Peter; Gluchowski, Peter: "On-Line Analytical Processing (OLAP)", in: Muksch, Harry, Behme, Wolfgang (Hrsg.): Das Data Warehouse-Konzept, 4. Aufl., Gabler Verlag, Wiesbaden, 2000.

Chamoni et al., 2004: Chamoni, Peter; Gluchowski, Peter: "Integrationstrends bei Business-Intelligence-Systemen", in: Wirtschaftsinformatik, 46/2004.

Chamoni, 1998: Chamoni, Peter: "Entwicklungslinien und Architekturkonzepte des Online Analytical Processing", in: Chamoni, Peter; Gluchowski, Peter (Hrsg.): Analytische Informationssysteme, Springer Verlag, Berlin, Heidelberg, New York, 1998.

Fayyad et al., 1996: Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic: "From Data Mining to Knowledge Discovery in Databases", in: AI Magazine, 17(3) 1996.

Few, 2006: Few, Stephen: "Information Dashboard Design", O'Reilly Media, Inc., 2006.

Finger, 2002: Finger, Ralf: "Historisierungskonzepte - Vortrag zu Data Warehouses und Data Marts", Information Works, 2002.

Gauer, 2006: Gauer, Werner: "Performance Measurement von Geschäftsprozessen", Diplomarbeit an der Rheinischen Fachhochschule Köln, 2006.

Gluchowski et al., 2006: Gluchowski, Peter; Kemper, Hans-Georg: "Quo Vadis Business Intelligence?", in: BI-Spektrum, 01/2006.

Gluchowski et al., 2008: Gluchowiski, Peter; Gabriel, Roland; Dittmar, Carsten: "Management Support Systeme und Business Intelligence, 2. Auflage", Springer Verlag, Berlin, Heidelberg, 2008.

Gluchowski, 2001: Gluchowski, Peter: "Business Intelligence - Konzepte, Technologien und Einsatzbereiche", in: HMD - Praxis der Wirtschaftsinformatik, 222/2001.

Groffmann, 1997: Groffmann, Hans-Dieter: "Das Data Warehouse Konzept", in: HMD - Praxis der Wirtschaftsinformatik, 195/1997.

Gronau, 2004: Gronau, Norbert: "Enterprise Resorce Planning und Supply Chain Management", Oldenbourg Verlag, München, Wien, 2004.

Hahne, 1998: Hahne, Michael: "Logistische Modelierung für das Data Warehouse", in: Chamoni, Peter; Gluchowski, Peter (Hrsg.): Analytische Informationssysteme, Springer Verlag, Berlin, Heidelberg, New York, 1998.

Herbst, 1997: Herbst, Axel: "Anwendungsorientiertes DB-Archivieren", Springer Verlag, Berlin, Heidelberg, New York, 1997.

Höhn, 2000: Höhn, Reinhard: "Der Data Warehouse Spezialist - Entwurf, Methoden und Umsetzung eines DWH", Addison-Wesley Verlag, München, Boston, et al., 2000.

Inmon, 1996: Inmon, W.H.: "Building the Data Warehouse - Second Edition", Wiley Computer Publishing, New York, u.a., 1996.

Jalali, 2007: Jalali, Amin: "Late-arriving dimension scenario (inferred members)", http://bisolutions.blogspot.com/, 26.09.2007.

Jarke et al., 1999: Quix, Christoph; Jarke, Matthias: "Data Warehouse Practice: An Overview", in: Jarke, Matthias; Lenzerini; Vassiliou; Vassiliadis: Fundamentals of Data Warehouses, Springer Verlag, Berlin, Heidelberg, New York, 1999.

Kaiser, 2006: Kaiser, Bernd-Ulrich: "Business Intelligence - an der Technik scheitern wir heute nicht mehr", in: HMD, 247/2006.

Kamp, 2006: Kamp, Stefan: "Implementierungsaufwand Siebel-basierter Data Warehouse Systeme", Diplomarbeit an der Universität Koblenz-Landau, 2006.

Kemper et al., 1998: Kemper, Hans Georg; Finger, Ralf: "Datentransformation im Data Warehouse", in: Chamoni Peter; Gluchowski, Peter (Hrsg.): Analytische Informationssysteme, Springer Verlag,Berlin, Heidelberg, 1998.

Kemper et al., 2004: Kemper, Alfons; Eickler, André: "Datenbanksysteme - Eine Einführung, 5. Auflage", Oldenbourg Verlag, München, Wien, 2004.

Kemper et al., 2006: Kemper, Hans-Georg; Walid, Mehanna; Unger, Carsten: "Business Intelligence - Grundlagen und praktische Anwendungen, 2.Auflage", Vieweg Verlag, Wiesbaden, 2006.

Kimball et al., 2000: Kimball, Ralph; Ross,Margy: "Slowly Changing Dimensions Are Not Always as Easy as 1, 2, 3", http://www.intelligententerprise.com, 2000.

Kimball, 1996: Kimball, Ralph: "The Data Warehouse Toolkit", Wiley Computer Publishing, New York, et.al., 1996.

Lehner, 2003: Lehner, Wolfgang: "Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden", dpunkt.verlag GmbH, Heidelberg, 2003.

Martin, 1998: Martin,Wolfgang: "Data Warehousing - Data Mining - OLAP", International Thomson Publishing, Bonn, et al., 1998.

Mertens, 2002: Mertens, Peter: "Business Intelligence - Ein Überblick", Arbeitspapier an der Arbeitspapier der Universität Erlangen-Nürnberg, 2/2002.

Mucksch et al., 2000: Muksch, Harry, Behme, Wolfgang: "Das Warehouse-Konzept als Basis einer Informationslogistik", in: Muksch, Harry, Behme, Wolfgang (Hrsg.): Das Data Warehouse-Konzept, 4. Aufl., Gabler Verlag, Wiesbaden, 2000.

Myers, 2007: Myers, Peter; Owens, Zach Skyles: "Implementing Slowly Changing Dimensions in the Data Flow", http://channel9.msdn.com/, 13.12.2007.

Schaarschmidt, 2001: Schaarschmidt, Ralf: "Archivierung in Datenbanksystemen", Teubner Verlag, Stuttgart, Leipzig, Wiesbaden, 2001.

Schinzer, 2000: Schinzer, Heiko: "MarktüberblickOLAP- und Data Mining-Werkzeuge", in: Muksch, Harry, Behme, Wolfgang (Hrsg.): Das Data Warehouse-Konzept, 4. Aufl., Gabler Verlag, Wiesbaden, 2000.

Schnizer et al., 1998: Schnizer, Heiko D.; Bange, Carsten: "Werkzeuge zum Aufbau analytischer Informationssysteme", in: Chamoni Peter; Gluchowski, Peter (Hrsg.): Analytische Informationssysteme, Springer Verlag, Berlin, Heidelberg, 1998.

Störl et al., 1998: Störl, Uta; Großmann, Gert: "Backup- und Recovery-Mechanismen in Datenbanksystemen", in: HMD - Praxis der Wirtschaftsinformatik, 200/1998.

Störl, 2001: Störl, Uta: "Backup und Recovery in Datenbanksystemen", Teubner Verlag, Stuttgart, Leipzig, Wiesbaden, 2001.

Stropek, 2005: Stropek, Rainer: "Slowly Changing Dimensions in SQL Server 2005 SSIS", http://www.cubido.at/, 21.08.2005.