



Thema:

**Optimierung der Entscheidungsunterstützung
zur Projekt- und Ressourcenplanung mit Hilfe von Business Intelligence
innerhalb der IT-Abteilung eines Automobilherstellers**

Diplomarbeit

Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Prof. Dr. Hans-Knud Arndt
Zweitgutachter: Prof. Dr. Thomas Schulze
Betreuer: Matthias Finster (GOE-3/2) Betreuung im Fachbereich

Vorgelegt von: Oliver Krohne

Abgabetermin: 07.02.07

Inhaltsverzeichnis

| | |
|--|-----|
| Inhaltsverzeichnis..... | II |
| Verzeichnis der Abkürzungen und Akronyme..... | IV |
| Abbildungsverzeichnis..... | VI |
| Tabellenverzeichnis..... | IX |
| 1 Motivation..... | 1 |
| 1.1 Aufgabenstellung und Zielsetzung..... | 2 |
| 1.2 Vorgehensweise..... | 2 |
| 2 Business Intelligence-Grundlagen..... | 3 |
| 2.1 Historie..... | 3 |
| 2.2 Begriffsabgrenzung und Definition..... | 5 |
| 2.3 Daten-Layer..... | 8 |
| 2.3.1 Strukturierte und unstrukturierte Daten..... | 8 |
| 2.3.2 Data Warehouse Konzept..... | 9 |
| 2.3.3 Multidimensionalität der Daten..... | 15 |
| 2.3.4 Multidimensionale Datenmodellierung..... | 17 |
| 2.4 Logik-Layer..... | 23 |
| 2.4.1 Konzeptorientierte Systeme..... | 25 |
| 2.4.2 Generische Basissysteme..... | 26 |
| 2.5 Präsentations-Layer..... | 29 |
| 2.6 Resümee..... | 29 |
| 3 Business Intelligence zur Projekt- und Ressourcenplanung..... | 30 |
| 3.1 Ausgangssituation..... | 31 |
| 3.1.1 Vorgehensweise zur Ist-Aufnahme..... | 32 |
| 3.1.2 Darstellung der Ausgangssituation..... | 33 |
| 3.2 Projektmanagement-Anwendung PMM..... | 37 |
| 3.2.1 Beschreibung der Anwendung..... | 38 |
| 3.2.2 Datenmodell..... | 48 |
| 3.3 Integration von Business Intelligence-Komponenten..... | 50 |
| 3.3.1 PMM-Daten-Layer..... | 55 |
| 3.3.2 PMM-Präsentations- bzw. Logik-Layer..... | 86 |
| 3.4 Resümee..... | 103 |
| 4 Optimierung der Entscheidungsunterstützung..... | 105 |
| 4.1 Potenzialanalyse..... | 105 |
| 4.1.1 Schwachstellen der Datenbereitstellung für Standard-Reports..... | 106 |
| 4.1.2 Schwachstellen der Datenbereitstellung für Ad-hoc-Analyse..... | 107 |
| 4.1.3 Optimierung..... | 109 |
| 4.2 Entwicklung eines Data Warehouse Konzeptes..... | 112 |
| 4.2.1 Datenselektion..... | 113 |

| | |
|---|-----|
| 4.2.2 Datenmodellierung..... | 115 |
| 4.2.3 Data Warehouse Architektur..... | 119 |
| 4.2.4 Weitere Vorgehensweise zur Einführung eines Data Warehouse..... | 122 |
| 4.3 Resümee..... | 124 |
| 5 Schlussbetrachtung..... | 126 |
| A. Ressourcenplanung in Microsoft Excel..... | 128 |
| B. Finanz- und Kapazitätenplanung in Microsoft Access..... | 129 |
| C. BusinessObjects Universen..... | 130 |
| D. PMM Standard-Reports..... | 131 |
| Literaturverzeichnis..... | 132 |
| Abschließende Erklärung..... | 137 |

Verzeichnis der Abkürzungen und Akronyme

| | |
|--------|--|
| 2NF | zweite Normalform |
| 3NF | dritte Normalform |
| BI | Business Intelligence |
| BO | BusinessObjects |
| BSC | Balanced Scorecard |
| C-DWH | Core Data Warehouse |
| CRM | Customer Relationship Management |
| CR.NET | Crystal Reports für .NET |
| CSV | Comma Separated Values |
| DML | Data Manipulation Language |
| DIN | Deutsche Institut für Normung |
| DLV | Dienstleistungsvereinbarung |
| DSS | Decision Support Systems |
| DWH | Data Warehouse |
| DWHMS | Data Warehouse Managementsystem |
| EIS | Executive Information Systems |
| ERM | Entity-Relationship-Model |
| ESS | Executive Support Systems |
| ETL | Extraction, Transformation and Loading |
| FASMI | Fast Analysis of Shared Multidimensional Information |
| Hrsg. | Herausgeber |
| IS | Information Systems |
| IT | Information Technology |
| JDBC | Java Database Connectivity |
| Jg. | Jahrgang |
| MDX | Multidimensional Expressions |
| mERM | multidimensional ERM |
| MIS | Management Information Systems |
| MIT | Massachusetts Institute of Technology |
| MML | Multidimensional Modeling Language |
| MS | Microsoft |
| MSS | Management Support Systems |
| mUML | multidimensional UML |
| MUS | Managementunterstützungssysteme |

| | |
|--------|----------------------------------|
| o. Jg. | ohne Jahrgang |
| o. O. | ohne Ort |
| o. V. | ohne Verfasser |
| ODBC | Open Database Connectivity |
| ODS | Operational Data Store |
| OLAP | Online Analytical Processing |
| OLTP | Online Transaction Processing |
| PP | Produktionsprozess |
| ROLAP | Relational Analytical Processing |
| SQL | Structured Query Language |
| UML | Unified Modelling Language |
| VB.NET | Visual Basic .NET |
| VS.NET | Visual Studio .NET |
| XML | Extensible Markup Language |

Abbildungsverzeichnis

| | |
|---|----|
| Abb. 2.1: Business Intelligence-Ordnungsrahmen..... | 6 |
| Abb. 2.2: Data Warehouse Architekturvarianten..... | 11 |
| Abb. 2.3: Die ODS erweiterte DWH Architektur..... | 12 |
| Abb. 2.4: Dreidimensionaler Daten-Würfel..... | 16 |
| Abb. 2.5: Ebenen der Datenmodellierung..... | 17 |
| Abb. 2.6: Grafische Darstellung: Entity-Relationship-Modell..... | 18 |
| Abb. 2.7: Das multidimensionale Entity-Relationship-Modell..... | 19 |
| Abb. 2.8: Das klassische Star Schema..... | 21 |
| Abb. 2.9: Das Snowflake Schema..... | 22 |
| Abb. 2.10: Übersicht verschiedener MSS..... | 24 |
| Abb. 3.1: Ablaufdiagramm – Ausgangssituation Reporterstellung..... | 36 |
| Abb. 3.2: Einordnung der BI-Komponenten von PMM..... | 38 |
| Abb. 3.3: PMM – Use-Case-Diagramm..... | 40 |
| Abb. 3.4: PMM – Projektdaten..... | 41 |
| Abb. 3.5: PMM – Plan-Ist-Vergleich..... | 42 |
| Abb. 3.6: PMM – Mitarbeiterkapazitäten..... | 44 |
| Abb. 3.7: PMM – Zeiterfassung/ Projektzeit..... | 44 |
| Abb. 3.8: PMM – Software-/ Lizenzkosten..... | 46 |
| Abb. 3.9: PMM – Zuordnung Software/ Organisationsstruktur..... | 46 |
| Abb. 3.10: PMM – Entity-Relationship-Modell..... | 49 |
| Abb. 3.11: Crystal Reports – Layout..... | 57 |
| Abb. 3.12: CR.NET – Verknüpfungsassistent..... | 58 |
| Abb. 3.13: CR.NET – Formeleditor..... | 60 |
| Abb. 3.14: Crystal Reports – Gruppierungen..... | 60 |
| Abb. 3.15: CR.NET – Monatssortierung nach festgelegter Reihenfolge..... | 61 |
| Abb. 3.16: Vollständiger Crystal Report..... | 61 |
| Abb. 3.17: CR.NET – Parameter erstellen..... | 62 |
| Abb. 3.18: CR.NET – Auswahl-Assistent..... | 63 |
| Abb. 3.19: BO Kompatible Datenbankschemata..... | 66 |
| Abb. 3.20: Vorgehensweise zur Universenerstellung..... | 67 |
| Abb. 3.21: BO Designer – Datenquelle und Tabellen auswählen..... | 69 |
| Abb. 3.22: BO Designer – Join erstellen..... | 70 |
| Abb. 3.23: BO Designer – Join/ Kardinalitäten definieren..... | 71 |

| | |
|--|-----|
| Abb. 3.24: BO Designer – Join-Pfadproblem (Schleife)..... | 73 |
| Abb. 3.25: BO Designer – Chasm Trap..... | 75 |
| Abb. 3.26: BO Designer – Join-Pfadproblem (Fan Trap)..... | 77 |
| Abb. 3.27: Klassen- und Objektansichtn der Entwickler- und Endanwendersicht..... | 78 |
| Abb. 3.28: BO Designer – Objekt erstellen..... | 79 |
| Abb. 3.29: BO Designer – Hierarchien definieren..... | 81 |
| Abb. 3.30: BO Designer – Universum Projektmanagement..... | 85 |
| Abb. 3.31: Standard-Report – Intervallauswahl..... | 87 |
| Abb. 3.32: PMM Berichtswesen Auswahlmenü..... | 89 |
| Abb. 3.33: Standard-Report – Zeiterfassung/ Intervall/ Gruppierung..... | 89 |
| Abb. 3.34: Standard-Report – Zeiterfassung/ Warnung..... | 90 |
| Abb. 3.35: Standard-Report – Projektarbeitszeiten der Mitarbeiter (jahresbezogen)..... | 90 |
| Abb. 3.36: Standard-Report – Software/ Organisationseinheit..... | 91 |
| Abb. 3.37: Standard-Report – Software/ Lizenzen..... | 92 |
| Abb. 3.38: Standard-Report – Hardware..... | 93 |
| Abb. 3.39: Standard-Report – Standort..... | 93 |
| Abb. 3.40: PMM – Controlling Auswahlbutton..... | 95 |
| Abb. 3.41: Ad-hoc Abfragen – Projektmanagement..... | 96 |
| Abb. 3.42: Ad-hoc Abfragen – Mitarbeiterkapazitätenplanung..... | 97 |
| Abb. 3.43: Ad-hoc-Abfragen – Ressourcenplanung..... | 98 |
| Abb. 3.44: SQL-Generierung der Ad-hoc-Abfrage..... | 99 |
| Abb. 3.45: Ad-hoc-Abfragen – Drill-down..... | 100 |
| Abb. 3.46: Ad-hoc-Abfragen – Slice & Dice..... | 100 |
| Abb. 3.47: Ablaufdiagramm – Reporterstellung mit BI-Komponenten..... | 102 |
| Abb. 4.1: Ressourcenbelastung bei Abfragen auf operative und dispositive Datenbestände..... | 110 |
| Abb. 4.2: Semantisches Data Warehouse Datenmodell..... | 113 |
| Abb. 4.3: Logisches Data Warehouse Datenmodell..... | 117 |
| Abb. 4.4: Ansatz für BI zur Projekt- und Ressourcenplanung..... | 121 |
| Abb. A.1: Bebauungsplanung in Microsoft Excel..... | 128 |
| Abb. A.2: Softwareanschaffungs- und Softwarewartungskostenplanung in MS Excel..... | 128 |
| Abb. B.1: Programm „Peanuts“ zur Finanz- und Kapazitätenplanung..... | 129 |
| Abb. B.2: Programm „Time“ zur Zeiterfassung..... | 129 |
| Abb. C.1: BO Designer – Universum Mitarbeiterkapazitätenplanung..... | 130 |
| Abb. C.2: BO Designer – Universum Ressourcenplanung..... | 130 |

Abb. D.1: Standard-Report – Letzter Monat..... 131
Abb. D.2: Standard-Report – Projektarbeitszeiten der Mitarbeiter (tagesbezogen)..... 131

Tabellenverzeichnis

| | |
|--|-----|
| Tab. 3.1: Methoden zur Ermittlung und Lösung von Join-Problemen..... | 77 |
| Tab. 3.2: Vereinfachte Ausgangstabelle: PROJECT_STRUCTURE..... | 82 |
| Tab. 3.3: Vereinfachte Zieltabelle ohne Elementtyp: PROJECT_STRUCTURE..... | 82 |
| Tab. 4.1: Operative vs. dispositive Datenbanken..... | 111 |
| Tab. 4.2: Dimensionstabelle: Zeit..... | 118 |

1 Motivation

Der Wettbewerb auf den Märkten ist heute weltweit konkurrenzbetonter und dynamischer als in der Vergangenheit. Erfolgreich sind Unternehmen, die Innovationen und Neuentwicklungen primär auf den Markt bringen (vgl. Holthuis (1999), S. 1). Besonders in der Automobilindustrie ist der Markt stark umkämpft, da der Kunde die Produkte der Hersteller enorm beeinflusst. Die Kunden wünschen immer leichtere, sicherere, komfortablere und kostengünstigere Fahrzeuge. Dazu ist es in der heutigen Zeit erforderlich, hochentwickelte Technologien einzusetzen. Diese Umstände des globalen Wettbewerbs sowie die massiven betriebswirtschaftlichen und technologischen Veränderungen stellen die Unternehmen vor erhebliche Herausforderungen, vor allem bezüglich der Gestaltung ihrer Informationsversorgung. Dabei stehen nicht mehr nur die Sammlung und der Austausch von Daten im Rahmen der Abwicklung der Geschäftsprozesse im Vordergrund, sondern vielmehr die fachgerechte Entscheidungsunterstützung (vgl. Seufert/Lehmann (2006), S. 21).

Auf der Managementebene werden zur Sammlung, Aufbereitung, Anzeige und Analyse von unternehmensweiten Informationen sogenannte *Management Support Systems* (MSS) zur Informationsgenerierung und Entscheidungsunterstützung verwendet. In diesem Zusammenhang hat sich seit den letzten Jahren eine Vielzahl von Werkzeugen und Methoden, wie Data Warehouse (DWH), Online Analytical Processing (OLAP) und Data Mining unter dem Begriff *Business Intelligence* (BI) etabliert. Jedoch ist es nicht nur auf der Managementebene von Bedeutung, für Entscheidungen Informationen über die relevanten Einflussfaktoren zu haben, sondern ebenso für Abteilungs- und Projektleiter. Ihr Anliegen ist eine optimale Ausnutzung der verfügbaren Ressourcen und eine optimale Planung und Steuerung der laufenden und zukünftigen Projekte, um in der Gesamtheit des Unternehmens ihren Einfluss auf die Wettbewerbsfähigkeit zu steigern. Um diesen Ansprüchen gerecht zu werden, entwickelt sich die Tendenz, BI und MSS ebenso auf den unteren Managementebenen einzusetzen.

Daher wird in dieser Diplomarbeit nach einer Abgrenzung des Begriffs BI und Erläuterung der damit verbundenen Ansätze und Systeme an einem praktischen Beispiel die Integration von BI-Komponenten zur optimalen Projekt- und Ressourcenplanung innerhalb einer IT-Abteilung eines Automobilherstellers vorgestellt. Die kritische Betrachtung dieser Integration zur Entscheidungsunterstützung führt im Abschluss dieser Arbeit zur Entwicklung eines DWH Konzeptes als Optimierungsansatz.

1.1 Aufgabenstellung und Zielsetzung

Ziel dieser Diplomarbeit ist die Optimierung der Projekt- und Ressourcenplanung bezüglich der Entscheidungsunterstützung innerhalb der Digitalen Fabrik- und Fertigungsplanung eines Automobilherstellers mit Hilfe von Business Intelligence Komponenten und der Entwicklung eines DWH Konzeptes. Anhand eines praktischen Beispiels werden Business Intelligence Komponenten in eine Anwendung zur Projekt- und Ressourcenplanung der oben erwähnten Abteilung integriert. Der Entwurf und die Implementierung der BI-Komponenten sollen im Zusammenhang mit der vorliegenden Diplomarbeit unter den gegebenen Voraussetzungen des Automobilherstellers umgesetzt werden. Die Verwendung der BI-Komponenten soll zur Verbesserung der Entscheidungsunterstützung beitragen. Anschließend werden aufgrund einer kritischen Betrachtung der integrierten BI-Komponenten Schwachstellen aufgezeigt und Methoden zur Behebung dieser vorgestellt. Dazu wird abschließend ein DWH Konzept entwickelt, um einen optimalen Nutzen der Entscheidungsunterstützung bezüglich Zeit, Kosten und Qualität innerhalb der Digitalen Fabrik- und Fertigungsplanung zu erzielen.

1.2 Vorgehensweise

Zu Beginn der vorliegenden Arbeit werden in Kapitel 2 die Grundlagen zum Verständnis des Themas „Optimierung der Entscheidungsunterstützung zur Projekt- und Ressourcenplanung mit Hilfe von Business Intelligence innerhalb der IT-Abteilung eines Automobilherstellers“ beschrieben und erläutert. Dies beinhaltet zunächst die historische Entwicklung der MSS und die Abgrenzung des Begriffs BI. Anschließend wird ein BI-Ordnungsrahmen vorgestellt, der die zurzeit in der Wissenschaft betrachteten BI-Methoden und -Systeme mit einbezieht. Der BI-Ordnungsrahmen dient als Rahmenkonzept für die Verwendung der BI-Komponenten in Kapitel 3 und . In Kapitel 3 werden die Ausgangssituation der Digitalen Fabrik- und Fertigungsplanung des Automobilherstellers und die Integration von BI-Komponenten zur Projekt- und Ressourcenplanung dargestellt und erklärt. Die Integration der BI-Komponenten wurde im Zuge dieser Diplomarbeit unter Berücksichtigung der Voraussetzungen der betrachteten Abteilung entworfen und implementiert. Im Weiteren wird in Kapitel die vorgestellte Integration von BI-Komponenten mit Hilfe einer Potenzialanalyse kritisch betrachtet, um vorhandene Schwachstellen aufzudecken und zu analysieren. Darauf aufbauend wird abschließend ein DWH Konzept als Verbesserungsvorschlag entwickelt, welches einen Optimierungsansatz für die Projekt- und Ressourcenplanung darstellt. Das letzte Kapitel (Schlussbetrachtung) dient der Zusammenfassung und gibt einen Ausblick über den zukünftigen Einsatz von DWH Lösungen.

2 Business Intelligence-Grundlagen

In diesem Kapitel werden die Grundlagen für die Thematik der vorliegenden Diplomarbeit erläutert, um im weiteren Verlauf die Vorgehensweise und die entwickelten Modelle nachvollziehen zu können.

Vorerst wird in diesem Kapitel ein einführender Überblick über die Entstehungsgeschichte von entscheidungsunterstützenden Systemen für das Management, den *Management Support Systems* (MSS) – im Deutschen Managementunterstützungssysteme – gegeben und der Begriff *Business Intelligence* (BI) abgegrenzt und definiert. Weiterhin wird ein BI-Ordnungsrahmen vorgestellt, der gängige BI-Methoden, -Konzepte, -Technologien und -Systeme, im Folgenden Komponenten genannt, vereint.

2.1 Historie

Informationssysteme, die das Management in ihren Entscheidungen unterstützen, werden schon seit den 60er Jahren eingesetzt. In dieser Zeit entstand eine Vielzahl unterschiedlicher Ansätze und Systeme, dessen Definition in der Literatur keineswegs einheitlich ist (vgl. Bauer/Günzel (2001), S. 11). MSS wird als Sammelbegriff aller Informations- und Kommunikationssysteme verwendet, welcher von SCOTT MORTEN als „the use of computers and related information technologies to support managers“ (Scott Morten (1983), S. 5) definiert wurde. Weiterhin lassen sich jedoch drei Klassen von MSS unterscheiden, die die historische Entwicklung wiedergeben und im Weiteren angeschnitten werden. Dabei handelt es sich um Management Information Systems (MIS), Decision Support Systems (DSS) und Executive Information Systems (EIS).

Das Bedürfnis, dem Management entscheidungsunterstützende Systeme bereit zu stellen, begann in den 60er Jahren des letzten Jahrhunderts mit den MIS. Die Informationen wurden ohne Modellbildung und logisch-arithmetische Bearbeitungen aus operativen Datenbeständen extrahiert (vgl. Gluchowski et al. (1997), S. 149). Die MIS scheiterten an der Benutzerakzeptanz und den technischen Möglichkeiten (vgl. Kemper et al. (2004), S. 1). In den 1970er Jahren sollten Manager durch DSS unterstützt werden. Diese stellten abgeleitete, entscheidungsrelevante Informationen durch modellgestützte Analyse-, Prognose- und Simulationsmethoden bereit. Da diese Systeme jedoch besonderes Fachwissen voraussetzten, konzentrierten sie sich ausschließlich auf die Unterstützung der reinen Entscheidungsprozesse (vgl. Gluchowski et al. (1997), S. 167f). Bis Mitte der 1980er Jahre konnten jedoch weiterhin kaum dynamische Analysen vom Entscheider selbst durchgeführt werden und es überwog die innerbetriebliche Sicht auf die Daten. Erst zum Ende der 1980er Jahre steigerte sich das Interesse an externen Daten

wie Börsenkursen und führte zu einer Entwicklung der EIS als spezielle Informationssysteme (vgl. Kurz (1999), S. 51). Dem Entscheider wurden durch EIS entscheidungsrelevante Daten zur Verfügung gestellt. Eine intuitiv benutzbare und individuell anpassbare Benutzeroberfläche ermöglichte den Zugriff auf interne und externe Quellen ohne Entscheidungsmodell zur Selektion oder Analyse (vgl. Holthuis (1999), S. 38). Jedoch beschränkten sich diese Systeme auf statische Beobachtungen. Eine Verbesserung der MSS wurde erst durch die gravierenden Veränderungen im Information-Technology-Markt (IT-Markt), die zunehmende IT-Unterstützung unternehmensspezifischer Geschäftsprozesse und die weltweite kommerzielle Nutzung der Internettechnologie erreicht. Ausschlaggebend waren zusätzlich die Einführungen des Data Warehouse Konzeptes durch INMON (vgl. Inmon (1992)) und des On-Line Analytical Processing (OLAP) durch CODD (vgl. Codd, et al. (1993)). Diese Konzepte führten zu neuen Funktionalitäten wie Exception Handling/ Reporting, Trend- und Prognoseanalysen, multidimensionalen Analysen und Data-Mining-Methoden. Daraufhin wurden die drei Gruppen der MSS den verschiedenen Managementgruppen zugeordnet, die nach KURZ wie folgt beschrieben werden: MIS können sehr eingeschränkte Reports und Analysen erstellen. Sie sind auf die Planung, Steuerung und Kontrolle der operativen Wertschöpfungskette ausgerichtet und stehen größtenteils dem Lower-Management (Management auf den untersten Ebenen wie Gruppenleitern) zur Verfügung. Dem Middle-Management (wie Abteilungsleitern) stehen für planerische Aufgaben sowohl die MIS als auch DSS zur Verfügung. Entscheider der obersten Management-Ebene (Top-Management) werden durch EIS, die eine erweiterte Datenanalyse auf internen und externen Daten bieten, in der Entwicklung neuer, strategischer Ziele und Alternativen für die zukünftige Geschäftsentwicklung unterstützt. Zusätzlich können die Entscheider für planerische Aufgaben auf DSS zurückgreifen (vgl. Kurz (1999), S. 99).

Heutzutage entwickelt sich der Trend, BI als einen integrierten unternehmensspezifischen Lösungsansatz zu sehen, der alle Managementebenen und MSS im gesamten Unternehmen einbezieht. Dies resultiert aus der enormen Zunahme des Datenbestandes, den starken Marktveränderungen sowie den zunehmenden internen und externen Anforderungen an Transparenz (vgl. Kemper/Baars (2006), S. 9). Weiterhin etablieren sich zunehmend sogenannte Business Intelligence Portale (BI-Portale), die BI-Komponenten unter einer Weboberfläche vereinen, welche dem Entscheider dann mittels eines Standard-Web-Browsers zur Verfügung stehen. Aufgrund dieser Veränderungen wurde der Begriff Business Intelligence eingeführt und ist erstmals auf Überlegungen der GARTNER GROUP zurückzuführen (vgl. Kurz (1999), S. 597; Anandarajan et al. (2004), S. 18 f.). Bis Ende 2007 wird allein der deutsche Markt für BI laut einer Studie der META GROUP aus dem Jahre 2004 jährlich um 16% wachsen (vgl. Seufert/Lehmann (2006), S. 22).

2.2 Begriffsabgrenzung und Definition

Zur begrifflichen Abgrenzung stellt sich zuallererst die Frage, was unter *Business Intelligence* (BI) zu verstehen ist. Der Begriff kann mit Geschäftszintelligenz übersetzt werden. Denn Unternehmen müssen intelligente Strategien, Prozesse und Werkzeuge einsetzen sowie neue innovative Lösungen finden, um zukünftig wettbewerbsfähig zu bleiben. Jedoch wird die oben erwähnte Übersetzung der korrekten Definition nicht gerecht.

In frühen Begriffsabgrenzungen der GARTNER GROUP wurde BI 1996 vor allem als Sammelbegriff für diverse IT-basierte Managementunterstützungen verwendet: „*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. 19).

In den darauf folgenden Jahren wurden verschiedene Begriffsstrukturierungen und -abgrenzungen diskutiert. BI wurde zum einen als enges und zum anderen als weites BI-Verständnis definiert. Zudem wurde es als analyseorientiertes und auch als prozessorientiertes BI-Verständnis definiert (vgl. Gluchowski (2001), S. 7; Grothe/Gentsch (2000), S. 19). Es gab jedoch keine standardisierte Definition.

Die wissenschaftliche Diskussion und die grundlegende Neuorientierung der MSS waren der Anlass, die Begriffsdefinition von BI neu zu überdenken. Daher wird der Begriff *Business Intelligence* heute als organisationspezifisches Konzept interpretiert. Der englische Begriff *Intelligence* hat hierbei die Bedeutung von *Information*, welche zu generieren, speichern, recherchieren, analysieren, interpretieren und verteilen ist (vgl. Kemper/Baars (2006), S. 9).

Definition:

„Unter *Business Intelligence* wird ein *integrierter, unternehmensspezifischer, IT-basierter Gesamtansatz* zur betrieblichen Managementunterstützung verstanden.“ (Kemper/Baars (2006), S. 9)

Nach dieser wissenschaftlichen Definition kann BI nicht käuflich erworben werden. BI muss für jedes Unternehmen organisationspezifisch konzipiert und implementiert werden. Auch die auf dem Markt angebotenen BI-Komponenten stellen lediglich Werkzeuge für die Entwicklung einzelner BI-Anwendungen zur Konkretisierung von Teilaspekten im unternehmensspezifischen Gesamtansatz dar. Zusätzlich bieten die BI-Komponenten tragfähige und miteinander verknüpfbare Lösungen zur Entscheidungsunterstützung (vgl. Kemper/Baars(2006), S. 9; Gluchowski/Kemper (2006), S. 12).

Laut Definition muss BI für jedes Unternehmen individuell entworfen und angepasst werden, um den Anforderungen des Unternehmens gerecht zu werden. Um die Gestaltungsfreiheit für jeden BI-Ansatz in einem Unternehmen einzugrenzen, wurde von KEMPER ET AL. ein Rahmenkonzept in Form eines BI-Ordnungsrahmens (s. Abb. 2.2, S. 11) entworfen. Dieser Ordnungsrahmen enthält alle wichtigen Bestandteile der BI Komponenten, Ansätze und Methoden.

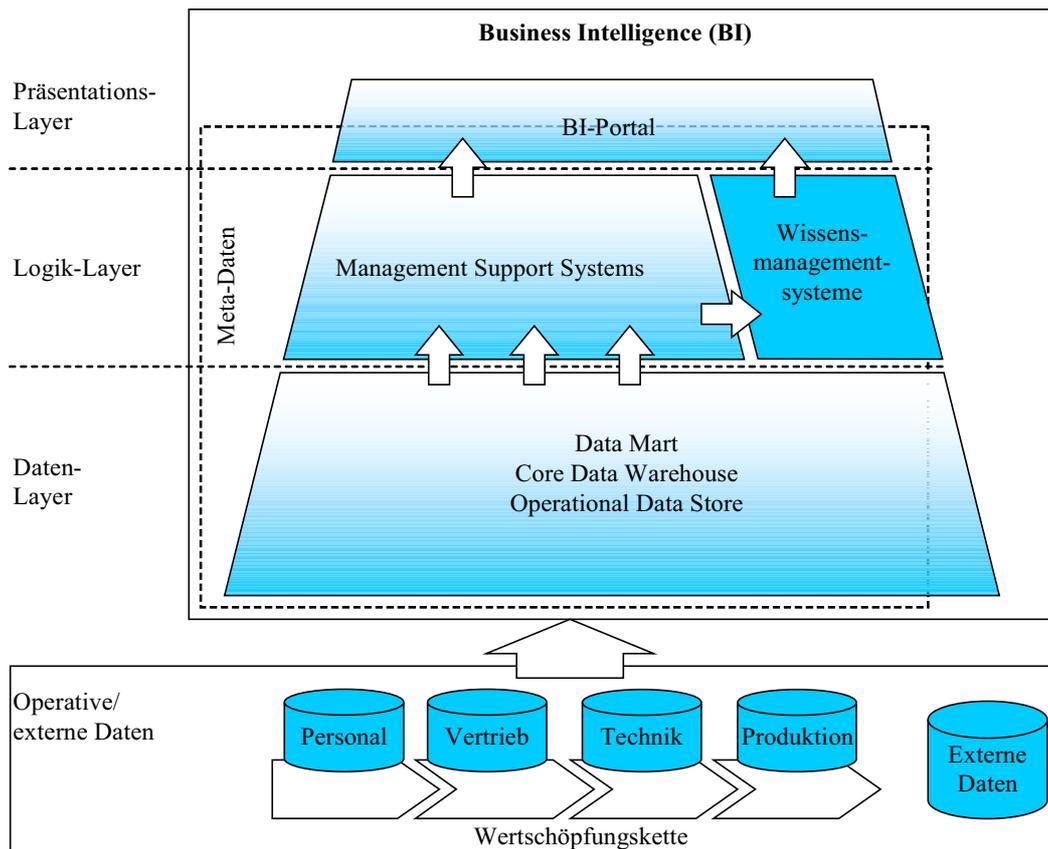


Abb. 2.1: Business Intelligence-Ordnungsrahmen

Quelle: In Anlehnung an: Kemper et al. (2004), S. 10

Der Ordnungsrahmen gliedert BI in drei Schichten, die die Grundlage zur Strukturierung der folgenden Abschnitte dieser Diplomarbeit bieten. Zusätzlich hält sich die Integration der BI-Komponenten zur Projekt- und Ressourcenplanung im weiteren Verlauf der vorliegenden Arbeit an den BI-Ordnungsrahmen.

Der Daten-Layer beinhaltet Konzepte zur Datenaufbereitung und -bereitstellung einheitlicher und betriebswirtschaftlich nutzbarer Daten. Der Logik-Layer stellt Konzepte, Verfahren und Systeme zur Informationsgenerierung, -speicherung und -verteilung zur Verfügung, wobei die Anwendungsorientierung im Vordergrund steht. Der Präsentations-Layer bereitet die Informationen individuell und personifiziert auf und bietet einen Zugriff sowie Analysemöglichkeiten zu den im Logik-Layer bereitgestellten

Funktionalitäten. Dies führt zu einer unternehmensweiten Wissensbasis und zu einer effektiven Entscheidungsunterstützung.

Bevor im Weiteren auf die verschiedenen Schichten und Konzepte eingegangen wird, werden zuvor die Begriffe Zeichen, Daten, Informationen und Wissen definiert.

Ein *Zeichen* wird als das kleinste zugreifbare Datenelement bei einer Programmausführung bezeichnet (vgl. Hansen (1992), S. 111). Es können z. B. Zahlen, Buchstaben oder Sonderzeichen sein.

Daten sind Zeichen oder Zeichenfolgen, die bekannte oder unterstellte Regeln einhalten und zur Verarbeitung verwendet werden (vgl. DIN (1992)).

Informationen sind Daten, die in einem Bedeutungszusammenhang stehen und die Personen zur Vorbereitung ihrer Entscheidungen und Handlungen nutzen (vgl. North (1998), S. 40).

Wissen wird im Allgemeinen nach Zeichen, Daten und Informationen als die höchste Ebene des Informationsflusses verstanden und wird als die „Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen“ definiert (Probst et al. (2003), S. 46; Kemper et al. (2004), S. 127).

Diese Hierarchie von Zeichen, Daten, Informationen und Wissen findet sich im BI-Ordnungsrahmen wieder. Denn Zeichen sind unter dem Daten-Layer aus den operativen Systemen als Daten abgespeichert. Daten werden im Daten-Layer konsistent und betriebswirtschaftlich aufbereitet. Im Logik-Layer entstehen durch die Verwendung der Daten in einem bestimmten Zusammenhang Informationen. Wissen erlangen die Entscheider durch die Nutzung dieser Informationen und durch die Verwendung von BI-Komponenten. Dieses Wissen wird zur Entscheidungsunterstützung eingesetzt.

Im Folgenden wird nun näher auf die verschiedenen Schichten und die dazugehörigen Konzepte eingegangen.

2.3 Daten-Layer

Unter Datenbereitstellung (engl. data delivery) wird die komplette Datenintegration, -aufbereitung und -speicherung zur konsistenten und stimmigen Datenhaltung verstanden. Nur durch intelligente Datenbereitstellung können im Logik-Layer wissenswerte und entscheidungsrelevante Informationen generiert und verteilt werden. Die bereitzustellenden Daten werden im Wesentlichen zwischen strukturierten, quantitativen und unstrukturierten, qualitativen Daten unterschieden.

2.3.1 Strukturierte und unstrukturierte Daten

Strukturierte, quantitative bzw. operative¹ Daten werden meist aus transaktionsorientierten, operativen Informationssystemen, *On-Line-Transaction-Processing-Systemen* (OLTP-Systemen), gespeichert und liegen bereits unternehmensintern in digitaler Form vor. Als OLTP-Systeme werden in der Regel operative Systeme bezeichnet, die Anfrage-, Einfüge- und Änderungsoperationen auf operativen Daten ausführen und damit die transaktionsorientierte Abwicklung des täglichen Geschäfts unterstützen. Sie sind auf betriebswirtschaftliche Funktionen ausgerichtet und geben den aktuellen Datenbestand, d. h. die Ist-Situation, wieder (vgl. Bange/Schinzer (2006)). Dies sind z. B. Datensätze aus Kundendatenbanken oder Kostenrechnungssystemen, die meist einen eindeutigen, statischen Aufbau haben. Dadurch sind sowohl die Syntax als auch die Semantik bekannt (vgl. Kurz (1999), S. 77). Das Datenmodell von operativen Datenbanken beruht auf einem rein relationalen, normalisierten Datenschema. Im Allgemeinen sind operative Systeme und Datenbanken für die Bewältigung großer Transaktionsmengen konzipiert und nicht für Analysezwecke. Um den besonderen Anforderungen der Entscheidungsunterstützung hinsichtlich Datenmengen und -modellen zu genügen, erfolgt die Datenbereitstellung und -integration in Data Warehouses durch Aufbereitung der operativen Daten aus OLTP-Datenbanken (s. Abschnitt 2.3.2). Die für Analysezwecke und entscheidungsorientierte Systeme benötigten Daten werden dispositive² Daten genannt, da sie – abgeleitet von der ursprünglichen Definition GUTENBERGS – „mit der Leitung und Lenkung der betrieblichen Vorgänge in Zusammenhang stehen“ (vgl. Gutenberg (1983), S. 3). Diese Daten müssen in geeigneter Struktur aufbereitet werden, um den managementunterstützenden Anforderungen gerecht zu werden.

¹ „operativ [lat.-nat.]: [...] (als konkrete Maßnahme) unmittelbar wirkend“ (Drosdowski et al. (1982), S. 542).

² „dispositiv [lat.-nat.]: anordnend, verfügend“ (Drosdowski et al. (1982), S. 191).

Unstrukturierte bzw. semistrukturierte, qualitative Daten haben keinen bzw. einen grob strukturierten Aufbau. Der Inhalt der Daten ist frei wählbar und keinesfalls vordefiniert. Diese Art von Daten liegt in gespeicherten E-Mails oder Dokumenten vor, aber auch das Internet oder Diskussionsforen bieten einen Zugriff auf unstrukturierte, qualitative Daten (vgl. Grothe/Gentsch (2000), S. 21).

Der Daten-Layer hat die Aufgabe sowohl die strukturierten, quantitativen Daten als auch die unstrukturierten, qualitativen Daten aus internen und externen Quellen in standardisierter, konsistenter Weise aufzubereiten und bereitzustellen. Heutzutage erfolgt dies meist in Data Warehouses für strukturierte, quantitative Daten sowie in Content- und Document Management Systems für unstrukturierte, qualitative Daten (vgl. Kemper/Baars (2006), S. 12). Da sich der Kern der vorliegenden Arbeit aber auf Data Warehouses und strukturierte, quantitative Daten bezieht, wird hier nicht näher auf die erwähnten Konzepte eingegangen, sondern im Folgenden das Data Warehouse Konzept und die multidimensionale Datenmodellierung beschrieben.

2.3.2 Data Warehouse Konzept

Der grundlegende Gedanke des *Data Warehouse* (DWH) Konzeptes lässt sich in Anlehnung an MUCKSCH und BEHME wie folgt beschreiben. Operative und externe Daten, d. h. strukturierte, quantitative Daten, werden durch Datensammlung und -integration in einer isolierten Datenbank aus dispositiven Daten für entscheidungsunterstützende Aufgaben und Analysezwecke aufbereitet. Diese werden dann anschließend für alle Arten von MSS als unternehmensweite Datenbank zur Verfügung gestellt. Dabei kennzeichnet die strikte Trennung von operativen und dispositiven Daten sowie Systemen das Data Warehouse (vgl. Mucksch/Behme (1998), S. 36). Die Definition des *Data Warehouse* stammt ursprünglich von INMON. Er definiert es als:

„subject oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions“ (Inmon (1992), S. 29).

Dadurch lassen sich die Haupteigenschaften eines DWH folgendermaßen beschreiben:

- *Subjektorientierung*

Die dispositiven Daten in einem DWH sind auf die managementunterstützenden Aufgaben (Leitung und Lenkung des Unternehmens) und Anforderungen (schneller, direkter und interaktiver Zugriff auf analysegerechte Daten) ausgerichtet. Daher sind die Daten nach Sachverhalten (Themen) aufbereitet.

- *Integration*

Die Daten aus operativen und externen Quellen müssen in eine konsistente, einheitliche Struktur des DWH transformiert werden (vgl. Kemper et al. (2004) S. 18).

- *Nicht Volatilität*

Bei einem DWH handelt es sich um eine dauerhafte, d. h. permanente Sammlung der Daten. Es existieren beim DWH nur zwei Operationen: Daten in das DWH integrieren oder auf diese zugreifen (vgl. Inmon/Hackathorn (1994), S. 10).

- *Zeitraumbezug*

Die Daten in einem DWH werden in verschiedenen Verdichtungsstufen über einen gewissen Zeitraum gespeichert. Dieser Zeitraum kann bis zu zehn Jahre betragen (vgl. Holthuis (1999), S. 73). Der Hintergedanke liegt darin, Trendanalysen oder zeitraumbezogene Überblicke der Daten zu Entscheidungszwecken heranziehen zu können.

Die zugrunde liegende Datenbank eines DWH kann objektorientiert, relational, multidimensional oder sogar hierarchischer Natur sein (vgl. Hahne (1999), S. 146). Es hängt meist vom Anbieter der Datenbank ab, welches Datenmodell verwendet wird. Anhand der oben beschriebenen Charakteristika eines DWH werden Zweck und Wert der Daten sichtbar. Es tauchen jedoch die Fragen auf, welche Komponenten zu einem DWH gehören und wie Daten integriert werden. Zudem muss die Frage beantwortet werden, welche Datenstruktur für MSS am besten geeignet ist. Um Antworten auf diese Fragen geben zu können, werden im Folgenden die verschiedenen DWH Architekturen vorgestellt und der DWH Lebenszyklus beschrieben. Die Struktur der Daten und die Modellierungstechniken werden im Abschnitt 2.3.3 und 2.3.4 besprochen.

Data Warehouse Architekturvarianten

In der Literatur ist eine Vielzahl unterschiedlicher Architekturvarianten beschrieben. Welche Architektur für ein Unternehmen oder eine Abteilung sinnvoll ist, muss jeweils spezifisch durch eine Anforderungsanalyse bei der Entwicklung eines DWH berücksichtigt werden. Im Folgenden wird auf die gängigsten Architekturvarianten eingegangen.

Die *virtuelle Data Warehouse Architektur* zeichnet sich dadurch aus, dass sie einen direkten Zugriff von den Endanwendersystemen, den MSS oder Analysesystemen auf die operativen und externen Daten erlaubt (s. Abb. 2.3, S. 12). Der Zugriff ist allerdings nur lesend, um ungewollte Änderungen an den operativen Daten zu verhindern (vgl. Schinzer/Bange (1999), S. 50). Eine Datenaufbereitung und -integration in einer isolierten Datenbank ist also nicht vorhanden. Die Struktur und die Tabellen der operativen Datenbank werden innerhalb der MSS in eine interne Datenbankstruktur und SQL-Objekte³ umgewandelt, die dem Endanwender die betriebswirtschaftliche Sicht widerspiegeln. Dadurch können die MSS überwiegend ähnliche Funktionalitäten wie mit Hilfe anderer DWH Architekturen bereitstellen.

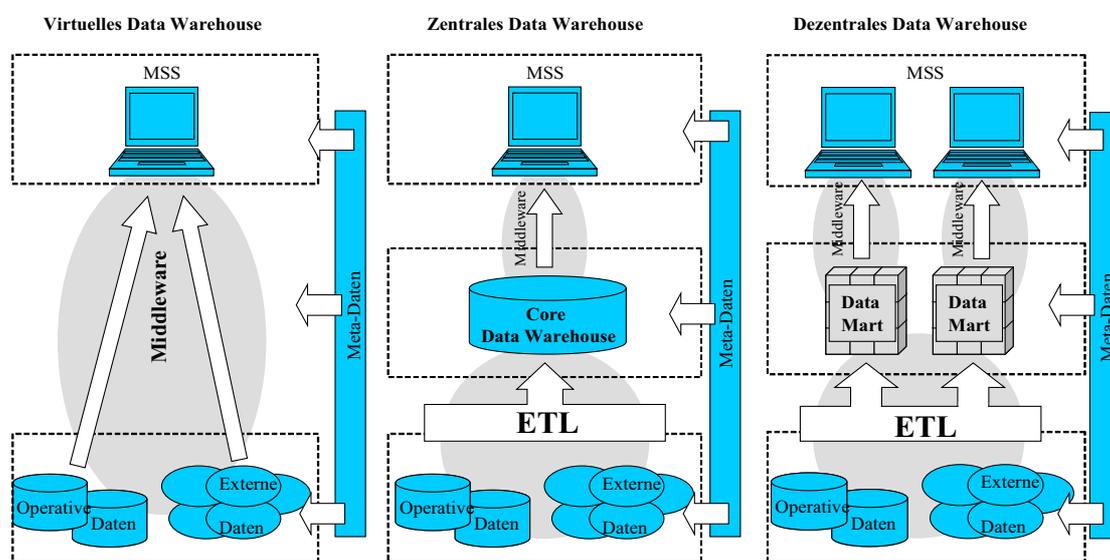


Abb. 2.2: Data Warehouse Architekturvarianten

Quelle: In Anlehnung an: Schinzer et al. (1997), S. 20

Bei einer *zentralen Data Warehouse Architektur* (s. Abb. 2.3, S. 12) werden alle dispositiven Daten getrennt von den operativen und externen Daten in einer isolierten Datenbank gehalten. Es folgt der Definition eines DWH (s. Abschnitt 2.3.2). Die operativen Daten werden durch *Extraction-Transformation-Load*-Prozesse (ETL-Prozesse) in dispositive Daten überführt. Die dispositiven Daten werden von einem *Data Warehouse Management-System* (DWHMS) verwaltet. Die Anfrageberechnungen und Auswertungen finden auf dem Server statt und nur die letztendlich benötigten Daten werden an den Client übermittelt. (vgl. Kemper et al. (2004), S. 19; Schinzer/Bange (1999), S. 51).

³ SQL-Objekte enthalten SQL-Befehle und spiegeln die betriebswirtschaftliche Sicht des Endanwenders wider. Bei der Abfrage des Endanwenders werden die verwendeten SQL-Objekte zu einem SQL-Ausdruck verknüpft und an die Datenbank versendet.

Eine *dezentrale Data Warehouse Architektur* (s. Abb. 2.3, S. 12) besteht aus subjektspezifischen oder abteilungsspezifischen Data Warehouses, den sogenannten *Data Marts*. Die dispositiven Daten werden bereits abteilungsspezifisch oder für gewisse Anforderungen performanceoptimiert durch ETL-Prozesse aufbereitet und bereitgestellt. Dies kann besonders bei örtlich getrennten Unternehmensbereichen von Nutzen sein (vgl. Schinzer/Bange (1999), S. 52).

Bei einer *Operational Data Store (ODS) erweiterten Data Warehouse Architektur* (s. Abb. 2.4, S. 16) handelt es sich um ein Zusammenwirken mehrerer, dispositiver Datenbanksysteme. Sie wird in der Literatur häufig präferiert, allerdings auch kontrovers diskutiert (vgl. Kemper et al. (2004); Inmon et al. (1996); Mucksch/Behme (2000)).

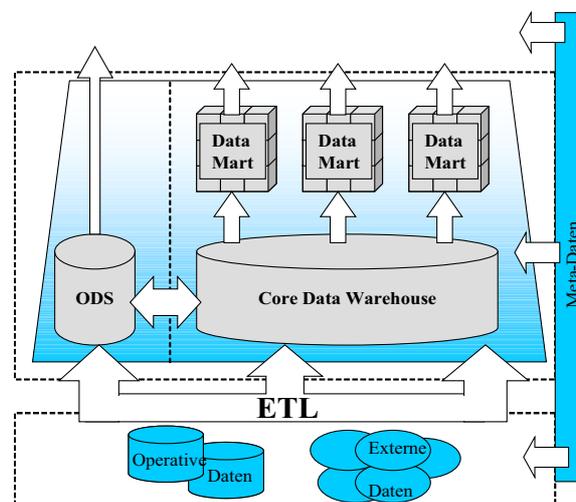


Abb. 2.3: Die ODS erweiterte DWH Architektur

Quelle: In Anlehnung an: Kemper et al. (2004), S. 20

Das Herzstück dieser Architektur ist das sogenannte *Core Data Warehouse (C-DWH)*. Es entspricht der oben genannten Definition eines DWH (s. Abschnitt 2.3.2). Die Data Marts haben die gleichen Eigenschaften wie die einer dezentralen DWH Architektur, sie sind abteilungsorientiert und werden in diesem Fall durch Transformationsprozesse ähnlich der Extraction-Transformation-Load-Prozesse (ETL-Prozesse) aus dem C-DWH aufbereitet. Der ODS ist eine subjektorientierte, integrierte, volatile Architekturkomponente. Seine Daten sind aktuell und detailliert (vgl. Inmon et al. (1996), S. 15). Der Unterschied zum C-DWH besteht daher darin, dass die dispositiven Daten zeitpunktaktuell gehalten und nicht historisiert werden. Denn jede Transaktion überschreibt die Daten aufs Neue. Weiterhin enthält der ODS keine Aggregationen⁴ und die Daten sind auf eine sofortige, greifbare Durchführung von Analysen und Auswertungen ausgerichtet (vgl. Inmon et al. (1996), S. 73; Mucksch (1999), S. 176).

⁴ Unter aggregierten Werten bzw. Aggregationen sind Werte zu verstehen, die durch eine Verdichtungsfunktion – meist die Summation – zusammengefasst werden (vgl. Hahne (1999), S. 153). Es können z. B. Tagesumsätze zu Monatsumsätzen zusammenaddiert werden.

Bei jeder DWH-Architektur existieren *Meta-Daten* bzw. ein *Meta-Datenbanksystem*. Es wird auch als Repository bezeichnet (vgl. Ortner (1999), S. 236 f.) Die Meta-Daten beschreiben die Datenstruktur der gespeicherten Daten des DWH. Sie enthalten einen Informationskatalog, der alle Objekte in der Terminologie der Endbenutzer beschreibt (vgl. Mucksch/Behme (2000), S. 22 f.). Meta-Daten, d. h. Repositories, dienen dem Endanwender als Hilfesystem und Informationskatalog über die gespeicherten Strukturen und Daten im DWH.

Voranehend wurden das DWH Konzept als separate Datenbasis zur Entscheidungsunterstützung und die verschiedenen Architekturen und Komponenten von DWH-Lösungen vorgestellt. Es bleibt jedoch offen, wie ein DWH entsteht und wie die Daten in das DWH gelangen. Der Prozess der Datenintegration und Bereitstellung aus den operativen und externen Daten im DWH wird als Transformationsprozess oder ETL-Prozess bezeichnet und wird während des Aufbaues eines DWH definiert. Auf diesen Aufbau eines DWH wird im folgenden Abschnitt eingegangen.

Aufbau eines Data Warehouse

Das DWH ist als Teil des BI-Gesamtkonzeptes zu sehen und legt den Grundstein für die Datenbereitstellung. Welche Architekturvariante und BI-Infrastruktur für ein Unternehmen in Frage kommen, muss individuell entschieden werden. Daher wird im nachfolgenden Rahmenkonzept ausschließlich der Aufbau eines DWH beschrieben, wobei Soft- oder Hardwarekomponenten unberücksichtigt bleiben (in Anlehnung an: Grothe/Gentsch (2000), S. 53 ff.). Das Rahmenkonzept wird an dieser Stelle vorgestellt, da es beim späteren Entwurf des DWH Konzeptes im Abschluss der vorliegenden Arbeit Verwendung findet:

- *Datenselektion*

Während der Datenselektion wird mit Hilfe der Fachabteilungen untersucht, welche internen und externen Daten für Analysezwecke benötigt werden. Weiterhin soll das DWH als Informationsbasis für gegenwärtige und zukünftige Entscheidungen dienen. Daher müssen zukünftige Bedürfnisse miteinbezogen werden. Es muss vermieden werden, dass Informationen fehlen oder nur unvollständig vorhanden sind, denn dies beeinträchtigt die korrekte Arbeitsweise des DWH. Im Allgemeinen wird in dieser Phase die betriebswirtschaftliche Problemstellung analysiert und ein Fachkonzept entworfen, welches durch ein semantisches Datenmodell als Grundlage für die konsistente, technische Umsetzung beschrieben werden kann (vgl. Holthuis (2000), S. 162).

- *Datenmodellierung*

In dieser Phase werden schließlich die Überlegungen und Begrifflichkeiten des Fachkonzeptes und der Datenselektion in eine logische Struktur übertragen und modelliert (s. Abschnitt 2.3.4). Dabei wird eine unternehmensindividuelle, betriebswirtschaftliche Datensemantik abgestimmt, damit die gezielten Abfragen im weiteren Verlauf auf einheitlichen, den Mitarbeitern im Unternehmen bekannten Begrifflichkeiten beruhen. Als Ergebnis werden die Daten des DWH in einem logischen Datenmodell modelliert, das die Grundlage für eine technische Implementierung bietet.

- *Datenextraktion/Datenimport (ETL-Prozess)*

Während dieser Phase werden die selektierten und im Datenmodell abgebildeten Daten extrahiert, transformiert und in das DWH geladen. Dieser Transformationsprozess wird als ETL-Prozess bezeichnet. Im Konkreten werden während des ETL-Prozesses die Phasen *Filterung*⁵, *Harmonisierung*⁶, *Verdichtung*⁷ und *Anreicherung*⁸ durchlaufen. Zur Unterstützung der DWH-Entwickler können spezielle ETL-Tools verwendet werden. Bei ETL-Tools können u.a. Vor- und Nachbedingungen sowie Formatanpassungen eingestellt werden, damit die Daten der späteren DWH-Norm entsprechen. Zum Abschluss der Phase wird der Datenimport durchgeführt, wobei es als sehr problematisch gilt, eine zeitgenaue Synchronisation der importierten Daten zu realisieren.

- *Betrieb und Pflege des DWH*

Nach abschließendem Import der Daten in das DWH stehen nun die dispositiven Daten den Front-End-Werkzeugen zur Verfügung. Diese Analysewerkzeuge besitzen die Logik, aus den aufbereiteten Daten entscheidungsrelevante Informationen zu generieren (s. Abschnitt 2.4). In dieser letzten Prozessphase müssen jedoch auch die Aktualisierung der Daten, die Überwachung des DWH und die Anpassung an spätere Gegebenheiten berücksichtigt werden. Diese Phase ist ein sich ständig wiederholender Ablauf, da die Wartung und Pflege eines DWH die Funktionalität und Verfügbarkeit sicherstellt.

⁵ *Filterung* bezeichnet die Extraktion der Daten aus externen und vor allem aus internen operativen Quellen und die Beseitigung von syntaktischen und inhaltlichen Fehlern (vgl. Kemper/Finger (1999), S. 81).

⁶ Die Phase der *Harmonisierung* stimmt betriebswirtschaftliche Begrifflichkeiten der gefilterten Daten ab (vgl. Kemper/Finger (1999), S. 86).

⁷ Während der *Verdichtung* werden die zuvor gefilterten und harmonisierten Daten aggregiert. Die Daten werden zu themenbezogenen Gruppen wie Produkte, Kunde oder Regionen zusammengeführt (vgl. Kemper/Finger (1999), S. 88).

⁸ Innerhalb der Phase, *Anreicherung*, werden die betriebswirtschaftlichen Kennzahlen aus gefilterten und harmonisierten Daten gebildet und gespeichert (vgl. Kemper/Finger (1999), S. 90).

Dieser Aufbau eines DWH ist notwendig, um operative in entscheidungsrelevante, d. h. dispositive Daten, umzuwandeln. Diese werden den Entscheidern zur Verfügung gestellt, um aussagekräftige Informationen generieren und analysieren zu können. Die Daten stehen nun dem Entscheider in multidimensionaler Struktur zur Verfügung. Daraus ergibt sich die Frage, wie die multidimensionale Struktur aussieht und welche Eigenschaften sie hat. Der folgende Abschnitt beschäftigt sich zunächst mit den Eigenschaften der multidimensionalen Daten und anschließend mit deren Modellierung. Diese Erläuterungen sind nötig, da das DWH Konzept im Abschluss der vorliegenden Diplomarbeit auf diesen Datenstrukturen aufbaut.

2.3.3 Multidimensionalität der Daten

Für MSS werden dispositive Daten zur Leitung und Lenkung des Unternehmens benötigt. Diese Daten werden meist anhand mehrerer problemrelevanter Kriterien analysiert. Fragestellungen wie „Welcher Umsatz wurde im November 2004 in der Region Deutschland mit dem Produkt X erzielt?“ betreffen z. B. Umsätze für bestimmte Produkte in einer festgelegten Region über einen definierten Zeitraum. Solche Fragestellungen basieren auf der intuitiven, betriebswirtschaftlichen Vorstellung der Endanwender, welche stark von den operationalen Systemen abweicht. Daher visualisieren MSS Datenanalysen in einer multidimensionalen Struktur (vgl. Holthuis (2000), S. 151). Weiterhin besteht der Anspruch, einem großen Anwenderkreis schnellen, direkten und interaktiven Zugriff auf analysegerechte dispositive Daten zu ermöglichen, welches als *On-Line-Analytical-Processing* (OLAP) verstanden wird (vgl. Grothe/Gentsch (2000), S. 58).

Mit Hilfe von OLAP (s. Abschnitt 2.4.2) kann der Endanwender Analysen in multidimensionalen Datenräumen durchführen. Diese Datenräume werden meist als dreidimensionale Würfel modelliert, können aber beliebig viele Dimensionen enthalten (s. Abb. 2.5, S. 17). Diese Würfel enthalten relevante, aufbereitete Kenngrößen (Fakten) wie Umsatzzahlen für Einkauf, Verkauf oder Lagerbestandszahlen. Jeder Würfel besteht zusätzlich aus verschiedenen Dimensionen. Jede Dimension, wie z. B. Zeit oder Ort, kann wiederum in verschiedene Hierarchieebenen (Hierarchie) gegliedert sein. Die Zusammenfassung von Daten entlang einer Hierarchie wird Verdichtung, Aggregation oder Gruppierung genannt. Die Verdichtung erfolgt durch eine Berechnungsvorschrift, die entsprechend als Verdichtungs-, Gruppierungs- oder Aggregationsfunktion bezeichnet wird (vgl. Herden (2001), S. 20).

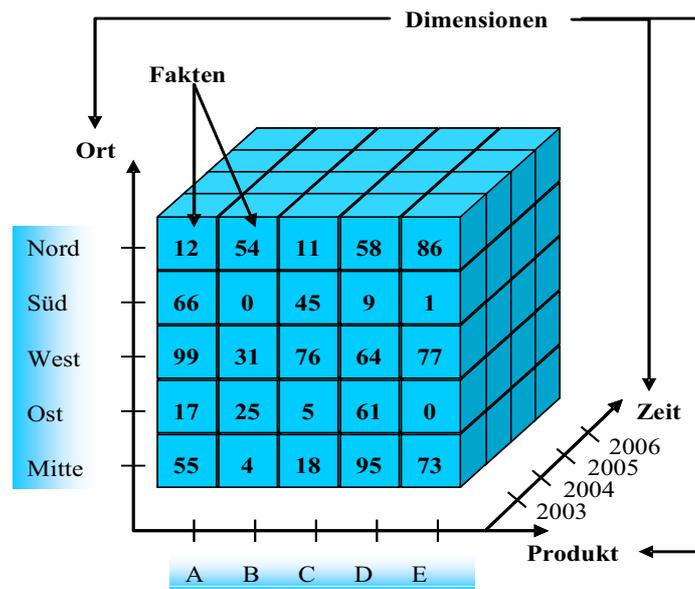


Abb. 2.4: Dreidimensionaler Daten-Würfel

Quelle: In Anlehnung an: Herden (2001), S. 20

Diese betriebswirtschaftliche, anwendungsorientierte Sicht auf die Daten und ihre Analysemöglichkeiten der Datenräume werden im weiteren Verlauf der vorliegenden Arbeit in Abschnitt 2.4 erläutert. Die Stärke der multidimensionalen Strukturen besteht in der intuitiven Datenbearbeitung. Diese folgt keinem Schema, ermöglicht die einfache Überprüfung neuer Ideen und Hypothesen und bietet Zugriff auf detaillierte Daten (vgl. Holthuis (2000), S. 152).

Um diese Datenräume zur Verfügung zu stellen, bedarf es speziell angepasster Datenstrukturen. Diese Strukturen hängen von der zugrunde liegenden DWH Datenbank ab. Prinzipiell wird aber davon ausgegangen, dass eine relationale Datenbank zugrunde liegt. Daher können Modellierungstechniken aus der relationalen Datenbanktheorie auch für das DWH Design angewendet werden (vgl. Hahne (1999), S. 146). Der Unterschied besteht darin, dass mehrdimensionale Strukturen abgebildet werden sollen und dies nach speziellen Modellen verlangt.

2.3.4 Multidimensionale Datenmodellierung

Im folgenden Abschnitt werden einführend die Grundlagen der Datenmodellierung beschrieben, um weiterführend die multidimensionale Datenmodellierung erläutern zu können.

Ein *Datenmodell* beschreibt „ein Wissen über die lebensweltliche Bedeutung (Semantik) sowie über die maschinelle Repräsentation und Manipulation von Daten“ (Wedekind (1997), S. 118), also grundsätzlich die Struktur der Daten, die aus der Realität abgebildet werden. Es werden semantische, logische und physische Datenmodelle unterschieden (s. Abb. 2.6, S. 18).

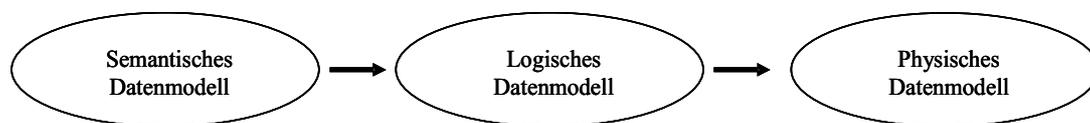


Abb. 2.5: Ebenen der Datenmodellierung

Quelle: Hahne (1999), S. 147

Physische Modelle sind auf die technische Umsetzung ausgerichtet (vgl. Hahne (1999), S. 146 f.). Da in der vorliegenden Diplomarbeit nicht auf die technische Umsetzung eingegangen wird, wird an dieser Stelle auf eine nähere Betrachtung physischer Modelle verzichtet. Im Unterschied zur technischen Ausrichtung der physischen Modelle beschreiben semantische und logische Modelle realitätsnahe Zusammenhänge.

Semantische Modelle spiegeln die typmäßige Struktur der Daten wider, ohne die Datenwerte zu beschreiben. Weiterhin sollen sie die Zusammenhänge der Verantwortungsbereiche eines Entscheiders möglichst vollständig, verständlich und überschaubar abbilden, damit auch Datenverarbeitungsunerfahrene, die im Modell abgebildeten Zusammenhänge verstehen können. Für die Datenbankdesigner soll es einen Ausgangspunkt zur Überführung in ein Datenbankmodell bzw. logisches Datenmodell bieten (vgl. Holthuis (2000), S. 159). Der bekannteste Vertreter dieser semantischen Modelle ist das *Entity-Relationship-Modell* (ERM). Auf dieses Modell wird im nachfolgenden Abschnitt näher eingegangen.

Semantische Datenmodellierung

Das *ERM* wurde 1976 erstmals von PETER PIN-SHAN CHEN eingeführt (vgl. Chen (1976)). Seine Komponenten sind Entitäten (engl. entities), deren Beziehungen (engl. relationships) untereinander und ihre Attribute (engl. attributes). In Abb. 2.7 (s. S. 19) ist die grafische Darstellung der Bestandteile des ERMs abgebildet. Eine *Entität* stellt ein Objekt der realen Welt oder der Vorstellungswelt dar. Es wird als Rechteck abgebildet. Eine *Beziehung* beschreibt eine Verknüpfung zwischen zwei oder mehreren Entitäten.

Beziehungen werden als Rauten dargestellt. *Attribute* legen die Eigenschaften von Entitäten oder Beziehungen fest und werden als abgerundete Rechtecke mit ihnen verbunden. Eindeutig beschreibende Eigenschaften der Entitäten, sogenannte Primärschlüssel, werden als unterstrichene Attribute markiert. Es existieren im ERM weitere Konstrukte wie die Spezialisierung/Generalisierung, die auch IST-Beziehung genannt wird (vgl. Heuer/Saake (2000), S. 67 f.). Die IST-Beziehung stellt Entität E2 als Spezialfall der Entität E1 dar (s. Abb. 2.7, S. 19).

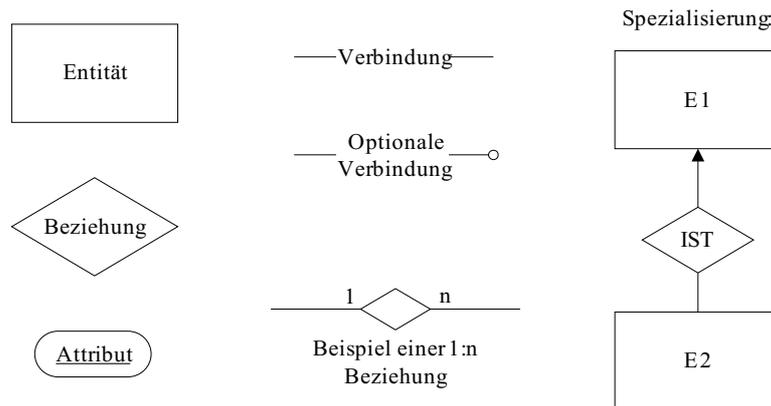


Abb. 2.6: Grafische Darstellung: Entity-Relationship-Modell

Quelle: In Anlehnung an: Chen (1976), S. 19 ff.

Des Weiteren existieren *Kardinalitäten*, um darstellen zu können, wie viele Instanzen eines Entitäten-Typs an einer Beziehung teilnehmen können. Die Kardinalitäten werden an die Verbindungen zwischen den Entitäten und der Beziehung geschrieben (vgl. Jarosch (2002), S. 68 f.; Heuer/Saake (2000), S. 68 ff.).

Nach Konzeption eines semantischen Modells wird es in einer weiteren Konzeptionsphase durch Transformationen in ein logisches Modell überführt, das die Realisierung einer Datenbank beschreibt. Hierzu wird in der Regel das *Relationenmodell* verwendet.

Logische Datenmodellierung

Das *Relationenmodell* ist ein Datenbankmodell für die Realisierung einer Datenbank und wird für die logische Modellierung verwendet. Es wurde 1970 von E. CODD (vgl. Codd (1970), S. 377 ff.) eingeführt und hat sich zum verbreitetsten Datenbankmodell entwickelt.

Relationen, die aus einer Menge von Zeilen bestehen, werden in Form von zweidimensionalen Tabellen abgebildet. Die Tabellenspalten enthalten die Attributwerte. Durch die Primärschlüssel⁹ sind die Zeilen (Tupel) eindeutig identifizierbar und beschreiben dadurch die Werte für eine Entität oder eine Beziehung. Die Beziehungen

⁹ Ein Primärschlüssel bezeichnet die Menge von Attributen, deren Werte einer Zeile einer Tabelle eindeutig bestimmt sind (vgl. Heuer/Saake (2000), S. 115).

zwischen Tabellen werden durch die Verwendung von Fremdschlüsseln¹⁰ definiert (vgl. Heuer/Saake (2000), S. 115 f.). Zur qualitativen und performanten Verbesserung von Datenmodellen in operativen Datenbanken werden während der logischen Datenmodellierung Verfahren zur Normalisierung und Minimalisierung herangezogen. Diese werden hauptsächlich zur Vermeidung von Redundanzen, also Mehrfachspeicherungen in der Datenbank eingesetzt (vgl. Heuer/Saake (2000), S. 238 ff.). In der Praxis werden jedoch kleine Verletzungen dieser Eigenschaften, meist bei der dritten Normalform (3NF), zu Zwecken von Verbesserungen der Wartung und Performance der operativen Datenbanken in Kauf genommen.

Diese Modellierungstechniken der relationalen Datenbanktheorie können, wie oben erwähnt, auch für das DWH Design und damit für den Aufbau von multidimensionalen Strukturen verwendet werden. Für multidimensionale Anforderungen sind sie jedoch erweitert und angepasst worden.

Semantische multidimensionale Modellierung

Es existiert eine Vielzahl von Modellierungstechniken für die semantische Ebene, also für die betriebswirtschaftlichen Zusammenhänge. Nachfolgend wird der Ansatz des multidimensionalen Entity-Relationship-Modells vorgestellt, da er für eine semantische Modellierung eine einfache und übersichtliche Darstellung bietet.

Das *multidimensional Entity-Relationship-Model* (mERM, vgl. Sapia et al. (1999)) ist eine Erweiterung des bekannten ERM (vgl. Chen (1976)), das speziell auf die multidimensionalen Anforderungen der Modellierung angepasst worden ist. Abb. 2.8 (s. S. 21) zeigt die grafische Darstellung eines mERM.

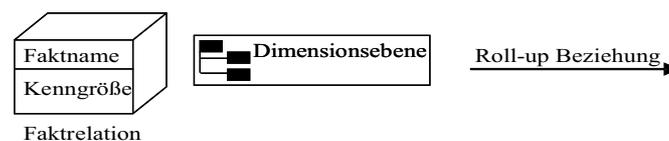


Abb. 2.7: Das multidimensionale Entity-Relationship-Modell

Quelle: In Anlehnung an: Kurz (1999), S. 177

Im mERM existiert eine *Faktrelation*, die einen eindeutigen Namen haben muss. Diese kann mit beliebig vielen Dimensionsebenen in Beziehung stehen und unbeschränkt viele *Faktdaten* (Kennzahlen) enthalten. Die *Dimensionsebenen* (Knoten) stellen die Hierarchien dar. Jede Dimensionsebene existiert nur einmal im gesamten Datenmodell. Mehrere Dimensionsebenen bilden zusammen mit den Roll-Up-Beziehungen (Kanten) einen Dimensionsgraphen (vgl. Kurz (1999), S. 175). Die *Roll-Up*-Beziehungen de-

¹⁰ Unter Fremdschlüssel wird eine Attributmengende verstanden, die in einer anderen Tabelle Primärschlüssel ist (vgl. Heuer/Saake (2000), S. 115).

finieren die Relation zwischen zwei Hierarchieebenen und bilden somit einen gerichteten, nicht-zyklischen Graphen, welcher unterschiedlich viele Aggregationspfade bzw. Verdichtungspfade besitzen kann. Weiterhin können ER-Konstrukte eingebaut werden und somit das mERM komplexer werden lassen.

Als eine Schwäche des mERM erweist sich die fehlende grafische Repräsentation abgeleiteter Attribute. Es unterstützt lediglich statische Aspekte der Daten, daher sind die Attributtypen nicht zu differenzieren (vgl. Herden (2001), S. 27). Das mERM kann jedoch zur Modellierung genutzt werden, es muss allerdings mit Modellierungseinschränkungen gerechnet werden.

Nach Entwurf dieses semantischen Datenmodells wird es in der nächsten Phase zum Aufbau eines DWH zu einem logischen Modell verfeinert.

Logische multidimensionale Modellierung

Zur logischen Modellierung von Data Warehouses auf relationaler Basis werden die Varianten des Star Schemas herangezogen. Diese haben das Ziel, multidimensionale Datenstrukturen in Relationenmodellen darzustellen (vgl. Hahne (1999), S. 152).

Im Allgemeinen kann ein Star Schema folgendermaßen definiert werden:

„Ein *Star* [...] Schema besteht aus einer variablen Anzahl von Fakttabellen, welche jeweils mit einer variablen Anzahl Dimensionstabellen verbunden sind. Dabei wird eine Dimension [...] genau durch eine relationale Tabelle logisch abgebildet“ (Kurz (1999), S. 157).

Im Gegensatz dazu bezeichnet McGUFF erstmals die Bereitstellung unterschiedlicher Fakttabellen als *Galaxie* (vgl. McGuff (1998), S. 1 f.). Zur Konsistenzsicherstellung wird von KEMPER ET AL. empfohlen, dass sich Fakttabellen strukturidentische Dimensionen teilen. In der Praxis werden Galaxien in der Regel für unterschiedliche Analysezwecke in Data Marts bereitgestellt (vgl. Kemper et al. (2004), S. 64; Hahne (1999), S. 158). Aus semantischer Sicht stellen die Faktdaten betriebswirtschaftliche Kennzahlen dar, die wichtige Zusammenhänge verdichten und diese in quantitativ messbarer Form wiedergeben (vgl. Horváth (2003), S. 566). Die Faktdaten sind meist numerischer Natur und bilden den Mittelpunkt der Datenanalyse (vgl. Poe (1996), S. 121). Zwischen den Dimensionstabellen existieren keinerlei Beziehungen. Sie bilden, angeordnet um eine Fakttable, eine sternähnliche Form nach.

Das *klassische Star* Schema (s. Abb. 2.9, S. 22) besitzt nur eine einzige große Fakt-tabelle, in der alle Faktdaten gespeichert sind. Der Primärschlüssel der Fakt-tabelle setzt sich aus den Primärschlüsseln der einzelnen Dimensionstabellen zusammen, den sogenannten Fremdschlüsseln der Fakt-tabelle. Eine 1:N-Beziehung verbindet die Fakt-tabelle mit den Dimensionstabellen (vgl. Kurz (1999), S. 159), welche die Attribute der Faktdaten bilden und aufgrund von Performanceverbesserungen im klassischen Star Schema denormalisiert sind (Abb. 2.9, S. 22).

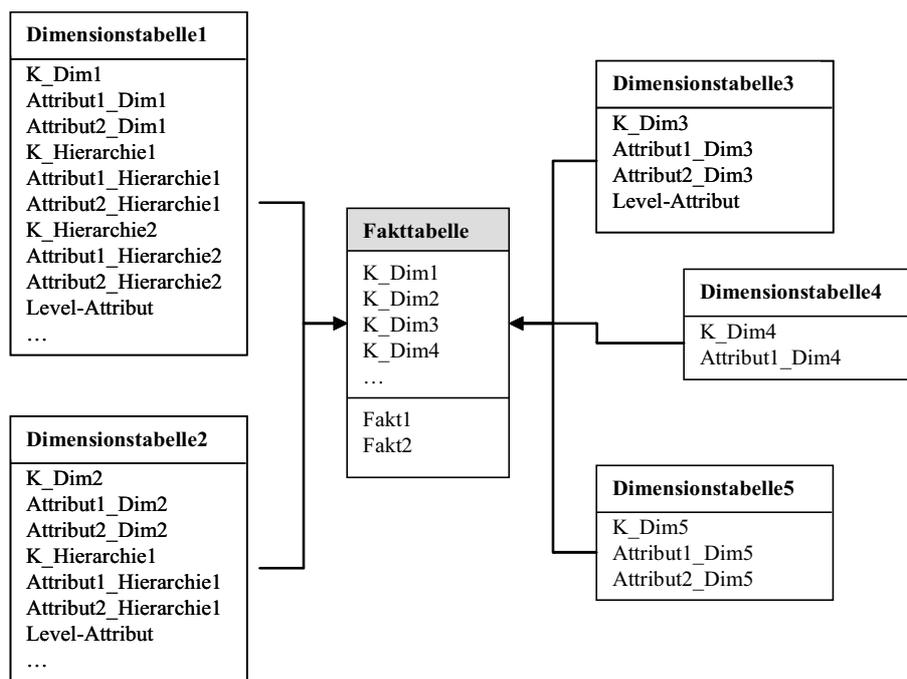


Abb. 2.8: Das klassische Star Schema

Quelle: In Anlehnung an: Hahne (1999), S. 153; Bauer/Günzel (2001), S. 201

Die Dimensionstabellen besitzen einen beschreibenden Charakter, indem sie die Geschäftsdimensionen der Faktdaten sowie ihre möglichen Hierarchieebenen nachbilden (vgl. Holthuis (1999), S. 196). Weiterhin können Aggregate der Hierarchieebenen in der Fakt-tabelle gespeichert werden. Dies setzt voraus, dass die verschiedenen Hierarchieebenen der Dimensionstabellen unterscheidbar sein müssen, da sonst falsche Aggregate durch Mehrfachzählungen von Einträgen entstehen können. Um die Hierarchieebenen der jeweiligen Dimensionstabelle beschreiben zu können, werden sogenannte *level*-Attribute eingefügt, die bei jeder Anfrage an das DWH angegeben werden und die jeweilige Hierarchiestufe oder -ebene beschreiben (vgl. Hahne (1999), S. 154). Eine Möglichkeit, um auf die level-Attribute zu verzichten, bietet der Modellierungsansatz nach dem Fact Constellation Schema.

Das *Fact Constellation* Schema ist eine Erweiterung des klassischen Star Schemas und wird als Spezialfall einer Galaxie bezeichnet (vgl. Bauer/Günzel (2001), S. 205). Die Aggregate werden nicht wie beim klassischen Star Schema in den Fakttabellen gehalten, stattdessen wird für jede Aggregation eine eigene Fakttabelle erstellt, wodurch eine Vielzahl von Fakttabellen entsteht. Dadurch werden keine level-Attribute benötigt. Eine weitere Alternative, um auf die level-Attribute des klassischen Star Schemas verzichten zu können, bietet das Snowflake Schema.

Das *Snowflake* Schema (s. Abb. 2.10, S. 24) besteht aus normalisierten bzw. partitionierten Dimensionstabellen.

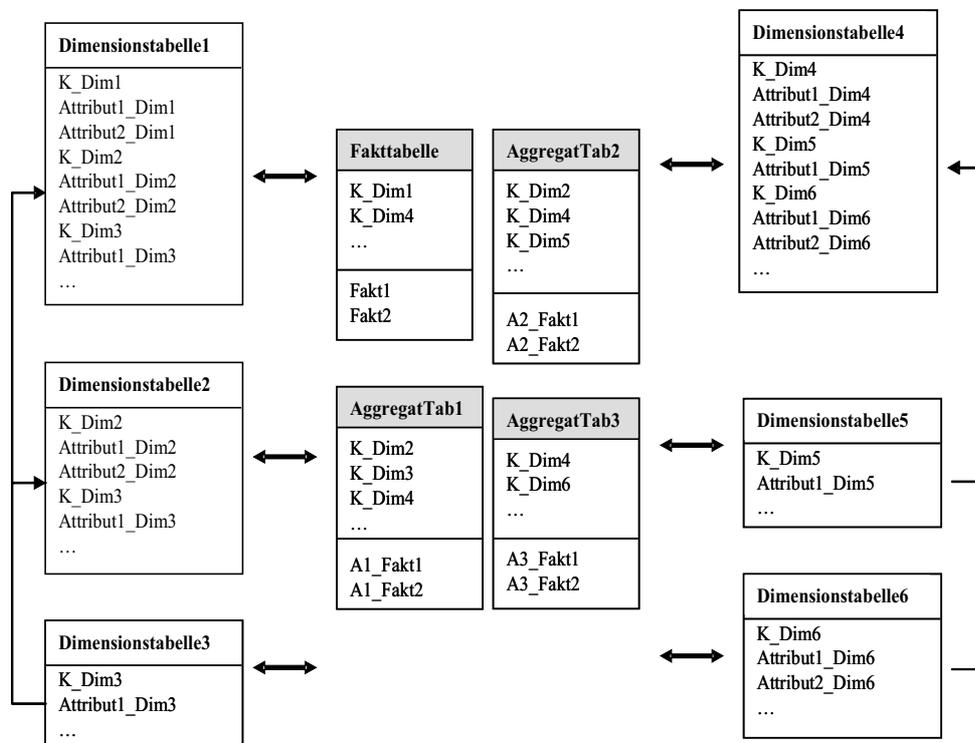


Abb. 2.9: Das Snowflake Schema

Quelle: In Anlehnung an: Hahne (1999), S. 164
(Beziehungen sind übersichtshalber vereinfacht)

Die Normalisierung der Dimensionstabellen zielt auf die Partitionierung nach dem level-Attribut des klassischen Star Schemas ab (vgl. Hahne (1999), S. 161). Für jede Hierarchieebene existiert somit eine Dimensionstabelle mit den eigenen und den Attributwerten der jeweils unteren Ebenen. Diese Dimensionstabellen werden als flache Hierarchietabellen verstanden. Weiterhin sind Aggregate wie beim Fact Constellation Schema in eigenen Fakttabellen gespeichert. Für jede Aggregation verschiedener Dimensionen existiert eine Fakttabelle. Durch die Anordnung der unterschiedlichen Fakt- und Dimensionstabellen erhält man die Struktur einer Schneeflocke.

Für die logische Modellierung von Data Warehouses auf multidimensionaler Basis existiert bis jetzt noch kein anerkanntes Modell. Es wird insbesondere für die Data Mart Entwicklung benötigt, da dort multidimensionale Datenbanken zugrunde liegen (vgl. Kemper et al. (2004), S. 36). Um den Rahmen dieser Diplomarbeit nicht zu sprengen, wird auf diese Ansätze nicht weiter eingegangen, da sie keine Verwendung zur Entwicklung des DWH Konzeptes finden.

Die Erläuterung des Daten-Layers waren nötig, da Konzepte und Methoden zur Integration von BI-Komponenten und der abschließenden Konzeption eines DWH zur Projekt- und Ressourcenplanung im weiteren Verlauf der vorliegenden Arbeit verwendet werden. Durch die Bereitstellung der Daten in der erwähnten multidimensionalen Struktur ist es Entscheidern mit Hilfe von Front-End-Werkzeugen oder sogenannten MSS möglich, Informationen zu generieren. Anschließend können diese Informationen mit interaktiven Navigations- und Analysemöglichkeiten präsentiert, gespeichert sowie verteilt werden (vgl. Bauer/Günzel (2001), S. 63). Diese MSS werden vom Logik-Layer bereitgestellt, welcher im Folgenden beschrieben wird.

2.4 Logik-Layer

Dieser Abschnitt widmet sich dem Logik-Layer des BI-Ordnungsrahmens aus Abschnitt 2.2. Diese Erläuterungen sind nötig, da im weiteren Verlauf der vorliegenden Arbeit die Verwendung des Logik-Layers zur Entscheidungsunterstützung an einem praktischen Beispiel gezeigt wird. Dafür werden MSS des Logik-Layers für die Integration von BI zur Projekt- und Ressourcenplanung eingesetzt. Dies soll zu einer Verbesserung der Entscheidungsunterstützung für die täglichen Geschäftsprozesse bezüglich Zeit, Kosten und Qualität führen.

Im Logik-Layer werden die Informationen für den Entscheider generiert und analysiert sowie Systeme zur Speicherung und Verteilung dieser Informationen bereitgestellt. Die Informationsgenerierung beinhaltet die Entdeckung relevanter Zusammenhänge, Muster, Musterbrüche oder Diskontinuitäten aus den aufbereiteten Daten. Hierbei ist nach vorbestimmten Hypothesen oder weitestgehend hypothesenfrei¹¹ vorzugehen, um anschließend spezielle Analysen und Auswertungen darauf aufbauen zu können (vgl. Grothe/Gentsch (2000), S. 20). Analyse im engeren Sinne bezeichnet die „systematische Untersuchung eines Gegenstands od. Sachverhalts hinsichtlich aller einzelnen Komponenten od. Faktoren, die ihn bestimmen“ (Drosdowski et al. (1982), S. 60). Bezogen auf BI heißt das, dass für die systematische Untersuchung der betriebswirt-

¹¹ Es liegen nur zum Teil Hypothesen vor, die durch das System in Form von Regeln oder Aussagen präsentiert werden (vgl. Grothe (1999), S. 5; Bissantz et al. (2000), S. 380).

schaftlichen Zusammenhänge Daten und Faktoren herangezogen werden müssen, um diese Zusammenhänge bestimmen zu können. Für diese Analysezwecke stellt der Logik-Layer MSS bzw. Analysesysteme bereit, die im späteren Verlauf dieses Abschnitts beschrieben werden. MSS generieren Informationen und geben dem Benutzer die Möglichkeit, Informationen in Form von Tabellen, Grafiken, Texten oder multimedialen Darstellungen¹² als Ergebnisse zu präsentieren. Anschließend können diese Analyseergebnisse in sogenannten Wissensmanagementsystemen abgespeichert und zur weiteren Verteilung bereitgestellt werden.

Wissen wird nach Definition (s. Abschnitt 2.2) zur Lösung von Problemen eingesetzt. *Wissensmanagement* (engl. *Knowledge Management*) wird das Konzept genannt, das die Summe aller technischen und organisatorischen Maßnahmen zur Unterstützung der Problemlösung darstellt (vgl. Hansen/Neumann (2001), S. 448). Systeme, die das betriebliche Wissensmanagement unterstützen, werden als *Wissensmanagementsysteme* bezeichnet (vgl. Kemper et al. (2004), S. 128). Wissensmanagementsysteme werden als Wissensspeicher im Unternehmen für berechnigte und interessierte Mitarbeiter eingesetzt. Da Wissensmanagementsysteme im weiteren Verlauf der vorliegenden Diplomarbeit keine Anwendung finden, wird dieses Konzept nicht weiter beschrieben. Stattdessen wird im folgenden Abschnitt auf die unterschiedlichen MSS eingegangen.

Management Support Systems (MSS) oder *Analysesysteme* unterstützen den Endanwender dabei, Entscheidungen zu treffen, indem sie die aufbereiteten dispositiven Daten aus dem Daten-Layer in einen anwendungsorientierten Kontext überführen, spezifisch aufbereiten und präsentieren. Dadurch werden die Daten inhaltlich erweitert und können als Informationen interpretiert werden, die für Entscheidungen genutzt werden können (vgl. Kemper et al. (2004), S. 79). Die MSS können in Anlehnung an KEMPER ET AL. in zwei große Gruppen eingeteilt werden, und zwar in die konzeptorientierten Systeme und die generischen Basissysteme (s. Abb. 3.3, S. 40).

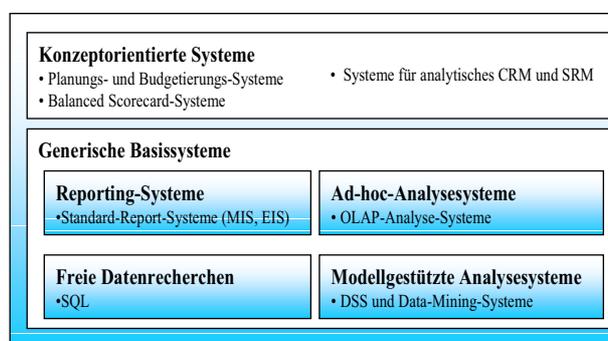


Abb. 2.10: Übersicht verschiedener MSS

Quelle: In Anlehnung an: Gluchowski/Kemper (2006), S. 16; Kemper et al. (2004), S. 84

¹² Berichte können um multimediale Darstellungen, d. h. Audio- und Videodateien erweitert werden (vgl. Bauer/Günzel (2001), S. 64).

2.4.1 Konzeptorientierte Systeme

Konzeptorientierte Systeme unterstützen gezielt Managementprozesse, indem sie betriebswirtschaftliche Konzepte wie die *Balanced Scorecard* (BSC) technologisch abbilden oder auf einen speziellen betriebswirtschaftlichen Anwendungsbereich ausgelegt sind, wie z. B. das *analytische Customer* (CRM) und *Supplier Relationship Management* (SRM) (vgl. Kemper et al. (2004), S. 84; Gluchowski/Kemper (2006), S. 17). Auf die Ausführung dieser Konzepte wird aufgrund des Umfangs in der vorliegenden Arbeit nicht näher eingegangen. Stattdessen wird im nachfolgenden Abschnitt auf das Konzept der Planung und Budgetierung eingegangen, da es im weiteren Verlauf dieser Diplomarbeit Anwendung findet.

Die Planung und Budgetierung gehören seit geraumer Zeit zum Aufgabengebiet des Managements und müssen auch auf Abteilungsebene durchgeführt werden. KEMPER ET AL. stellen dar, dass mit ihrer Hilfe kurz- oder langfristige Ziele definiert und innerhalb der Organisation koordiniert werden. Die Planung und Budgetierung dienen weiterhin der systematischen Auseinandersetzung mit zukünftigen Projekten und der Erfolgssicherung des Unternehmens. Dabei stellen Budgets einen in Geldeinheiten bewerteten Plan aller Einnahmen und Ausgaben der voraussichtlichen Aktivitäten für eine bestimmte Organisationseinheit und einen definierten Zeitraum dar. Ferner führen KEMPER ET AL. an, dass im Allgemeinen zwischen der strategischen, taktischen und operativen Planung unterschieden wird. Die strategische Planung bezieht sich auf die grundsätzliche Ausrichtung des Unternehmens und einen Zeitraum von fünf Jahren. Durch die taktische Planung hingegen werden konkrete Ziele für das Gesamtunternehmen erarbeitet sowie Ressourcen und Maßnahmen zur Zielerreichung in den nächsten zwei bis fünf Jahren festgelegt. Die operative Planung beschäftigt sich mit der Planung der wertschöpfenden Geschäftsprozesse und konzentriert sich auf einen Jahreszeitraum (vgl. Kemper et al. (2004), S. 119 f.). Bei der operativen Planung werden Plan- und Istkosten zu den jeweiligen Projekten während der Geschäftsprozesse verbucht. Die Planwerte werden im Vorhinein festgelegt und über einen definierten Zeitraum für die unterschiedlichen Projekte, Aktivitäten etc. eingeplant. Istwerte bilden den tatsächlichen Kostenstand der aktuellen Projekte ab. Durch das System sind daraufhin Plan- und Ist-Vergleiche zwischen den Kosten darstellbar. An diesen können Meilensteine¹³ beurteilt werden, indem z. B. Fragestellungen wie „Ist das Projekt bereits zu teuer?“ beantwortet und beurteilt werden.

¹³ Meilensteine bilden den Abschlusspunkt jeder Phase im Projekt und können als definierte termingebundene Sachergebnisse verstanden werden (vgl. Litke (2004), S. 29).

Zu weiteren Planungsaufgaben gehören Ablaufkoordination der einzelnen Projekte, die Mitarbeiterkapazitätenplanung sowie die Ressourcenplanung von Soft- und Hardware. Zur taktischen Planung können Systeme über einen größeren Zeitraum die übersichtliche Planung übernehmen und unterstützen. Teilweise können diese Systeme zu verschiedenen Arten der Unternehmenssimulationen eingesetzt werden. Hierunter fallen What-If-Abfragen. Bei diesen werden die Auswirkungen eines Ereignisses oder einer Maßnahme simuliert und untersucht (vgl. Kemper et al. (2004), S. 122). Im Allgemeinen erfordert die Unterstützung von Planungs- und Budgetierungsprozessen eine Bereitstellung von MSS als Planungswerkzeuge für die Projekte, Ressourcen, Datenverteilung, Forecast-Analyse und Simulation (vgl. Bange (2006), S. S.72).

2.4.2 Generische Basissysteme

Eine weitere Unterteilung der MSS ist nach KEMPER ET AL. in die generischen Basissysteme möglich. Diese fassen Systeme zusammen, die als eigenständige Anwendungen in eine BI-Infrastruktur integriert werden können (vgl. Kemper et al. (2004), S. 83). Darunter fallen die *freie Datenrecherche*¹⁴, *Reporting-Systeme*, *Ad-hoc-Analysesysteme* und *modellgestützte Analysesysteme*¹⁵. Im Anschluss wird auf Reporting- und Ad-hoc-Analysesysteme eingegangen, da diese Anwendungen im weiteren Verlauf der vorliegenden Diplomarbeit Verwendung finden.

Unter *Reporting-Systeme* – im Deutschen Berichts-Systeme – werden all die Systeme zusammengefasst, die die Erstellung und Bereitstellung von Reports im Sinne des betrieblichen Reporting (dt. Berichtswesen) ermöglichen (vgl. Kemper et al. (2004), S. 110). Darunter fallen u. a. die schon erwähnten MIS und EIS. Reporting-Systeme erlauben meist den Zugriff auf unterschiedliche Datenquellen und sind in der Lage diese Daten als Reports auszugeben (vgl. Mischke (2006), S, 112). Der grundsätzliche Zweck von Reports besteht darin, Voraussetzungen dafür zu schaffen, dass die Entscheider den vorliegenden Datenbestand, also die wichtigsten Fakten und Zusammenhänge, schneller erkennen und somit effektiv Entscheidungen treffen können. Es gibt grundsätzlich zwei Arten von Reports:

¹⁴ Systeme für eine freie Datenrecherche sehen die Benutzung einer Datenmanipulationssprache wie die Structured Query Language (SQL) vor.

¹⁵ Modellgestützte Analyseysteme bedienen sich bestimmter Algorithmen oder Regeln, um bestimmte Zusammenhänge oder Muster in den Daten zu erkennen, darunter fallen u. a. DSS oder Data-Mining-Systeme.

- *Standard-Reports* sind einmalig erstellte Reports, die nicht verändert werden können. Sie werden entweder in bestehende Anwendungen für selbständige Auswertungen der Mitarbeiter integriert (s. Abschnitt 3.3.1) oder werden durch Reporting-Systeme in periodischen oder aperiodischen Abständen generiert und verteilt, wie z. B. bei Frühwarnsystemen, wenn Grenzwerte überschritten werden (vgl. Kemper et al. (2004), S. 111 f.).
- *Ad-hoc-Reports* sind spontane Reports, die jederzeit durch den Entscheider oder Reportersteller generiert werden können. Systeme für Ad-hoc-Reports bieten dem Endanwender viele Interaktionsmöglichkeiten bei der Reporterstellung. Solche Reports werden meist mit Ad-hoc-Analysesystemen oder OLAP-Systemen erstellt und verteilt.

Ad-hoc-Analysesysteme auch OLAP-Systeme genannt, bieten dem Endanwender die Möglichkeit, selbständig, schnell und mit geringem Aufwand sowohl individuelle Ad-hoc-Analysen als auch komplexe betriebswirtschaftliche Analysen durchführen zu können (vgl. Chamoni/Gluchowski (1997), S. 25). Sie gehen auf die Einführung des Begriffs OLAP durch CODD ET AL. zurück (vgl. Codd et al. (1993)).

Sie definierten OLAP als „the name given to the dynamic enterprise analysis required to create, manipulate, animate and synthesize information from [...] „Enterprise Data Models“ [...]. This includes the ability to discern new or unanticipated relationships between variables, the ability to identify the parameters necessary to handle large amounts of data, to create an unlimited number of dimensions (consolidation paths) and to specify cross-dimensional conditions and expressions“ (Codd et al. (1993), S. 8 f.).

OLAP ist demnach ein Analyseansatz, um Informationen generieren, manipulieren, animieren und synthetisieren zu können. Dies ermöglicht das Erkennen von neuen oder unbekanntem Beziehungen zwischen Variablen, das Aufzeigen von relevanten Parametern in großen Datenbeständen, das Erstellen von zahlreichen Dimensionshierarchien und das Angeben von mehrfachhierarchischen Bedingungen und Ausdrücken.

Für OLAP-Systeme existiert seit 1995 ein anerkannter Anforderungskatalog aus fünf Kernpunkten. Er wurde von N. PENDSE und R. CREETH im Zuge des OLAP-Reports unter dem Namen *Fast¹⁶ Analysis¹⁷ of Shared¹⁸ Multidimensional¹⁹ Information²⁰* (FASMI) veröffentlicht (vgl. Pendse/Creeth (1995), S. 50 ff.; Pendse (2006)). Aufgrund dieser Anforderungen an OLAP-Systeme steht dem Endanwender ein schneller, direkter und

¹⁶ *Fast* sieht schnelle Antwortzeiten bei Abfragen an die dispositive Datenbank vor.

¹⁷ *Analysis* fordert anwenderfreundliche, intuitive Analysen, Berechnungen und Präsentationsformen.

¹⁸ *Shared* verlangt nach Zugriffsschutz und Mehrbenutzerbetrieb der dispositiven Datenbestände.

¹⁹ *Multidimensional* soll die semantische Sicht auf die Daten für den Endanwender darstellen.

²⁰ Die *Informationen* sollen nicht von Datenmenge oder -herkunft beeinflusst werden.

interaktiver Zugriff auf analysegerechte dispositive Daten zur Verfügung. Für die Analyse in multidimensionalen Datenräumen steht dem Endanwender zusätzlich eine Vielzahl von Operationen bereit.

Im Weiteren werden die gängigsten multidimensionalen Operationen in Anlehnung an BAUER und GÜNZEL verkürzt beschrieben (vgl. Bauer/Günzel (2001), S. 105 f.).

- *Pivotierung/ Rotation*: Der Datenwürfel wird für die aktuelle Analyse in eine geeignete Position bewegt.
- *Drill-down & Roll-up*: Durch das Drill-down wird die Hierarchieebene detaillierter angezeigt. Roll-up oder auch Drill-up ist die inverse Operation, bei der die Daten auf die höhere Ebene zusammengefasst werden.
- *Drill-through & Drill-across*: Die Drill-through-Operation ermöglicht das weitere Herunterbrechen der letzten Hierarchieebene durch Übergehen in eine andere, meist detailliertere Datenquelle. Drill-across ermöglicht das Wechseln zwischen zwei Datenwürfel, die die gleichen Dimensionshierarchien benutzen.
- *Slice & Dice*: Mit der Slice-Operation wird das Zerschneiden der Datenwürfel in anwenderindividuelle Sichten oder Ebenen bezeichnet. Dice bezieht sich auf das Herunterbrechen der Würfelteile für konkrete Kombinationen von Dimensionselementen.

Durch diese Operationen hat der Endanwender uneingeschränkte Möglichkeiten, die Daten aus verschiedenen Blickwinkeln und unter Einfluss verschiedener Faktoren zu analysieren.

Heutzutage wird angestrebt, den Zugriff auf verschiedene MSS und die gespeicherten Informations- und Wissensbestände über einen Präsentations-Layer zu gewährleisten. In neuesten Forschungsgebieten und zum Teil in der Praxis werden daher sogenannte Business Intelligence Portale realisiert.

2.5 Präsentations-Layer

Die oberste Schicht des BI-Ordnungsrahmens aus Abschnitt 2.2 stellt den Informationszugriff dar. Der Zweck eines Präsentations-Layers besteht darin, dem Benutzer einen geeigneten und arbeitsplatzspezifischen Zugriff auf alle nötigen Informationen und Funktionalitäten zu gewährleisten (vgl. Gluchowski/Kemper (2006), S. 17). Dazu werden in der Forschung und teilweise in der Praxis *Business Intelligence Portale* (BI-Portale) angestrebt, die einen direkten Zugriff auf konsistente, heterogene Daten und alle verfügbaren BI-Komponenten bieten. BI-Portale integrieren die verschiedenen Komponenten aus den Bereichen der konzeptorientierten Systeme, der generischen Basesysteme und des unternehmensweiten Wissensmanagements unter einer einheitlichen Benutzeroberfläche und Benutzerführung (vgl. Gluchowski/Kemper (2006), S. 17; Kemper et al. (2004), S. 136).

2.6 Resümee

In den vorangegangenen Abschnitten wurde zunächst die Entwicklung der MSS beschrieben, um anschließend den Begriff BI abzugrenzen und als integrierten, unternehmensspezifischen, IT-basierten Gesamtansatz zu definieren. Darauf aufbauend wurde ein BI-Ordnungsrahmen aus aktuellen wissenschaftlichen Erkenntnissen vorgestellt, der die gängigen IT-basierten BI-Komponenten zur unternehmensspezifischen Integration einer BI-Infrastruktur vereint. Diese grundlegenden Erläuterungen waren erforderlich, da im weiteren Verlauf der vorliegenden Diplomarbeit der BI-Ordnungsrahmen als Rahmenkonzept für den Einsatz von BI in der Praxis verwendet wird. Dadurch wird sichergestellt, dass sich die Integration der BI-Komponenten an ein wissenschaftlich etabliertes Konzept halten. Der BI-Ordnungsrahmen beinhaltet einen Daten-, Logik- und Präsentations-Layer.

Insgesamt lässt sich festhalten, dass die vorgestellten Methoden und Systeme des Daten- und Logik-Layers, Daten aufbereiten und Informationen generieren. Der Präsentations-Layer bietet darauf aufbauend den anwenderfreundlichen Zugriff auf die zur Verfügung stehende Informationsbasis, um daraus Wissen zur Problemlösung zu erlangen. Diese Erkenntnisse sollen im weiteren Verlauf der vorliegenden Diplomarbeit dazu verwendet werden, die Entscheidungsunterstützung durch die Erweiterung von Reporting- und Analysefunktionalitäten einer Projektmanagement-Anwendung zu verbessern. Dadurch sollen Kosten durch die Zeitersparnis während der täglichen Geschäftsabläufe eingespart und die Qualität der Planung gesteigert werden. Im folgenden Kapitel wird näher auf diesen Einsatz von BI zur Projekt- und Ressourcenplanung eingegangen.

3 Business Intelligence zur Projekt- und Ressourcenplanung

Im vorangegangenen Kapitel wurden die verschiedenen Komponenten und Technologien, die im Zusammenhang mit Business Intelligence auftreten, vorgestellt und beschrieben. Der in 2.2 eingeführte BI-Ordnungsrahmen dient in diesem Kapitel als Rahmenkonzept zur Eingliederung der verwendeten BI-Komponenten. Um das Ziel der vorliegenden Arbeit, nämlich die Optimierung der Projekt- und Ressourcenplanung bezüglich der Entscheidungsunterstützung innerhalb der Digitalen Fabrik- und Fertigungsplanung, zu verfolgen, werden zur Optimierung BI-Komponenten eingesetzt.

Bevor die Umsetzung der Integration von BI-Komponenten erläutert wird, soll vorerst die Ausgangssituation der Digitalen Fabrik- und Fertigungsplanung und ihre Aufgabengebiete beschrieben werden. Anschließend wird die Projektmanagement-Anwendung PMM vorgestellt und erläutert, welche durch BI-Komponenten erweitert werden soll. Der letzte Teil des Kapitels widmet sich der Optimierung der Projekt- und Ressourcenplanung bezüglich der Entscheidungsunterstützung innerhalb der erwähnten Abteilung durch die Integration und Verwendung von BI-Komponenten in PMM (s. Kapitel 3.3). Diese Erweiterung von PMM durch BI wurde im Zusammenhang mit der vorliegenden Diplomarbeit entworfen und implementiert. Dabei wurden die gegebenen Voraussetzungen innerhalb der betrachteten Abteilung berücksichtigt. Die bereits integrierten Komponenten werden genauer betrachtet. Es wird auf den PMM-Daten-Layer und auf den PMM-Präsentations- bzw. Logik-Layer eingegangen. Der Daten-Layer behandelt die Speicherung und Aufbereitung der Daten für die speziellen Analyseverfahren. Der Präsentations- bzw. Logik-Layer widmet sich der Informationsgenerierung durch Standard-Reports und Ad-hoc-Analysen sowie dem benutzerfreundlichen Zugriff zur Entscheidungsunterstützung.

Zuvor sei allerdings noch zu sagen, dass es sich bei der vorliegenden BI-Lösung um eine abteilungsinterne Lösung handelt. Eine unternehmensweite Betrachtung würde den Umfang der vorliegenden Diplomarbeit aufgrund der Größe des betrachteten Automobilherstellers überschreiten. Außerdem lassen die Voraussetzungen der betrachteten IT-Abteilung keine unternehmensweite Betrachtung zu. Trotz der abteilungsinternen Betrachtung halten sich die folgenden Ausführungen an die BI-Definition als integrierten, unternehmensspezifischen IT-basierten Gesamtansatz und den vorgestellten Ordnungsrahmen.

3.1 Ausgangssituation

Die Ausgangssituation wird in Form einer Ist-Aufnahme dargestellt. Unter Ist-Aufnahme wird die quantitative und qualitative Erfassung des Ist-Zustandes eines geschlossenen Systems unter Beachtung des zu untersuchenden Tatbestandes verstanden. Sie bildet zusammen mit der Potenzialanalyse (s. Abschnitt 4.1) die Ist-Analyse (vgl. Krallmann et al. (1999), S. 60).

Die Ist-Aufnahme ist von großer Bedeutung bei einer Systemanalyse im Unternehmen. Deren Erhebungsmethoden unterteilen sich nach KRALLMANN ET AL. in folgende Methoden, wobei zwischen Primär- und Sekundärerhebung unterschieden wird (vgl. Krallmann et al. (1999), S. 60):

Primärerhebung

- *Interviewmethode*: Die Mitarbeiter werden persönlich befragt. Die Methode differenziert zum einen zwischen Einzel-, Gruppenbefragung und Konferenz. Zum anderen sind standardisierte²¹, nicht-standardisierte²², offene²³ und verdeckte²⁴ Befragung möglich.
- *Fragebogenmethode*: Ein vorgefertigter Fragebogen wird an die Mitarbeiter ausgehändigt. Dabei ist zwischen differenziertem²⁵ und standardisiertem²⁶ Fragebogen zu unterscheiden.
- *Berichtsmethode*: Die Mitarbeiter verfassen einen eigenständigen Bericht über ihre Arbeitsgebiete.
- *Beobachtungsmethode*: Die Arbeitsabläufe, Verhaltensweisen der Mitarbeiter und jede Art von Zuständen werden vom Untersuchenden aufgenommen.

Sekundärerhebung

- *Inventurmethode*: Sie wird ausschließlich ohne Beteiligung der Mitarbeiter durchgeführt. Die Methode ist eine sog. Dokumentenanalyse, um Informationen über die Arbeitsabläufe und Strukturen des Unternehmens zu erhalten.

Im Weiteren wird auf die Vorgehensweise zur Ist-Aufnahme der Ausgangssituation und deren Beschreibung eingegangen.

²¹ Fragen werden vorher festgehalten und nicht verändert.

²² Fragen können in unterschiedlicher Reihenfolge gestellt werden, Zusatzfragen sind möglich.

²³ Dem Beteiligten ist bewusst, dass er befragt wird.

²⁴ Dem Beteiligten ist nicht bewusst, dass er befragt wird.

²⁵ Der Fragebogen wird auf eine bestimmte Gruppe von Mitarbeitern zugeschnitten

²⁶ Jeder Mitarbeiter bekommt den selben Fragebogen

3.1.1 Vorgehensweise zur Ist-Aufnahme

Zu Beginn der Ist-Aufnahme sind Beobachtungsmethoden und vorwiegend die Inventurmethode zum Einsatz gekommen, d.h. es wurden Dokumente und andere schriftliche Informationsquellen herangezogen, um die Arbeitsgebiete und -abläufe der betrachteten Abteilung zu verstehen. Durch die Dokumentenanalyse haben Dokumente wie Unterlagen zu Mitarbeiterschulungen und diverse Informationsquellen der Unterabteilungen im Intranet zum Verständnis der Arbeitsgebiete der Abteilung geführt. Zusätzlich hat das Handbuch zum System-Entwicklungs-Prozess zum weiteren Verständnis der Geschäftsprozesse beigetragen, da es die Vorgehensweise zur Einführung von Informationssystemen im betrachteten Automobilkonzern vorschreibt.

Die Systeme „Time“ und „Peanuts“ wurden untersucht, um die bisher verwendeten Informationssysteme zur Unterstützung der täglichen Abläufe innerhalb der betrachteten Abteilung kennen zu lernen. Zudem wurden die Systeme konkret im Hinblick auf die Entscheidungsunterstützung analysiert.

Die Erkenntnisse aus der Inventurmethode und den Beobachtungen sind jedoch nicht ausreichend gewesen, um die geforderten Anforderungen zur Verbesserung der Entscheidungsunterstützung darzulegen. Deswegen kam eine weitere Methode der Primärerhebung zur Ist-Aufnahme zum Einsatz. Die Mitarbeiter wurden durch die Interviewmethode offen, aber nicht standardisiert befragt. Daher wurden kontinuierlich in Wochenabständen Konferenzen bzw. Meetings mit dem Abteilungs- und den Projektleitern einberufen, um die Anforderungen an die spätere Projektmanagement-Anwendung und die geforderten Analysemöglichkeiten zu erörtern. Zudem wurde der Fortschritt der Entwicklung der Projektmanagement-Anwendung und ihrer BI-Komponenten präsentiert.

Für die konkrete Integration der BI-Komponenten und zur Erfüllung der geforderten Analysemöglichkeiten zur Projekt- und Ressourcenplanung wurde wiederum eine Dokumentenanalyse durchgeführt. Dabei kamen hauptsächlich das PMM Lastenheft (Henke (2006)), die Anwendungshandbücher der BI-Komponenten (Business Objects (2004a-e), Peck (2003a), Peck (2003b), McAmis (2004)) und das Handbuch zur BI-Strategie des betrachteten Automobilherstellers zum Einsatz. Zum einen führte dies zum Verständnis der zugrunde liegenden Datenbankstruktur sowie deren Tabellen und Beziehungen. Zum anderen wurden, ergänzend zu den Mitarbeitergesprächen, die Anforderungen an die Projektmanagement-Anwendung deutlich. Dadurch konnten die BI-Komponenten unter Berücksichtigung der Voraussetzungen des Automobilherstellers gezielt in die Anwendung integriert werden.

Im Kommenden wird die Ausgangssituation und die Arbeitsgebiete der Digitalen Fabrik- und Fertigungsplanung dargelegt und darauf aufbauend die Projektmanagement-Anwendung „PMM“ beschrieben.

3.1.2 Darstellung der Ausgangssituation

Die *Digitale Fabrik- und Fertigungsplanung* ist eine Unterabteilung des IT-Bereichs eines großen Automobilherstellers. Die Aufgabe des IT-Bereichs ist die weltweite Sicherstellung der bedarfsgerechten, wirtschaftlichen Konzeption, Entwicklung und Bereitstellung von Informationssystemen (IS) für sowohl externe Auftraggeber als auch für den Produktprozess (PP) in allen internen Bereichen des Konzerns.

In der *Digitalen Fabrik- und Fertigungsplanung* werden konzernweit Methoden und Systeme zur Unterstützung der Planungsbereiche Digitale Fabrik, Fabrikplanung, Fertigungsplanung und -simulation sowie für den Werkzeug-, Vorrichtung- und Versuchsbau zur Verfügung gestellt. Dabei erstrecken sich die Aufgabengebiete von der Prozessberatung über System- und Anwenderservice für alle Rechner und Systemplattformen bis hin zur Konzeption, Umsetzung und Einführung von IS-Lösungen und marktverfügbaren Standardlösungen. Hierbei ist sicherzustellen, dass jedes System einem bestimmten Prozess, d. h. einem bestimmten Aufgabenbereich bzw. einer Unterabteilung im Konzern im PP zugeordnet wird. Um diese Verantwortung überblicken und strategisch planen zu können, wurde ein sogenannter Bebauungsplan (s. Abb. A.1, S. 128) aufwendig zusammengestellt, der für jeden Prozess die derzeit verwendete und zukünftig benötigte Software auflistet. Dadurch ist ein sogenannter Ist-Soll-Vergleich möglich.

Weiterhin ist die betrachtete Abteilung für die Ressourcenplanung von Soft- und Hardware verantwortlich, worunter die Standardisierung der Software und das Lizenz- und Releasemanagement fällt. Hierbei ist festzuhalten, welche Art von Lizenzen für welche Software vorhanden sind und beschafft werden müssen. Zudem wird festgehalten, wie viele Anwender die Lizenz verwenden und welche Anschaffungs- und Wartungskosten für die jeweilige Software bei einem bestimmten Prozess im PP anfallen. Für jede Hardware (Computer, Drucker usw.) in dem Aufgabenbereich der IT-Abteilung wird der Standort festgehalten.

Zusätzlich zur Ressourcen- und Finanzplanung von Soft- und Hardware gehört die Planung und Durchführung von sowohl laufenden als auch zukünftigen Projekten (z. B. Entwicklung einer Software) der eigenen Abteilung zu den weiteren Aufgabengebieten der betrachteten IT-Abteilung. Diese Projektplanung beinhaltet auch die Mitarbeiterkapazitätenplanung. Der Projektleiter legt hierbei für jeden internen und externen Mit-

arbeiter die Kapazitäten in Form eines Zeitanteils für das laufende und kommende Jahr auf geplante Aktivitäten fest. Darunter fallen alle Tätigkeitsfelder der Mitarbeiter wie „Prozessberatung durchführen“, „Softwarecode programmieren“ oder „Dokumentation erstellen“. Um im Weiteren die Personalkosten und den Aufwand der Projekte bestimmen zu können, ist es in diesem Zusammenhang wichtig, die Arbeits- und Projektzeiten der Mitarbeiter nachvollziehen zu können.

Dem Abteilungs- bzw. Projektleiter muss infolgedessen ein genauer Überblick über alle laufenden und einmaligen Kosten von Projekten, die dazu verwendeten Mitarbeiter-, Software- und Hardwareressourcen zur Verfügung stehen, um eine optimale Planung und Budgetierung der Projekte und Ressourcen durchführen zu können.

Vor Einführung der Projektmanagement-Anwendung PMM²⁷ wurde von den Abteilungs- und Projektleitern die strategische Planung und Kontrolle ihrer Projekte, Kapazitäten und Ressourcen aufwendig mit Hilfe von mehreren Anwendungen durchgeführt. Die Aufstellung des Bebauungsplans und die strategische Forecast-Planung über die Lizenzenverwaltung sowie über die Anschaffungs- und Wartungskosten der Software wurde zeitintensiv mit Microsoft Excel (MS Excel) erstellt und gepflegt (s. Abb. A.1, Abb. A.2, S. 128). Es wird eine taktische Planung von den Abteilungs- und Projektleitern angestrebt, indem die Forecast-Planung für eine Zeitspanne von fünf Jahren erstellt wird. Damit sind die Kostenplanungen und Budgetierungen der Ressourcen für die nächsten Jahre möglich.

Für die Entscheidungsunterstützung zur Ressourcenplanung könnten lediglich die Excel-Tabellen auf Papier gedruckt, per Email versendet oder in verschiedene Formate, die Excel bietet, exportiert werden. Es war weder die Ad-hoc-Abfrage verschiedener Daten möglich noch konnten individuelle Reports erstellt werden.

Für die Projekt- und Kapazitätenplanung wurden mehrere Datenbanken und Systeme verwendet. Ein System zur Finanz- und Kapazitätenplanung, ein anderes für die Zeiterfassung der Mitarbeiter. Das System „Peanuts“ wurde zur Finanz- und Kapazitätenplanung der Projekt- und Personalkosten eingesetzt (s. Abb. B.1, S. 129). Diese Anwendung stellte ein sukzessiv gewachsenes, relationales Datenbankmanagementsystem auf Microsoft Access-Basis dar. Bisher wurden die Daten aus dem anderen System, das Zeiterfassungssystem „Time“, manuell in die Finanz- und Kapazitätenplanung übertragen und dort weiterverarbeitet. Das Zeiterfassungstool „Time“ wurde von jedem Mitarbeiter benutzt, um die Anwesenheitszeiten zu erfassen und eine Übersicht über diese zu erlangen (s. Abb. B.2, S. 129). Es ist ebenfalls ein relationales Datenbankmanagementsystem auf Microsoft Access-Basis mit mehreren Access-Datenbanken.

²⁷ Die Anwendung wird an späterer Stelle vorgestellt (s. Abschnitt 3.2.1).

Exportmechanismen von Reports oder diversen Übersichten waren nur innerhalb der Microsoft Office-Familie möglich.

Für die Entscheidungsunterstützung zur Projekt- und Kapazitätenplanung konnten lediglich statische Reports generiert werden, d. h. der Benutzer konnte keine individuellen Reports bzw. Abfragen generieren. Dadurch wurde eine sehr eingeschränkte Analyse zugelassen.

Anhand des Ablaufdiagramms²⁸ in Abb. 3.1 (s. 36) wird erneut der enorme Aufwand für die Reporterstellung dargestellt. Bevor ein Report erstellt werden konnte, musste der Verwendungszweck des Reports festgestellt werden. Je nach Verwendungszweck mussten unterschiedliche Systeme, wie oben beschrieben, verwendet werden. Für die Projekt- und Kapazitätenplanung mussten sogar zwei unterschiedliche Systeme verwendet werden, um die aktuellen Projektzeiten und Personalkosten zu erhalten. Denn die Projektzeiten der Mitarbeiter mussten manuell in die Finanz- und Kapazitätenplanung „Peanuts“ übertragen werden. Die Erstellung der Reports war zum einen lediglich innerhalb MS Excel möglich, worunter zusätzlich der Export in die bereitgestellten Formate fällt. Zum anderen konnte der Report weder mit individuellen Anpassungen noch mit einheitlichen Formaten auf Papier gedruckt werden.

Die Beschreibungen der Ausgangssituation verdeutlichen, dass eine Optimierung der Geschäftsprozesse von Vorteil ist, um Zeit und Kosten einzusparen. Denn die aufwändige Nutzung von mehreren Anwendungen und Datenbanken hatte bisher einen enormen Zeitverlust durch das Erfassen und Pflegen des Datenbestandes zur Folge. Ebenso hat die Erstellung von Reports einen enormen Zeitaufwand beansprucht. Zudem konnten die Reports nicht angepasst oder individuell bearbeitet werden. Dadurch hatten die Entscheider eine eingeschränkte Sicht auf die vorhandenen Daten. Somit ist die Entscheidungsunterstützung nicht effizient. Weiterhin entsprechen die Anwendungen nicht den Anforderungen der Abteilung und wurden unzureichend benutzt. Diese Einflussfaktoren lassen die Planungsqualität sinken und Kosten werden durch verlängerte Projektzeiten oder übermäßig bezahlte Fremdfirmenmitarbeiter erhöht.

²⁸ Als Ablaufdiagramm wird die grafische Darstellung einer logischen Abfolge von Entwicklungs- und Abfolgeprozessen bezeichnet (vgl. Koschnick (1996), S. 2).

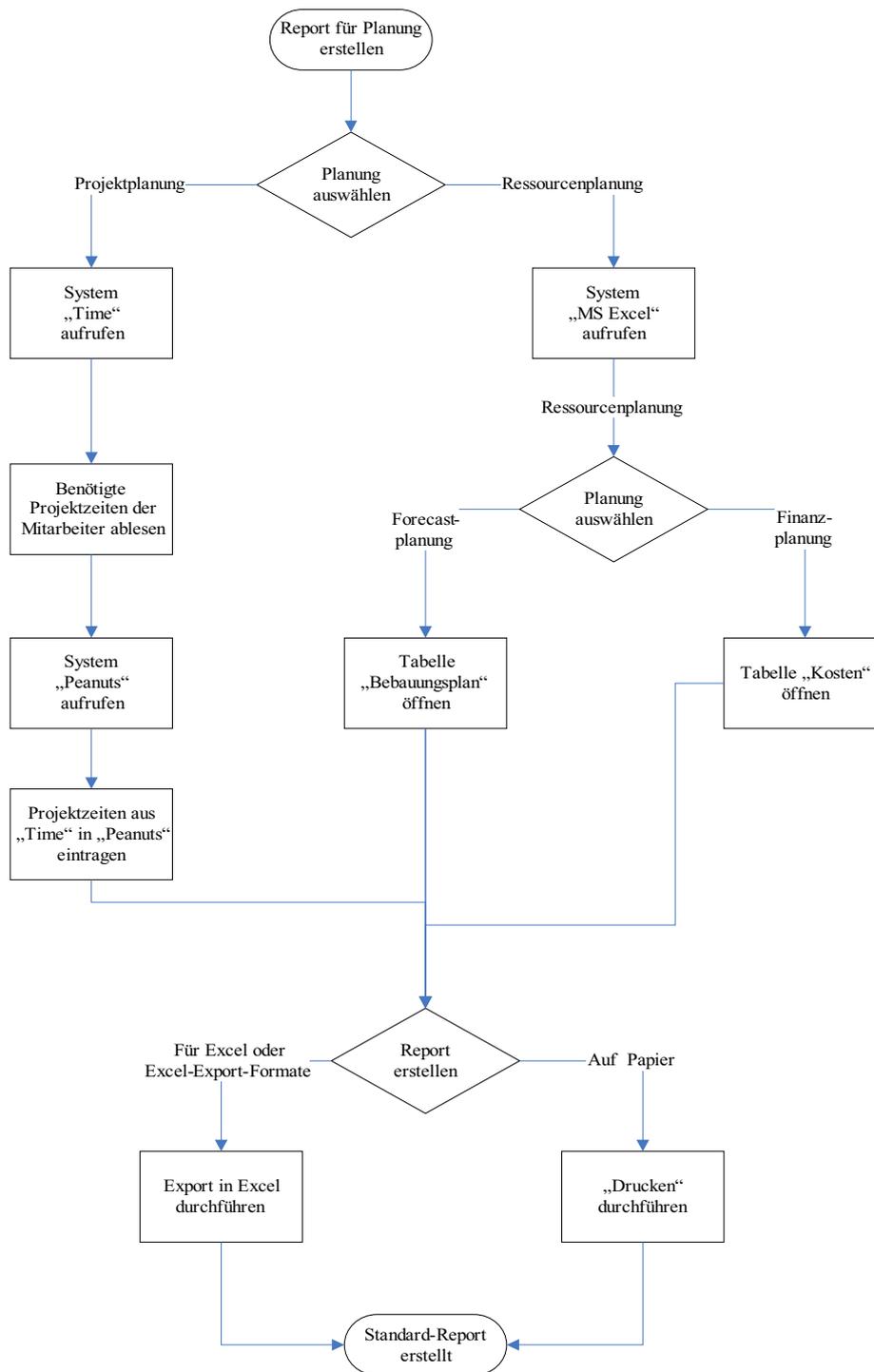


Abb. 3.1: Ablaufdiagramm – Ausgangssituation Reporterstellung

Die Optimierung des Dreigespanns Kosten, Zeit und Qualität war ausschlaggebend für die Neuentwicklung der *Projektmanagement-Anwendung* „PMM“ zur Projekt-, Kapazitäten- und Ressourcenplanung. Diese Anwendung soll alle Kernfunktionalitäten der ursprünglichen Datenbanken und Anwendungen übernehmen und durch eine benutzerfreundliche Oberfläche zur Verfügung stellen. Zusätzlich sollen die Funktionalitäten zur Entscheidungsunterstützung verbessert werden.

3.2 Projektmanagement-Anwendung PMM

Unter dem Begriff Projektmanagement werden nach DIN 69901 die Aufgaben, Organisation, Techniken und Mittel der Führung zur Abwicklung eines Projekts verstanden (vgl. DIN (1987)). Damit unterstützt das Projektmanagement die operative Planung im Unternehmen. Projektmanagement-Anwendungen werden eingesetzt, um diese Vorgänge zu unterstützen.

Die Projektmanagement-Anwendung PMM der Digitalen Fabrik- und Fertigungsplanung kann weiterhin in die Gruppe der MSS eingeordnet werden, da es Teilfunktionalitäten der konzeptorientierten MSS zur Planung und Budgetierung mit Funktionalitäten von generischen Basissystemen für die Reporterstellung und Ad-hoc-Analysen verknüpft. PMM wurde entwickelt, um die Projekt-, Kapazitäten- und Ressourcenplanung zu optimieren. Zudem sollte der Zugriff von mehreren Anwendungen und Datenbanken auf eine gemeinsame Datenbasis und Anwendungsumgebung reduziert werden. PMM ist vom Entwurf bis hin zur Implementierung und Wartung eine interne Neuentwicklung des betrachteten Automobilherstellers. Von der Verwendung und Einführung einer Standard-Lösung wurde abgesehen, da die Kosten für die Eigenentwicklung niedriger ausfallen und durch die Neuentwicklung die Bedürfnisse und Anforderungen der Abteilung direkt in die Anwendung implementiert und angepasst werden können.

Die Anforderungen der Abteilung sahen vor, dass PMM die Kernfunktionalitäten der ursprünglichen Anwendungen und Datenbanken zur Verfügung stellt und eine Möglichkeit zur aussagekräftigen dynamischen Analyse bieten sollte. Zu jedem Zeitpunkt sollen der Projektfortschritt und die verursachten Kosten nachvollzogen und analysiert werden können.

Im praktischen Teil dieser Diplomarbeit wurden Teilfunktionalitäten von PMM entwickelt und der Bereich „Berichtswesen“ und „Controlling“ unter den gegebenen Voraussetzungen der betrachteten Abteilung entworfen und implementiert. Darunter fallen die Integration von Standard-Reports und die dynamische Erstellung von Ad-hoc-Abfragen sowie Durchführung von Ad-hoc-Analysen mittels BI-Komponenten. Derzeit ist PMM als lauffähige Testversion für den Abteilungsleiter und die Projektleiter im Einsatz. Die Funktionalitäten von PMM und die BI-Komponenten werden im Weiteren behandelt.

3.2.1 Beschreibung der Anwendung

PMM ist eine mit Visual Basic .NET (VB.NET) programmierte Projektmanagement-Anwendung und soll von dem Abteilungsleiter und den Projektleitern der Digitalen Fabrik- und Fertigungsplanung für strategische Planungszwecke und Analysen genutzt werden. Zudem sollen alle Mitarbeiter der Projekte ihre Arbeitszeiten und Projektzeiten verwalten können.

PMM ist auf einer Client-Server-Architektur aufgebaut. Die Datenhaltung basiert auf einer transaktionsorientierten, relationalen Datenbank²⁹ auf dem Server. Damit sind die Daten für jeden berechtigten Benutzer über die Client-Schnittstelle verfügbar, wodurch ein Mehrbenutzerbetrieb gewährleistet werden kann. Eine genauere Beschreibung der Tabellen und ihrer Verwendung erfolgt an späterer Stelle und in Abschnitt 3.2.2. Das PMM-Front-End koordiniert die Verbindung zwischen Client, Server, den Projektmanagement-Funktionalitäten und den angebundenen BI-Komponenten (s. Abb. 3.2, S. 38). Die Abb. 3.2 (s. 38) zeigt die verschiedenen Schichten in Bezug auf PMM und seine BI-Komponenten in Anlehnung an den BI-Ordnungsrahmen aus Kapitel 2.2.

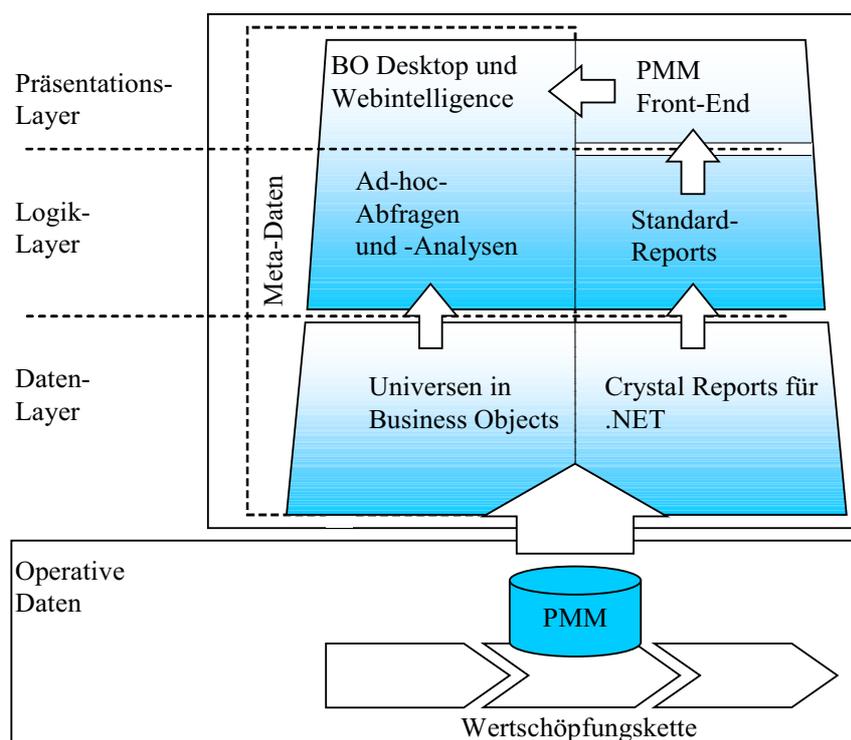


Abb. 3.2: Einordnung der BI-Komponenten von PMM

Der Präsentations-Layer umfasst hier nicht, wie in Abschnitt 2.5 beschrieben, den unternehmensweiten Zugriff auf alle verfügbaren MSS, sondern den abteilungsorientierten Zugriff der zur Verfügung stehenden BI-Komponenten in PMM. Wir haben diese durch integrierte Standard-Reports mit *Crystal Reports für .NET* und durch eine Schnittstelle

²⁹ Die Datenbank basiert auf Microsoft SQL Server 2000 als relationales Datenbankmanagementsystem.

zur Web-basierten Benutzeroberfläche von *BusinessObjects* für Ad-hoc-Abfragen und -Analysen realisiert. Der Präsentations-Layer baut auf dem Logik-Layer auf, der für den Endanwender sowohl standardisierte Reports als auch Ad-hoc-Abfragen und -Analysen zur Informationsgenerierung bereitstellt (s. Abschnitt 3.3.2). Im Rahmen von PMM ist der Präsentations-Layer mit dem Logik-Layer durch die verwendeten Anwendungen eng miteinander verknüpft und wird daher nicht separat betrachtet.

Im Daten-Layer werden die Daten einerseits durch Crystal Reports für .NET und andererseits durch BusinessObjects für die Logik-Schicht aufbereitet. Ziel der Entwicklung von PMM ist der Einsatz zum umfangreichen Projektmanagement. PMM soll von Abteilungs- bzw. Unterabteilungsleitern, Projektleitern und Mitarbeitern genutzt werden. Das Anwendungsfalldiagramm in Abb. 3.19 (s. S. 66) verdeutlicht grob die verfügbaren Anwendungsfälle und nimmt dabei Bezug auf die Berechtigungen der einzelnen Benutzer. Ein Anwendungsfalldiagramm beschreibt die Zusammenhänge zwischen unterschiedlichen Anwendungsfällen untereinander sowie zwischen Anwendungsfällen und deren Akteuren. Anwendungsfälle beschreiben eine Vielzahl von Aktivitäten eines Systems aus der Sicht ihrer Akteure (vgl. Oestereich et al. (1999), S. 354). Dabei werden die Anwendungsfälle durch Ellipsen dargestellt. Somit zeigt das Anwendungsfalldiagramm aus Abb. 3.19 die Anwendungsfälle des Systems PMM (z. B. Mitarbeiter anlegen). Aus Abb. 3.19 (s. S. 66) wird ersichtlich, dass ein Projektleiter z. B. Projekte und Aktivitäten anlegen sowie Reports generieren darf.

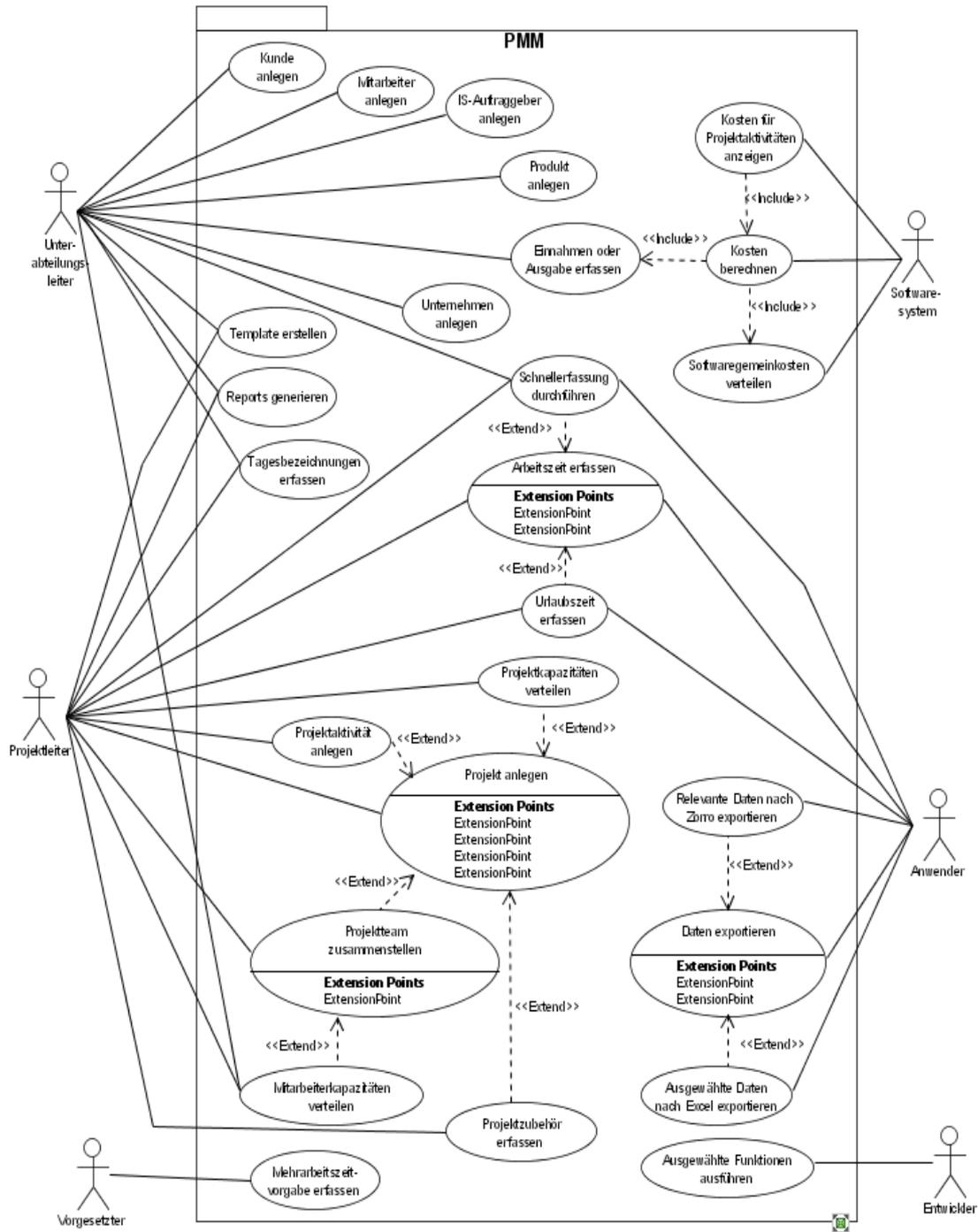


Abb. 3.3: PMM – Use-Case-Diagramm

Quelle: Henke (2006), S. 6

Die Hauptfunktionalitäten der Anwendung sind jedoch komplexer als abgebildet und werden im Weiteren ohne Bezug auf Berechtigungen vorgestellt:

- *Planung und Budgetierung*

Die Projekte werden nach Jahren geordnet in einer hierarchischen Baumstruktur angezeigt. Jedes Projekt enthält ein Teilprojekt, das zudem Prozesse, verwendete Softwarelizenzen, Dienstleistungsvereinbarungen (DLVs) oder Aktivitäten enthalten kann.

Jedes Element verfügt über die gleichen einstellbaren Eigenschaften wie z. B. „Elementtyp“, „Name“ oder „Nummer“ (Abb. 3.4, s. S. 41). Diese sind in den folgenden Tabellen der Datenbank abgelegt:

`_PROJECT_STRUCTURE` (rekursive Tabelle)
`_ELEMENT_TYPE` (Eigenschaften-Tabelle)
`_ELEMENT_CLASSIFICATION` (Eigenschaften-Tabelle)
`_ELEMENT_STATUS` (Eigenschaften-Tabelle)

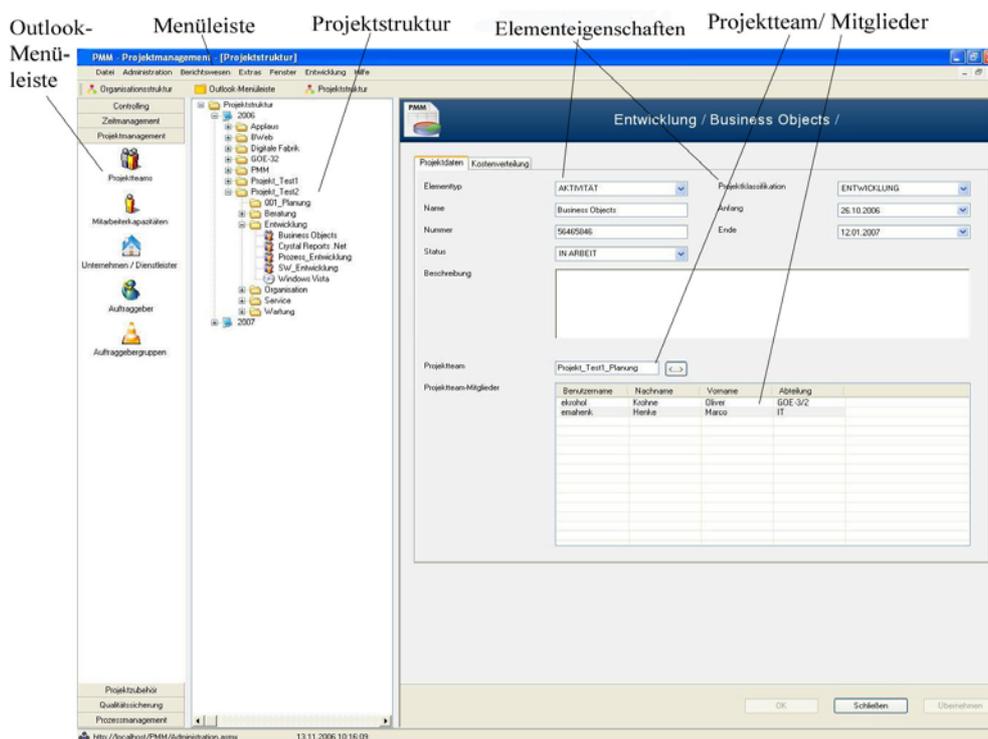


Abb. 3.4: PMM – Projektdaten

Zusätzlich können zu jedem Bauelement der Projektstruktur Plan-, genehmigte Budget-Werte oder Ist-Werte gebucht werden. Der Plan-Wert eines Elements ist der angenommene Budget-Wert für dieses Element. Der genehmigte Budget-Wert ist der letztendlich von der Unternehmensleitung zugewiesene Budget-

Wert³⁰ für dieses Element. Für die Budget-Werte ist nur ein Eintrag pro Element zugelassen. Somit werden die Budgets der Unterelemente eines Teilprojekts bzw. eines Projekts aggregiert und als einzelner Budget-Wert³¹ angezeigt (s. Abb. 3.5, S. 41).

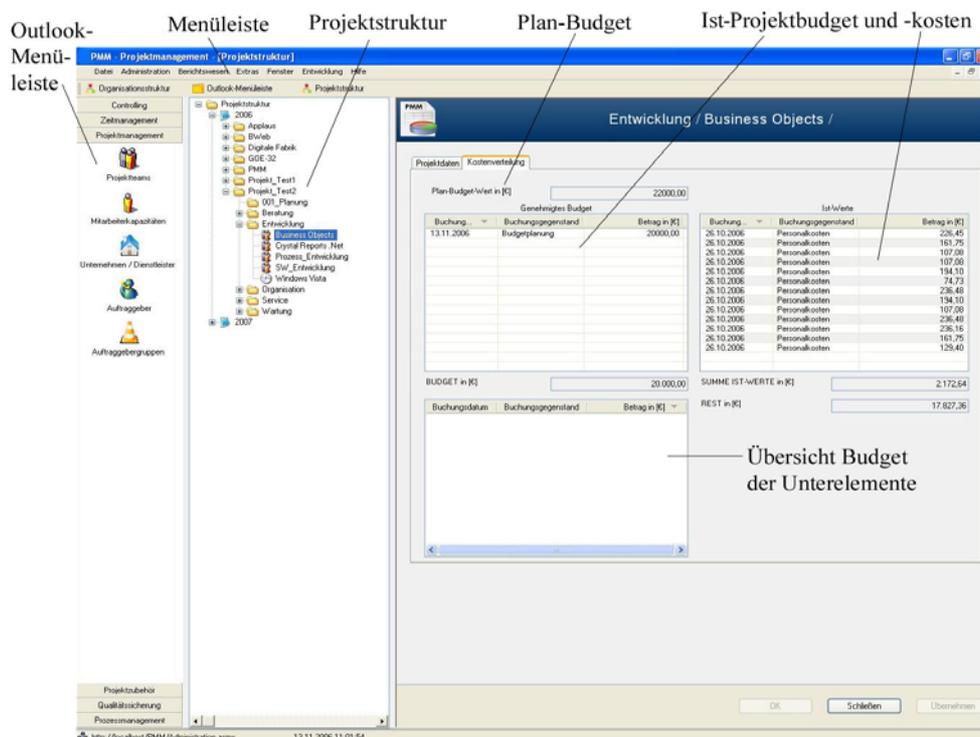


Abb. 3.5: PMM – Plan-Ist-Vergleich

Gegen den genehmigten Budget-Wert werden die Ist-Werte gebucht und zusammen als Gesamtkostenrechnung³² im Hauptbuch (Tabelle³³: _GENERAL_LEDGER) auf der Datenbank gespeichert und verrechnet. Ist-Werte sind sowohl Ausgaben wie z. B. Personalkosten oder Softwarebeschaffungen als auch Einnahmen durch Dienstleistungen wie z. B. „IS entwickeln und einführen“ oder „Prozessberatung durchführen“.

Die Mitarbeiter protokollieren ihre Projektstunden mit Hilfe der Zeiterfassung über ihnen zugewiesene Aktivitäten aus dem Projektstrukturbaum, welche als Personalkosten in die Kostenrechnung mit einfließen. Somit sind Aussagen über verschiedene Kosten (Personalkosten, Anschaffungskosten usw.) und Aufwände

³⁰ Hierbei sei anzumerken, dass meist nur für ganze Projekte Budget-Werte genehmigt werden und die Projekt- und Abteilungsleiter die weiteren Budgetverteilungen auf die Unterelemente selbstständig durchführen.

³¹ Um einen detaillierten Überblick über die Teilbudgets des genehmigten Budgets zu bekommen, werden diese nach Anwahl des genehmigten Budgets in einer kleinen Liste unter dem Ist-Budget-Konto angezeigt (s. Abb. 3.5, S. 41).

³² Unter Gesamtkostenrechnung wird hier die Verrechnung aller Einnahmen und Ausgaben mit dem genehmigten Budget-Wert verstanden.

³³ Mit „Tabelle“ wird im Weiteren auf die Tabellen der PMM Datenbank verwiesen.

möglich. Die Aufwände lassen sich anhand der verwendeten Projektstunden der Mitarbeiter für die jeweiligen Aktivitäten berechnen. Durch die Verteilung der Budgets und die Auflistung der Kosten und Einnahmen bietet das System zu jedem Zeitpunkt einen aktuellen Status des Projektbudgets sowie des Jahresbudgets. Dies dient der Überwachung und Steuerung der Projekte (s. Abb. 3.4; Abb. 3.5, S. 41). Weiterhin bietet die Übersicht der zuvor geplanten und der aktuellen Budget-Werte einen Plan-Ist-Vergleich der Projektbudgets.

Zu jedem Projekt kann zusätzlich ein Anfangs- und ein Enddatum festgelegt werden. Vorgesehen ist die Implementierung von Meilensteinen für Projekte und Aktivitäten, um über einen Ampelbericht³⁴ erkennen zu können, ob Meilensteine bereits erreicht wurden. Bis zur Fertigstellung dieser Funktionalität können bereits Aussagen wie „Ist mein Projekt schon zu teuer?“ oder „War meine Budgetplanung korrekt?“ getroffen werden.

Eine weitere Planungsfunktion, die durch die Projektstruktur ermöglicht wird, sind Forecast-Berechnungen. Sie können durchgeführt werden, indem zukünftige Projekte anhand historischer oder erneut auftretender Projekte geplant werden. Dazu können Templates als Beispielprojekte angelegt oder die Kopie eines historischen Projekts angepasst werden. Unter Zuhilfenahme dieser Mechanismen wird das Treffen zukünftiger Aussagen bezüglich Projektentwicklungszeit und -kosten ermöglicht. Dieses Verfahren ist allerdings nicht direkt mit einer Trendanalyse vergleichbar, da die Trendanalyse die Kostenentwicklung über die nächsten Jahre mit Hilfe von BI-Komponenten aufzeigt. Die Trendanalyse mit Verwendung von BI-Komponenten untersucht die historischen Daten eines DWH auf Muster und Zusammenhänge und stellt sie in einem Trendverlauf über die Zeit dar. Daraus können schließlich Trends abgeleitet werden. Im Falle von PMM findet allerdings keine echte Historisierung statt. Die Forecast-Berechnungen sind lediglich als Planungsfunktion und nicht als Analysefunktion zu verstehen.

In PMM werden die Projektplanung und die Ressourcenplanung getrennt betrachtet. Es bietet darüber hinaus Funktionalitäten zur Ablaufkoordination, da Mitarbeiter in Projektteams eingeteilt werden können. Diese Teams können daraufhin Elementen im Baum zugeordnet werden. Diese Einteilung der Mitarbeiter kann durch die Mitarbeiterkapazitätenplanung übersichtlich nachvollzogen werden.

³⁴ Ein Ampelbericht listet die betrachteten Aktivitäten auf und markiert jede Aktivität mit einer Farbe. Rot steht für „Nicht begonnen“. Gelb steht für „In Bearbeitung“. Grün steht für „Abgeschlossen“.

Daraufhin werden die Personalkosten zu den jeweiligen Projekten durch die angelegten Stundenlöhne (Tabelle: `_HOURLY_WAGE`) und die Projektarbeitszeiten (Tabelle: `_TIME_X_PROJECT_STRUCTURE`) zu Personalkosten in der Gesamtkostenrechnung verrechnet. Diese Personalkosten werden ebenso auf der Datenbank gespeichert. Die Personalkosten werden ebenso auf der Datenbank durch die Beziehungen zwischen den folgenden Tabellen gespeichert:

```

_GENERAL_LEDGER
_TIME_X_PROJECT_STRUCTURE
_WORKING_TIME_PER_DAY
_USER

```

Jedoch kann jeder Mitarbeiter nur über jene Aktivitäten berichten, für die der Projektleiter im Vorhinein Kapazitäten für das Jahr festgelegt hat. Diese Jahreskapazitäten (Tabelle: `_USER_X_PROJECT_STRUCTURE`) verhelfen zur organisatorischen Planung der Verteilung von Projektaktivitäten und fließen nicht automatisch in die Budgetplanung ein. Mitarbeiter können, wie oben erwähnt, zusätzlich zur Steuerung der Projektorganisation in Projektteams (Tabelle: `_PROJECT_TEAM`) eingeteilt werden, dies hat jedoch keine Auswirkungen auf die Kostenrechnung. Die Speicherung der Stundenlöhne und der ausgeführten Aktivitäten (z. B. Prozessberatung durchgeführt, Pflichtenheft erstellt, Programmcode erweitert, Dokumentation erstellt usw.) ermöglichen, Aussagen über die entstandenen Personalkosten und die verwendeten Personalkapazitäten der Projekte treffen zu können. Diese Daten werden in der Planung und Budgetierung berücksichtigt.

- *Ressourcenplanung*

Durch PMM ist eine ausführliche Betrachtung der im Unternehmen verfügbaren Ressourcen möglich, um dem Aufgabengebiet der erwähnten Abteilung gerecht zu werden. Die frühere aufwendige Erstellung des Bebauungsplans sowie des Lizenz- und Releasemanagements wird nun durch die Verwaltung der Software in PMM geregelt.

Zu jeder Software (Tabellen: `_SOFTWARE` und `_LICENSE`) werden allgemeine Daten sowie Anschaffungs- und Wartungskosten abgelegt. Dadurch sind für jede Software bzw. Softwarelizenz die Kosten verfügbar (s. Abb. 3.8, S. 46). Zur Bedarfsplanung können weiterhin Softwarelizenzen zu bestimmten Organisationseinheiten der Organisationsstruktur des PP zugeordnet werden und die Plan- und Ist-Werte der benötigten und verwendeten Softwarelizenzen eingetragen werden (s. Abb. 3.9, S. 46).

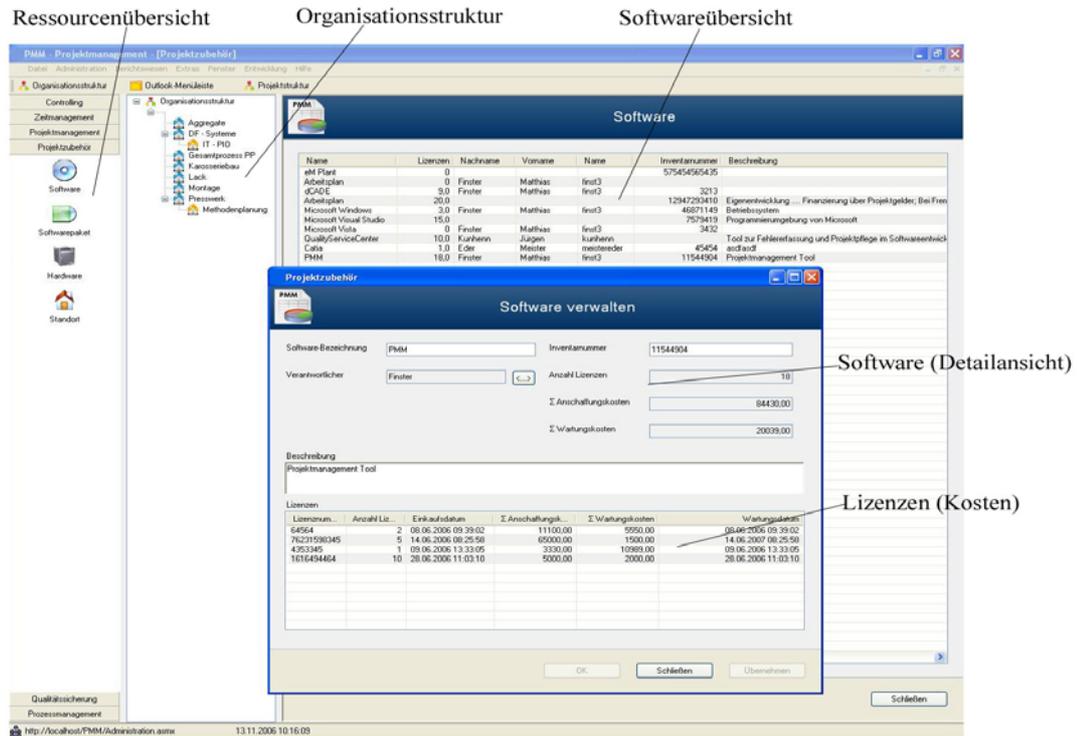


Abb. 3.8: PMM – Software-/ Lizenzkosten

Die Elemente der Organisationsstruktur und deren Eigenschaften werden in folgenden Tabellen abgespeichert:

- _ORGA_STRUCTURE (rekursive Tabelle)
- _ORGA_ELEMENT_TYPE (Eigenschaften-Tabelle)

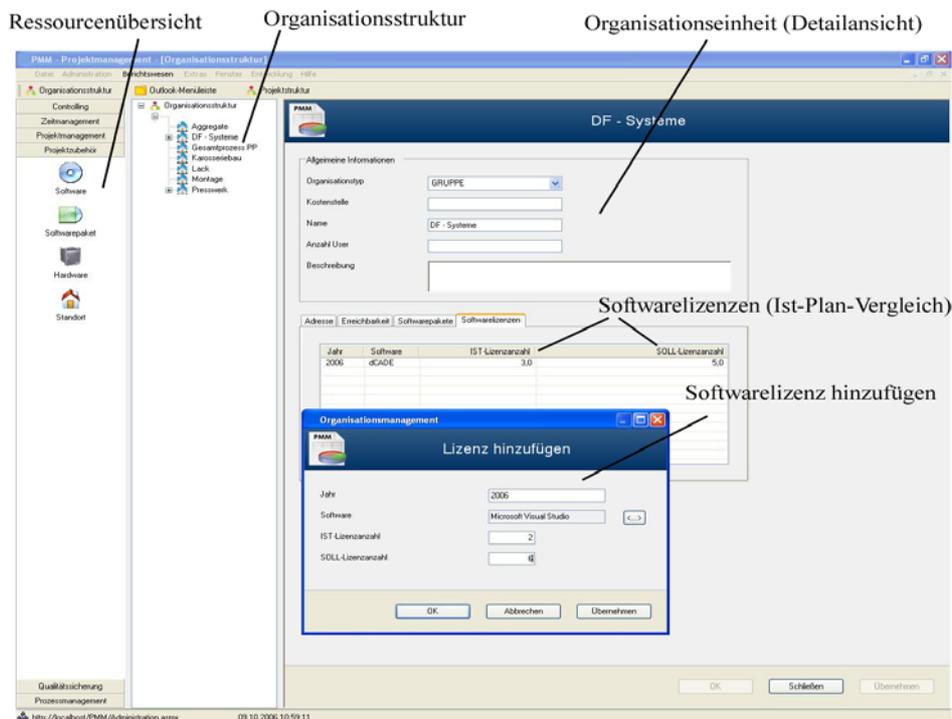


Abb. 3.9: PMM – Zuordnung Software/ Organisationsstruktur

Durch die vordefinierten Standard-Reports ist eine Auswertung über Plan- und Ist-Verwendung von Softwarelizenzen ähnlich des Bebauungsplans möglich. Weitere Analysen erlauben Aussagen über Softwareanschaffungskosten und -wartungskosten in den verschiedenen Bereichen oder Regionen des Unternehmens. Zur Ressourcenplanung zählen auch Auskünfte zur verwendeten Hardware und deren Standorte, die in folgenden Tabellen gespeichert sind:

```
_HARDWARE  
_COMPUTER  
_HARDWARE_TYPE  
_BUILDING  
_FLOOR  
_ROOM
```

Hierbei ist jedoch vorerst nur die Abteilungshardware berücksichtigt. Dabei werden lediglich Stammdaten und Kosten der verschiedenen Hardware aufgenommen und ihre Standorte festgehalten. Für die Ressourcenplanung sind im Hinblick auf die Arbeitsplatzverfügbarkeit Aussagen über die Kapazitäten in verschiedenen Büros („Wie viele Mitarbeiter befinden sich in Büro X?“) und die verwendeten bzw. zur Verfügung stehenden Netzwerkanschlüsse (Tabelle: `_NETWORK`) möglich.

- *Controlling/ Berichtswesen*

Das Controlling und das Berichtswesen bilden zusammen das Kernstück von PMM, das alle Analysemöglichkeiten über die Daten bereitstellt. Es wurde im praktischen Teil dieser Diplomarbeit realisiert. Das standardisierte Reporting und die konfigurierbaren Ad-hoc-Analysen ermöglichen eine Aussage über den jeweiligen Projektstatus und die verfügbaren Kapazitäten (s. Abschnitt 3.3.1). Darüber hinaus werden Analysemöglichkeiten bezüglich Finanzen, Personal und Ressourcen durch die Datenaufbereitung und -generierung in BusinessObjects bereitgestellt (s. Abschnitt 3.3.2).

Zusammenfassend kann festgehalten werden, dass PMM demnach eine zeitpunktgenaue Finanz-, Kapazitäten-, Projekt- und Ressourcenplanung sowie zahlreiche Analysemöglichkeiten bietet. PMM gewährleistet die differenzierte Betrachtung des Zeitaufwandes für verschiedene Projektabschnitte bzw. Projektaktivitäten. Auf diese Weise können zu jedem Zeitpunkt der Projektfortschritt und die verursachten Kosten nachvollzogen werden. Zudem unterstützt PMM die Abteilungs- und Projektleiter zum einen bei der Ressourcenplanung im Zusammenhang mit dem Software- und Lizenzmanagement im gesamten Unternehmen. Zum anderen bietet es Unterstützung bei der Planung

damit verbundener Anschaffungs- und Wartungskosten. Zu den oben genannten Hauptfunktionalitäten ermöglicht PMM zusätzlich den Export ausgewählter Daten nach MS Excel und den Export von Reports ins MS Excel-, MS Word-, PDF-, HTML-, XML- oder RichText-Format.

Im folgenden Abschnitt wird auf das zugrunde liegende Datenmodell von PMM eingegangen, da die Grundlage des Daten-Layers verdeutlicht werden soll.

3.2.2 Datenmodell

PMM greift auf eine transaktionale, relationale Microsoft SQL Server 2000 Datenbank als Datenhaltung zu. Einen semantischen Gesamtüberblick der Entitäten und deren Beziehungen untereinander gibt das ERM in Abb. 3.10 (s. S. 49). Die Datenbankstruktur ist nach zweiter Normalform (2NF) und teilweise nach dritter Normalform (3NF) normalisiert.

Eine Beschreibung zur Verwendung der Tabellen wurde bereits während der Anwendungsbeschreibung gegeben (s. Abschnitt 3.2.1).

Anhand des ERMs und der bereits erwähnten Ausführungen (s. Abschnitt 3.2.1) geht hervor, dass es sich um eine operative Datenbank für die Unterstützung des Wertschöpfungsprozesses handelt. Aufgrund der Voraussetzungen in der betrachteten Abteilung sollen die Reporting- und Analysefunktionen direkt auf dieser operativen Datenbank aufsetzen. Dies widerspricht allerdings der Auffassung, dass Reporting- und Analyse-Systeme auf einer dispositiven Datenbank aufzusetzen haben. Bei der Integration von BI-Komponenten wurde jedoch die in Abschnitt 2.3.2 erwähnte virtuelle Data Warehouse Architektur gewählt. Dadurch können die Reporting- und Analysefunktionalitäten von PMM direkt auf der operativen Datenbank aufsetzen. Diese Architekturvariante wird aufgrund der Kosten- und Zeitersparnis der Entwicklung eines separaten DWH vorgezogen. Die BI-Komponenten sollen demnach trotz operativer Datenbank die nötigen Funktionalitäten zur Abfrage und Analyse bereitstellen. Dadurch soll eine verbesserte Entscheidungsunterstützung erreicht werden. Diese Erweiterung durch BI-Komponenten wird im weiteren Verlauf erläutert.

3.3 Integration von Business Intelligence-Komponenten

Die Ausgangssituation der Digitalen Fabrik und Fertigungsplanung wurde in Abschnitt 3.1 dargelegt. Daraus lässt sich erkennen, dass die Reportingfunktionalitäten der früheren Anwendungen nur statischer Natur waren und keinerlei dynamische Reporterstellung gewährleisteten. Daher waren auch keine Ad-hoc-Analysen möglich.

Daraus folgte, dass an PMM die Anforderung gestellt wurde, dynamische Reportgenerierung bzw. Ad-hoc-Analysen zu ermöglichen, um die vorhandenen Daten übersichtlich aufzubereiten, darzustellen und zu analysieren. Für sofortige standardisierte Auswertungen wie z. B. monatliche Zeiterfassungen, Projektstunden der Mitarbeiter oder Plan-Ist-Vergleiche der Softwareressourcen im Unternehmen sollen Standard-Reports dennoch erhalten bleiben. Weiterhin sollen durch Ad-hoc-Analysen in Bezug auf die Projektplanung u. a. Aussagen über die aktuellen Projektkosten sowie die damit verbundenen Personalkosten und Mitarbeiterkapazitäten getroffen oder Plan-Ist-Vergleiche der Projektbudgets übersichtlich dargestellt werden. Daher sollen die Kosten anhand der Dimensionen „Mitarbeiter“, „Zeit“ und „Projekte“ analysierbar sein. Zusätzlich sollen Forecast- bzw. Trendanalysen ermöglicht werden, d. h. anhand historischer und aktueller Projekte sollen zukünftige Projekte geplant und eingeschätzt werden können. What-If-Analysen sollen weitere Simulationsergebnisse über die Projektverläufe geben.

In Bezug auf die Ressourcenplanung sollen Abfragen erstellt werden können, die Plan-Ist-Aussagen ähnlich des Bebauungsplans (s. Abb. A.1, S. 128) wiedergeben. Weiterhin sollen durch die Abfragen, Kosten anhand der Dimensionen „Ressource“, „Zeit“ und „Organisationseinheit“ analysiert werden können. Ferner sollen Ad-hoc-Analysen Informationen über die Mitarbeiter und ihre Arbeitszeiten preisgeben.

Um derart flexible und komplexe Analysemöglichkeiten zur Verfügung zu stellen, bedarf es der Verwendung von einzelnen in Abschnitt 2.4.2 erwähnten BI-Komponenten, die Standard-Reporting und Ad-hoc- bzw. OLAP-Analysen zulassen.

Wie in der Definition zu BI (s. Abschnitt 2.2) erläutert, sind die am IT-Markt angebotenen BI-Komponenten lediglich Werkzeuge für die Entwicklung einzelner BI-Anwendungen zur Konkretisierung von Teilaspekten im unternehmensspezifischen Gesamtansatz. Sie sollen tragfähige und miteinander verknüpfende Lösungen zur Entscheidungsunterstützung bieten. Daraufhin wurde die Auswahl der auf dem IT-Markt verfügbaren BI-Komponenten mit den Voraussetzungen der betrachteten Abteilung abgestimmt.

In den letzten Jahren ist zu den bekanntesten Vertretern des IT-Marktes wie Crystal Reports oder Cognos eine Vielzahl von Report- und Analyse-Systemen hinzugekommen. Marktführende Datenbankhersteller wie Microsoft oder Oracle gehen dazu über, ihrem Produkt ein Reportprogramm mitzuliefern. Jedoch sind diese ausschließlich mit der eigenen Datenbank verwendbar und reichen meist nur für einfache Reports und Analysen mit geringer Grafik aus (vgl. Mischke (2006), S. 112). Stattdessen sollten Report- und Analyse-Systeme kompatibel mit möglichst vielen Datenquellen sein, d. h. je mehr verschiedene Datenquellen mit einem geringen Aufwand erreichbar und auswählbar sind, desto umfangreicher sind die Importmöglichkeiten und desto flexibler verläuft die Arbeit mit dem System (vgl. Mischke (2006), S. 112). In diesem Zusammenhang nennt MISCHKE Importschnittstellen wie Zugriff auf gängige Datenbanken (z. B. SQL Server, My SQL, Informix, DB2, Oracle, etc.), Zugriffe über *Open Database Connectivity*³⁵ (ODBC) oder *Java Database Connectivity*³⁶ (JDBC), Zugriff auf multidimensionale Datenbanken oder die Unterstützung zur Einbindung von Textdateien (z. B. *Extensible Mark-up Language* (XML) oder *Comma Separated Values* (CSV)). Beim Import von relationalen Datenbanken ist es unerheblich, ob es sich um eine operative oder eine dispositive Datenbank handelt. Zum Import von multidimensionalen Datenbanken muss der Zugriff explizit von einer Importfunktion bereitgestellt werden. Als weitere wichtige Eigenschaften der Reporting-Systeme zählt MISCHKE Funktionalitäten wie flexible Generierung von SQL-Befehlen und Speicherung dieser in Repositories auf. Weiterhin fügt MISCHKE hinzu, dass Templates, Gruppierungs-, Aggregierungsfunktionalitäten, Parameterverwendung sowie Rechtevergabe zur benutzerspezifischen Auswahl der Tabellen oder Datensätze vorhanden sein sollen (vgl. Mischke (2006), S. 112 f.). OLAP- bzw. Ad-hoc-Analysesysteme sollen, wie schon im Abschnitt 2.4.2 beschrieben, unterschiedliche Auswert- und Analysemöglichkeiten unter Berücksichtigung der FASMI-Regeln bereitstellen. Die Entscheidung, welches der auf dem Markt vorhandenen Reporting-Analyse-Systeme als BI-Komponente in PMM integriert wurde, fiel unter Berücksichtigung einer virtuellen Data Warehouse Architektur auf *Crystal Reports für .NET* (CR.NET) zur Standard-Report-Generierung und *BusinessObjects* (BO) für Ad-hoc-Analysen.

CR.NET ist die API (Applikationsprogrammierschnittstelle) für VS.NET des Reporting-Systems *Crystal Reports* der Firma Business Objects. Die API wird verwendet zur Integration der Fähigkeiten von Crystal Reports in andere Programme. CR.NET ist derart konzipiert, dass es sich nahtlos in die anderen Komponenten von VS.NET einfügt. Es ist eine reine Reporting-System-Komponente, d. h. dass es – wie in

³⁵ODBC ist eine standardisierte Datenbankschnittstelle, die SQL als Datenbanksprache verwendet und den Zugriff auf relationale Datenbanken ermöglicht (vgl. Bauer/Günzel (2001), S. 135).

³⁶ JDBC ist eine Programmierschnittstelle der Java-Plattform, die Zugriff auf verschiedene relationale Datenbanken gewährleistet.

Abschnitt 2.4.2 beschrieben – für das betriebliche Berichtswesen konzipiert ist. Es wird dazu verwendet, Daten aus Datenquellen zu ziehen und diese übersichtlich darzustellen.

BO ist ein BI-Tool der gleichnamigen Firma Business Objects, das auf jedes Bedürfnis zur Entscheidungsunterstützung individuell anpassbare Tools zur Überwachung, Steuerung und Analyse von Informationen zur Verfügung stellt. Im Konzern des in der vorliegenden Arbeit betrachteten Automobilherstellers wird BusinessObjects (BO) XI Release 2 als Konzernstandard verwendet. Laut Herstellerangaben enthält BO XI BI-Komponenten der Datenintegration für den Zugriff, die Aufbereitung und Integration von Daten. Weiterhin bietet es Funktionen des Performance Managements zur Abstimmung von Strategien und Maßnahmen. Zusätzlich können Reports sowie Ad-hoc-Abfragen und -Analysen durch Funktionalitäten der Abfrage und Analyse erstellt werden. Zur Organisation von BI-Werkzeugen, -Berichten und -Anwendungen hält BO ebenso eine BI-Plattform bereit (vgl. Business Objects (2006)). Damit zählt BO zu den OLAP-Systemen und berücksichtigt zusätzlich die FASMI Regeln, die in Abschnitt 2.4.2 dargestellt wurden. In dieser Diplomarbeit wurde sich für die Anwendung PMM auf die BI-Komponenten der Abfrage und Analyse beschränkt.

CR.NET und BO wurden aus folgenden Gründen als BI-Komponenten zur Integration in PMM verwendet:

Crystal Reports für .NET (CR.NET)

- Es sorgt für eine kostengünstige Beschaffung, da die Grundfunktionalitäten (ausreichend für Standard-Reports) im Liefer- und Lizenzumfang von Visual Studio .NET³⁷ (VS.NET) enthalten sind.
- Es bietet umfangreiche Importmöglichkeiten der Daten aus gängigen Datenbanken über OLE DB³⁸ oder über Datenbankschnittstellen wie ODBC. Dadurch unterstützt es die virtuelle Data Warehouse Architektur. Weiterhin ermöglicht es Importmöglichkeiten der Daten aus Datenquellen wie MS Excel oder MS Access sowie aus XML-Dateien.

³⁷ PMM wurde in Visual Basic .NET in der VS.NET Entwicklungsumgebung programmiert, somit ist die Lizenz vorhanden.

³⁸ Bei OLE DB handelt es sich um eine von Microsoft produzierte Datenbanktechnologie, die einen Zugriff auf andere Datenbanken unabhängig von ihrer Speicherform ermöglichen (vgl. Microsoft (2006)).

- Es ist voll in die VS.NET Entwicklungsumgebung integriert, womit Reports direkt in die Anwendung (hier PMM) integriert werden können. Dies hat den Vorteil, dass das Benutzerrechte-Konzept³⁹ und das Look-and-Feel⁴⁰ von PMM übernommen werden.
- Es stellt Funktionalitäten zur Reporterstellung von Gruppierung, Summierung über Parameterverwendung bis hin zur Einbindung von Grafiken zur Verfügung.
- Es ermöglicht den Export der Reports in das MS Excel-, MS Word-, pdf-, html-, xml- und rtf-Format zur Verteilung im Unternehmen. Damit stehen standardisierte Exportformate zur Verfügung.
- Nachteilig ist jedoch, dass CR.NET als alleinige Lösung nicht ausreichend ist. Dies wird daraus ersichtlich, dass die Reports nicht zur Laufzeit von PMM verändert werden können, mit Ausnahme der Interaktivität aufgrund von Parametern. Dies setzt zum einen voraus, dass im Vorfeld eine gründliche Auseinandersetzung mit den benötigten Reports stattfinden muss, zum anderen Änderungen mit zusätzlichem Aufwand in VS.NET implementiert werden müssen. CR.NET ermöglicht zwar dynamische Reports, lässt jedoch keine Ad-hoc-Analysen zu. Um dies bewerkstelligen zu können, wird BusinessObjects (BO) verwendet.

BusinessObjects (BO)

- Abgesehen davon, dass BO bei dem betrachteten Automobilhersteller als Konzernstandard für BI-Lösungen vorhanden und einzusetzen ist, bietet es einen enormen Funktionsumfang, der individuell auf verschiedene Bedürfnisse zur Überwachung, Steuerung und Analyse von Informationen anpassbar ist.
- BO erlaubt den Import aller gängigen relationalen Datenbanken über OLE DB oder ODBC. Es unterstützt den Zugriff auf operative Datenbanken als virtuelle Data Warehouse Architektur sowie den Zugriff auf relationale DWH.
- Es gewährt den Export der fertigen Reports in das html-, rtf-, txt-, pdf- und MS Excel-Format oder die Speicherung im Repository für den unternehmensweiten Zugriff.

³⁹ Durch die Integration von Crystal Reports in VS.NET kann der Programmcode für die Anwendung und die Reports in jeglicher Hinsicht geändert und angepasst werden.

⁴⁰ Bei Look-and-Feel handelt es sich um das Erscheinungsbild und die Bedienung der Benutzeroberfläche.

- Die Datenbereitstellung und die Informationsgenerierung sind getrennt, somit benötigt lediglich der Entwickler des sogenannten Universums Kenntnisse über die zugrunde liegenden Datenbanken. Der Endanwender kann ohne Datenbank- oder SQL-Kenntnisse im Look-and-Feel der betriebswirtschaftlichen Sicht Ad-hoc-Reports und -Analysen erstellen.
- Das Prinzip der Universen ermöglicht für jeden Anwendungszweck und jede Betrachtungsweise eines Endanwenders die Erstellung einer speziellen, individuellen Sichtweise.
- Es ist eine einfache Wartung durch Änderung der Universen möglich.
- Reports und Analysen können in Repositories gespeichert werden und durch das Benutzerrechtekonzept nur ausgewählten Endanwendern zur Benutzung bereitgestellt werden.
- Das Benutzerrechtekonzept bietet anwenderspezifische Einstellungsmöglichkeiten und Rechtevergaben für Benutzer und Gruppen.
- Nach Presseinformationen des Herstellers wurde BO zum dritten Mal in Folge zum BI-Markt-Führer ernannt und ist zudem führend im Bereich Abfrage-, Reporting- und Analyse-Tools für Endanwender (vgl. Klaus/Voigt (2006), S. 1).
- BO hält sich an die FASMI Regeln und gehört somit zu den OLAP-Systemen, wodurch der Endanwender einen schnellen, direkten und interaktiven Zugriff auf analysegerechte dispositive Daten erhält. Dies setzt eine Datenaufbereitung im Daten-Layer voraus.

Abschließend lässt sich sagen, dass PMM entwickelt wurde, um die Geschäftsprozesse im Hinblick auf die Projekt- und Ressourcenplanung effizienter zu gestalten. Darüber hinaus sollen BI-Komponenten verwendet werden, um die operativen Daten aus dem Wertschöpfungsprozess auszuwerten und zu analysieren. Dazu soll einerseits für Standard-Reports CR.NET und andererseits für Ad-hoc-Abfragen und -Analysen BO verwendet werden. Dies soll zur zusätzlichen Verbesserung der Geschäftsprozesse durch Verbesserung der Entscheidungsunterstützung beitragen.

Im folgenden Abschnitt wird detaillierter auf die Vorgehensweisen zur Datenbereitstellung für Reporting- und Analysezwecke eingegangen. Zudem wird die Informationsgenerierung der vorgestellten Tools, BO und CR.NET beschrieben, um die BI-Komponenten von PMM darstellen zu können. Dies ist erforderlich, um den Endanwendern auswertbare und analysegerechte Daten für die Entscheidungsunterstützung darbieten zu können. Erst dadurch wird die Qualität der Entscheidungsunterstützung verbessert und der Zeitaufwand von Entscheidungsphasen verringert. Dies führt zusätzlich zur Kostenreduzierung.

3.3.1 PMM-Daten-Layer

Wie bereits in Abschnitt 2.3 erklärt wurde, ist der Daten-Layer für die dispositive Datenbereitstellung zuständig. Dazu gehört die komplette Datenintegration, -aufbereitung und -speicherung. Dies führt zu einer konsistenten Datenhaltung. Hierbei sind die verschiedenen Datenquellen die Grundlage für die spätere Informationsgenerierung. Es können entweder strukturierte, quantitative Daten oder semi- bzw. unstrukturierte, qualitative Daten aufbereitet werden. Die aufbereiteten Daten werden meist in multidimensionaler Struktur zur effizienteren Analyse aus mehreren operativen Datenbanken in einem DWH mit einer relationalen oder multidimensionalen Datenbank als Grundlage gespeichert.

Im Falle von PMM sollen jedoch nur die durch PMM erstellten strukturierten, quantitativen, operativen Daten in einer relationalen Datenbank gespeichert und analysiert werden. Daher sollen die Reports und Analysen direkt auf dieser operativen Datenbank ausgeführt werden, um dem Aufwand des Aufbaus eines DWH entgegen zu wirken und demnach Kosten sparen zu können. Da diese Datenbank nur strukturierte, quantitative Daten enthält, entfällt die Aufbereitung von semi- bzw. unstrukturierten, qualitativen Daten. Die Vorgehensweise der Datenaufbereitung ist die gleiche wie bei vielen Reporting- bzw. Ad-hoc-Analysesystemen (vgl. Mischke (2006), S. 112). Das Tool zeigt Tabellen einer oder mehrerer Datenbanken übersichtlich an. Zwischen den einzelnen Tabellen erstellt der Entwickler Beziehungen, aus denen das System SQL-Abfragen erstellt. Anschließend können verschiedene Datensätze für die Auswertung verwendet, manipuliert und dargestellt werden.

Im Folgenden werden die Unterschiede der Datenbereitstellung von CR.NET und BO beschrieben und die Integration in PMM erläutert. Dadurch wird sichergestellt, dass der Endanwender sich auf entscheidungsrelevante Daten für seine Entscheidungen stützt.

Datenbereitstellung in Crystal Reports für .NET

CR.NET besitzt zur Erstellung von Manipulationen und Berechnungen eine eigene Formel-Sprache. Zusätzlich beinhaltet es eine Vielzahl von Funktionen, die dazu verwendet werden können, Rohdaten in präsentationsfähige Reports mit Graphen, Diagrammen, Summen usw. zu verwandeln (vgl. McAmis (2004), S. 2). Die Reports, die bei CR.NET die Eigenschaften von *Standard-Reports* haben, müssen während der Anwendungsentwicklung erstellt werden. Daher ist es nicht möglich, Ad-hoc Analysen zu erstellen. Da jedoch vordefinierte und einheitliche Standard-Reports für jeden Endanwender vorhanden sein müssen, wurden für das Projektmanagement-System PMM Standard-Reports (im weiteren Verlauf Reports genannt) in die Anwendung integriert.

CR.NET hat zwei unterschiedliche Sichtweisen. Einerseits die Endanwendersicht, die den Logik-Layer, also die Informationsgenerierung während der Laufzeit des Programms, widerspiegelt (s. Abschnitt 3.3.2). Andererseits verdeutlicht die Entwicklersicht die Sicht des Daten-Layers, welche die Datenbereitstellung widerspiegelt (s. Abb. 3.11, S. 57). Diese Datenbereitstellung der Entwicklersicht beinhaltet die Anbindung verschiedener Datenquellen über Tabellenfelder, die Auswahl, Manipulation und Aufbereitung der Datensätze über Formelfelder, Parameterfelder, Gruppenfelder, SQL-Ausdrucksfelder sowie deren übersichtliche Darstellung in einem Report. Aufgrund dieser Eigenschaften kann CR.NET als ETL-Komponente verstanden werden, da die Rohdaten durch die Selektion gefiltert, anschließend harmonisiert, verdichtet und in Form von Berechnungen angereichert werden. Diese Eigenschaft dient der umfangreichen Manipulation der Datensätze und deren übersichtlichen Darstellung.

Die Darstellung der Entwicklersicht ist auf die Anwendungsentwicklung von VS.NET beschränkt. Auf die Anwendungsentwicklung in VS.NET wird im weiteren Verlauf dieser Diplomarbeit aufgrund des Umfangs verzichtet. Stattdessen wird die Erstellung und Integration eines Crystal Reports in eine mit VS.NET entwickelte Anwendung beschrieben. Um einen Report zu erstellen, der zukünftig in der entwickelten Anwendung (hier PMM) abgerufen werden kann, legen wir die Vorgehensweise (Punkte 1 – 3) fest. Als begleitendes Beispiel verwenden wir die Erstellung eines Reports der Zeiterfassung mit für den Endanwender auswählbarem Zeitraum und übersichtlicher Jahres/Monats-Darstellung (s. kompletten Crystal Report in der Entwicklersicht Abb. 3.16, S. 61 und in der Endanwendersicht Abb. 3.33, S. 89).

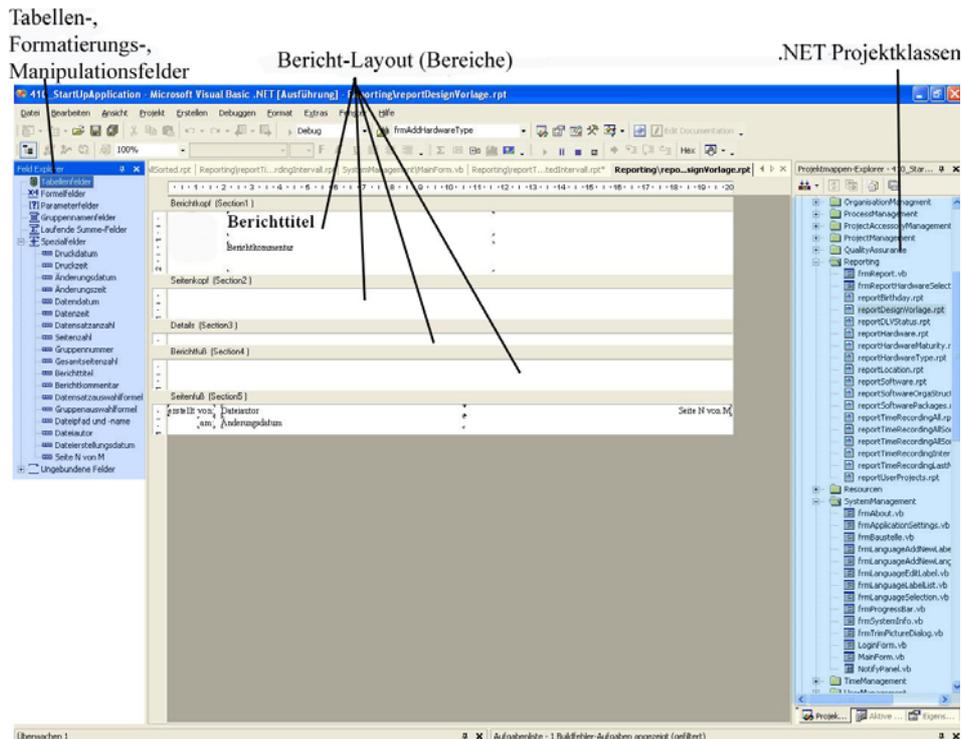


Abb. 3.11: Crystal Reports – Layout

1) Auswahl der Datenquellen und Datensätze

Nachdem dem VS.NET-Projekt eine neue Crystal Reports Datei hinzugefügt wurde, müssen die relevanten Datenquellen für diesen Report eingebunden und die Tabellen oder Dokumente ausgewählt werden. Dem Entwickler stehen Schnittstellen zu verschiedenen relationalen Datenbanken über OLE DB und über die Datenbankschnittstelle ODBC sowie zu Datenquellen im Excel-, Access- und XML-Format zur Verfügung.

Für den Import einer relationalen Datenquelle ist es unerheblich, ob es sich um eine operative Datenbank oder ein DWH handelt. Im betrachteten Projekt PMM wird direkt auf die operative Datenbank lesend⁴¹ zugegriffen, um die Datensätze nicht zu verändern. Nach Auswahl der Tabellen aus der Datenbank erscheint ein Verknüpfungsassistent, der die Tabellen in einer Übersicht darstellt. Daraufhin werden Methoden zur Erstellung von Beziehungen zwischen den Tabellen zur Verfügung gestellt (s. Abb. 3.12, S.58). Diese Beziehungen bzw. Joins⁴² können auf unterschiedliche Weise definiert werden. Es besteht die Auswahl zwischen INNER JOIN⁴³, LEFT OUTER JOIN⁴⁴, RIGHT OUTER JOIN⁴⁵ und FULL OUTER JOIN⁴⁶.

⁴¹ CR.NET erlaubt nur lesenden Zugriff auf Datenquellen.

⁴² Ein Join verknüpft jeweils die Zeilen zweier Tabellen über gemeinsame Attributwerte zu einer neuen Tabelle (vgl. Heuer/Saake (2000), S. 116).

⁴³ Inner Join oder Join verbindet alle Zeilen, bei denen in beiden Tabellen ein entsprechender Datensatz existiert.

⁴⁴ Left Outer Join verknüpft zwei Tabellen und behält alle Zeilen der linken Tabelle bei.

⁴⁵ Right Outer Join verknüpft zwei Tabellen und behält alle Zeilen der rechten Tabelle bei.

⁴⁶ Full Outer Join verknüpft zwei Tabellen und behält alle Zeilen beider Tabellen bei.

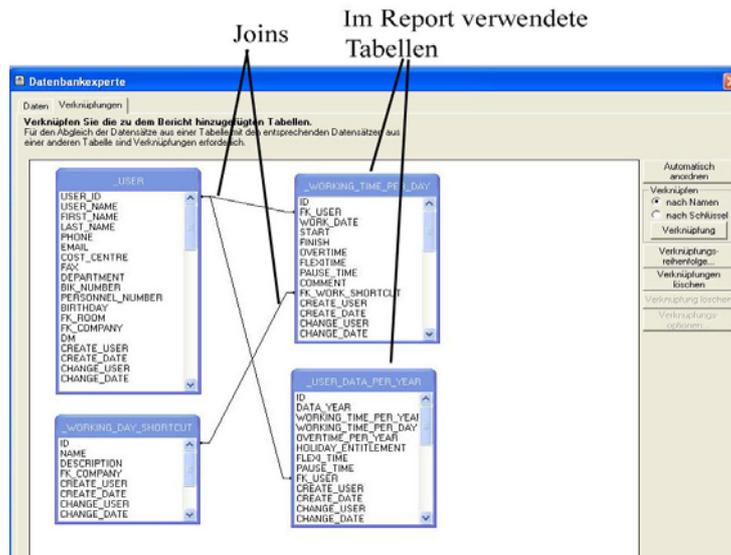


Abb. 3.12: CR.NET – Verknüpfungsassistent

Im begleitenden Beispiel haben wir die nachfolgenden Tabellen ausgewählt und durch INNER JOIN miteinander in Beziehung gesetzt, da nur die Datensätze ausgewählt werden sollen, bei denen die Fremdschlüssel mit den Primärschlüsseln der verschiedenen Tabellen übereinstimmen.:

```

_USER
_WORKING_TIME_PER_DAY
_WORKING_DAY_SHORTCUT
_USER_DATA_PER_YEAR

```

Für jeden Report müssen die Datenbanken neu eingebunden sowie die Tabellen ausgewählt und verknüpft werden.

2) Formatierung, Manipulation und Darstellung der Datensätze

Nach Auswahl und Vorbereitung der Datenquellen können die benötigten Daten aufbereitet werden, d. h. sie können einfach nur angezeigt oder durch Verwendung von Programmiermethodiken oder Formeln manipuliert werden und durch Einordnung in die entsprechenden Bereiche übersichtlich dargestellt werden. Die Elemente können gruppiert und Aggregationen auf verschiedenen Gruppenebenen durchgeführt werden. Jeder Bereich kann weiterhin durch verschiedene Formatierungen bearbeitet werden.

Ein Report-Layout orientiert sich an folgendem Schema (s. Abb. 3.11, S. 57):

```

Berichtkopf47
  Seitenkopf48
    Gruppenkopf49
      Detaildaten50
    Gruppenfuß51
  Seitenfuß52
Berichtfuß53

```

Im vorliegenden Beispiel mussten wir wesentliche Manipulationen und Formatierungen durchführen, um den gewünschten Report (s. Abb. 3.33, S. 89) zu erhalten. Alle Zeitangaben sind in der Tabelle `_WORKING_TIME_PER_DAY` der Datenbank zur einfacheren Berechnung als `Decimal`-Datentyp abgespeichert. Diesen Datentyp mussten wir durch die Anwendung der Formelfelder in einen `String`-Datentyp im Format `hh:mm`⁶⁰ konvertieren. Dies ist in `CR.NET` durch die Verwendung des Formeleditors möglich (s. Abb. 3.13, S.60). Wir haben eine Formel als Formelfeld implementiert, in der die Dezimalzahl in Stunden und Minuten getrennt und in eine 24-Stunden-60-Minuten-Darstellung als `String` konvertiert wird.

Nach Anwendung der Manipulationen und Formatierungen kann das Formelfeld in den Report eingefügt werden. Im begleitenden Beispiel haben wir `@START` verwendet, das den Datensatz `START` der Arbeitszeit repräsentiert. Alle anderen Zeitangaben folgen dem Beispiel auf die gleiche Weise. Weiterhin ist das Datum als `nvarchar`-Datentyp (d. h. `String`) abgespeichert. Diesen haben wir durch eine standardisierte Formel in einen `DateTime`-Datentyp konvertiert und durch Formatierungen kann der Datensatz im gewünschten Format (`TT.MM.YYYY`⁶⁰) im Report angezeigt werden. Zur Übersichtlichkeit und gezielten Anzeige von Datensätzen haben wir die Daten nach Jahreszahlen und Monaten gruppiert (s. Gruppierungen in der Entwicklersicht: Abb. 3.14, S. 60; Gruppierungen in der Endanwendersicht: Abb. 3.33, S. 89).

⁴⁷ Dieser Teil wird nur einmalig zu Beginn des Reports gedruckt und kann das Firmenlogo oder Angaben zur Reporterstellung enthalten.

⁴⁸ Der Seitenkopf erscheint auf jeder Seite des Reports erneut und kann seitenspezifische Angaben enthalten.

⁴⁹ Der Gruppenkopf wird für jede Gruppe erneut gedruckt und enthält Gruppierungen von Datensätzen wie z. B. nach Regionen oder Jahreszahlen. Im Beispiel wird nach Jahreszahlen und Monaten gruppiert.

⁵⁰ Dieser Bereich wird für jeden Datensatz wiederholt und enthält meist die tatsächlichen Daten.

⁵¹ Im Gruppenfuß können z. B. Aggregationsfunktionen eine Zusammenfassung der Gruppendaten präsentieren.

⁵² Der Seitenfuß wird am Ende jeder Seite einmalig gedruckt. Hier können z. B. Seitenzahlen angegeben werden.

⁵³ Der Berichtfuß ist der Abschluss des Reports und kann eine Gesamtzusammenfassung des Reports enthalten.

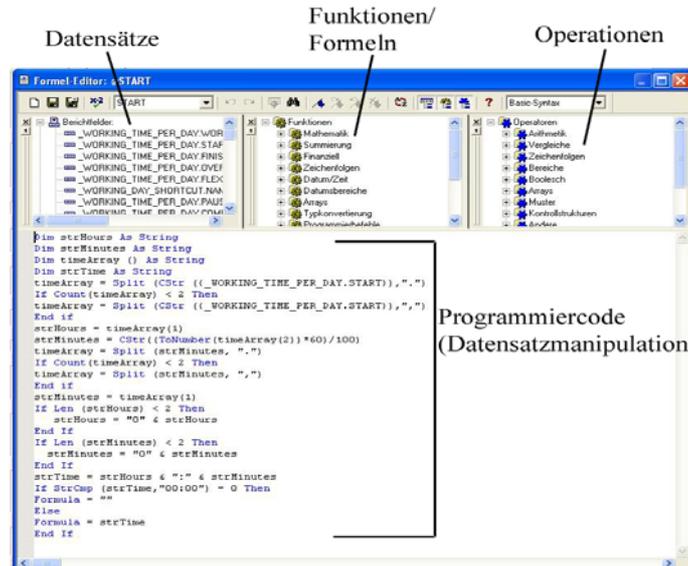


Abb. 3.13: CR.NET – Formeleditor

Die Jahreszahlen haben wir direkt ohne Konvertierungen aus der Tabelle `_USER_DATA_PER_YEAR` ausgewählt⁵⁴.

The screenshot shows a Crystal Reports report layout. It includes sections for 'Berichtskopf a', 'Berichtskopf b', 'Seitenkopf', 'Gruppenkopf 1', 'Gruppenkopf 2a', 'Gruppenkopf 2b', 'Details', 'Gruppenfuß 2', 'Berichtfuß', and 'Seitenfuß'. The main data area is a table with columns for 'Arbeitszeit', 'Mehrarbeitszeit', 'Gleitzeit', and 'Urlaubsanspruch'. The table is grouped by 'Name Gruppe 1' and 'Name Gruppe 2'. Annotations with arrows point to 'Reportbereiche' (pointing to the header sections), 'Gruppenbereiche' (pointing to the group header sections), and 'Gruppenergebnisse' (pointing to the data rows).

Abb. 3.14: Crystal Reports – Gruppierungen

Die Monate sind in den Tabellen nicht vorhanden und müssen durch die Anwendung eines standardisierten Formelfeldes erstellt werden:

MonthName (Month ({@WORK_DATE}))

⁵⁴ Formatierungen wurden durchgeführt, um jeweils nur das Jahr selbst anzeigen zu lassen und keine Tausendertrennzeichen oder Dezimalkommata.

Anschließend haben wir die Monate in festgelegter Reihenfolge sortiert (s. Abb. 3.15 S. 63), um die gewünschte Jahres-Monats-Übersicht zu erhalten.

Selbstdefinierte Gruppennamen
und Reihenfolge

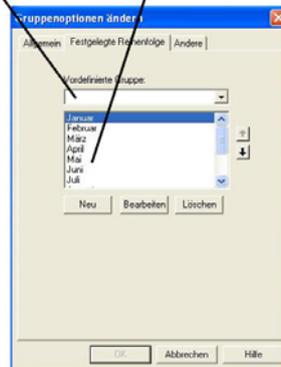


Abb. 3.15: CR.NET – Monatssortierung nach festgelegter Reihenfolge

Am jeweiligen Gruppenfuß haben wir eine Summe der Arbeitszeiten der einzelnen Monate und der Jahre mit Hilfe einer Aggregationsfunktion (hier Summation) erstellt (s. Abb. 3.14, S. 60). Am Ende jedes Reports haben wir die komplette Arbeitszeit im Berichtfuß summiert dargestellt (s. Abb. 3.16, S. 61). Diese Summen werden zur Laufzeit berechnet und im Report angezeigt.

Im Bericht verwendete Formel-,
Parameter- und Gruppenfelder

Anwendername
Datumsintervall

Datensatzbereich
Gruppenergebnisse

Gesamtergebnisse (Report)

Abb. 3.16: Vollständiger Crystal Report

Um einem Report dynamisches Verhalten zu ermöglichen, können Parameter verwendet werden, die zur Laufzeit, d. h. während der Reportgenerierung, abgefragt werden. Parameter werden eingesetzt, um dem Endanwender die Wahl eines beliebigen Zeitraums des Reports freizustellen oder um abzufragen, welche Teile des Reports angezeigt werden sollen und welche nicht. Je mehr dynamische Aspekte ein Report haben soll, desto größer ist der Aufwand während der Entwicklung bzw. Erstellung des Reports.

In dem begleitenden Beispiel des Zeiterfassungsreports haben wir Parameter eingefügt, um zur Laufzeit einen Zeitraum für die Zeiterfassungsausgabe bestimmen zu können und um dem Endanwender nur seine eigene Zeiterfassung ausdrucken zu lassen (s. Abb. 3.31, S. 87; Abb. 3.33, S. 89). Die Erstellung der Parameter erfolgt über Parameterfelder (s. Abb. 3.17, S.62).

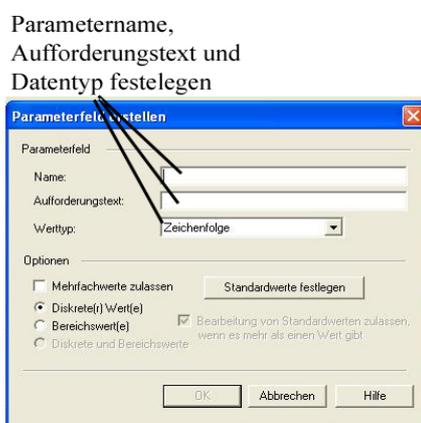


Abb. 3.17: CR.NET – Parameter erstellen

Hier können der Datentyp und diskrete Werte bzw. Bereichswerte festgelegt und eine Aufforderung⁵⁵ für den Endanwender eingetragen werden.

Für den Zeitraum der Zeiterfassung des begleitenden Beispiels haben wir Bereichswerte und für die Identifikation des Endanwenders diskrete Werte festgelegt. Zusätzlich muss der „Auswahl-Assistent“ verwendet werden, der Restriktionen für bestimmte Datensätze des Reports festlegt (s. Abb. 3.18, S. 63). Hierbei haben wir festgelegt, dass der ausgewählte Zeitraum (?DateIntervall) mit den manipulierten Datenfeldern⁵⁶ (@WORK_DATE) der Arbeitszeit übereinstimmen muss, damit der korrekte Zeitraum im Report dargestellt wird. Weiterhin haben wir festgelegt, dass die BenutzerID (_USER.USER.ID) mit dem Parameter für die Identifikation des Endanwenders (?UserID⁵⁷) und das betrachtete Jahr (Year({@WORK_DATE})) mit den jahres-

⁵⁵ Eine Aufforderung kann sein z. B. „Geben Sie einen Zeitraum an, für den die Zeiterfassung ausgedruckt werden soll.“.

⁵⁶ Jedes Datum muss in einen DateTime-Datentyp konvertiert werden, da die Datensätze des Datums auf der Datenbank im String-Datentyp vorliegen.

⁵⁷ Der Parameter zur Anwenderidentifikation wurde innerhalb des Programmcodes festgelegt.

bezogenen⁵⁸ Daten (`_USER_DATA_PER_YEAR.DATA_YEAR`) übereinstimmen müssen, damit jeder Endanwender nur seine eigenen Daten dargestellt bekommt.

Formel, die im Bericht verwendet wird,
um nur gewisse Werte anzeigen zu lassen

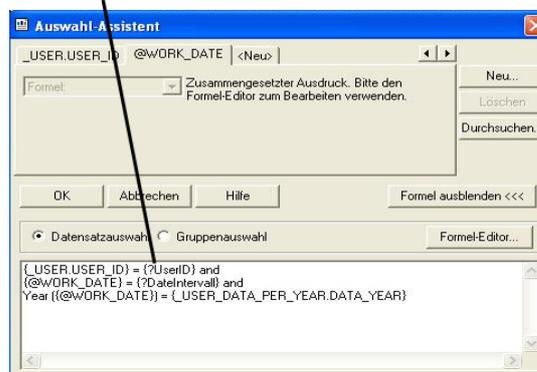


Abb. 3.18: CR.NET – Auswahl-Assistent

Die Parameter werden dann zur Laufzeit abgefragt (s. Abb. 3.31, S. 87). Es ist jedoch möglich, die Parameter durch Programmiermethodiken im Vorhinein festzulegen. Dies haben wir im Falle der Identifikation des Endanwenders umgesetzt. Dabei wurde die ID des aktuellen Benutzers als Parameter übergeben und mit den gespeicherten Benutzerdaten auf der Datenbank abgeglichen. Anhand dieser Methodik wird anschließend ein benutzerspezifischer Report generiert.

Als weitere Manipulation können eigene SQL-Abfragen erstellt und dem Report hinzugefügt werden. Diese werden zum einen verwendet, wenn spezielle Abfragen an die Datenbank gestellt werden müssen, die standardmäßig nicht unterstützt werden. Zum anderen werden SQL-Abfragen verwendet, wenn Tabellen benutzt werden sollen, die zuvor nicht eingebunden wurden. Um den Endanwender zu einer im Vorfeld festgelegten Aktion aufzufordern oder um ihn über Grenzwerte zu informieren, können Warnmeldungen eingefügt werden. Diese Reportwarnmeldungen sind benutzerdefinierte Meldungen, die zeigen, dass Daten im Report bestimmte Bedingungen erfüllen (vgl. Mischke (2006), S. 113). Im vorliegenden Beispiel haben wir Warnmeldungen eingefügt, falls die Urlaubszeit oder die Überstundenzeit ins Negative fallen (s. Abb. 3.34, S. 90). Dabei dient eine einfache Formel, „Wert < 0“, zur Änderung der Hintergrundfarbe des Formelfeldes als Warnmeldung.

Zur weiteren übersichtlichen Darstellung können in beliebigen Bereichen des Reports Diagramme eingefügt werden. CR.NET hält Säulen-, Linien-, Zonen-, Kreis-, Rad-, 3D-, XY-, Netz-, Blasen- und Kurs-Diagramme bereit. Im begleitenden Beispiel haben wir keine Diagramme benutzt, sondern nur bei einem Report, der die Projektarbeitszeiten der Mitarbeiter anzeigt. Bei diesem Report haben wir Kreis-Diagramme einge-

⁵⁸ Die jahresbezogenen Daten spiegeln eine Gruppe wider.

fügt, um dem Projektleiter übersichtlich und schnell die Verteilung der Projektarbeitszeiten jedes Mitarbeiters prozentual anzeigen zu lassen (s. Abb. 3.35, S. 90).

Jeder Bereich, jede Gruppe, jeder Datensatz und jedes Objekt kann formatiert oder manipuliert werden. Außerdem können beliebige Textfelder übersichtshalber hinzugefügt werden. Auf die ausführliche Beschreibung der erwähnten Funktionalitäten wird aufgrund des Umfangs verzichtet.

3) Reportintegration

Im letzten Schritt wird der fertig aufbereitete Report (s. Abb. 3.33, S. 89) in die Anwendung integriert, indem der sogenannte CrystalReportViewer auf einer VB.NET-Form bereitgestellt und mit dem erstellten Report verbunden wird.

Diese vorangegangene Vorgehensweise kann für jeden Report erneut ausgeführt werden, um somit individuelle Reports zu erhalten. Die Erstellung von sogenannten Template-Reports ermöglicht die Weiterverwendung vorhandener Reports. Zusätzlich bietet ein Report-Assistent eine vereinfachte Reporterstellung verschiedener Reportformate an.

In CR.NET können unterschiedlichste Reports erstellt werden. Von einer einfachen Listenform (s. Abb. D.1, S. 131) über Gruppierungen mit Drill-down-Funktionalitäten (s. Abb. 3.33, S. 89) bis hin zu Kreuztabellen. Kreuztabellen zeigen Werte bzw. Ergebnisse der ausgewählten Daten in den Kreuzungspunkten der Kopfzeilen und -spalten an. Dieses Reportformat lässt auf einfache Weise Vergleiche zu und deckt Trends auf. Ein weiteres Format ist die Verwendung von Unterberichten. Dies sind normale Reports, die in einen anderen Report integriert werden, um dadurch kombinierte Reports anzeigen lassen zu können. Die Erstellung dieser kombinierten Reports bezweckt einerseits die übersichtliche Darstellung von Daten, die in keinem Zusammenhang zueinander stehen, andererseits die Koordination von nicht miteinander verknüpften Daten. Zudem ermöglichen kombinierte Reports, einem Endanwender verschiedene Ansichten derselben Daten bereitzustellen (vgl. Mischke (2006), S. 113).

CR.NET ist kein OLAP- bzw. Ad-hoc-Analysesystem. Es kann lediglich zum einfachen Reporting und zur Integration in eine Anwendung verwendet werden.

Insgesamt lässt sich daher festhalten, dass die Aufbereitung und Manipulation der Daten in CR.NET eine übersichtliche Darstellung durch Standard Reports in PMM ermöglicht. Diese Auswertungen können für die Entscheidungsunterstützung verwendet werden. Für Ad-hoc-Analysen mussten wir allerdings auf BusinessObjects XI Release 2 zurückgreifen, da dieses umfangreiche Analysewerkzeuge anbietet. Erst dadurch kann eine verbesserte Entscheidungsunterstützung gewährleistet werden.

Datenbereitstellung in BusinessObjects

Um sowohl Ad-hoc-Abfragen als auch Analysen der Daten aus PMM durchführen zu können, ist es Voraussetzung, auf eine der in Abschnitt 2.3.4 beschriebenen multidimensionalen Datenstrukturen zuzugreifen. Da PMM jedoch auf eine operative Datenbank zugreift, ist es notwendig, diese entweder in eine multidimensionale Struktur in ein DWH zu überführen oder Mechanismen für den anwenderfreundlichen, direkten Zugriff in Form einer virtuellen Data Warehouse Architektur bereitzustellen. Aufgrund des hohen Zeitaufwands und der immensen Kosten wurde auf die Einführung und Entwicklung eines DWH verzichtet. Stattdessen wird auf die Implementierung eines virtuellen Data Warehouses mit Hilfe von BO zurückgegriffen.

In BO gibt es ähnlich – wie bei CR.NET – zwei Sichtweisen, wobei die eine den Entwickler und die andere den Endanwender widerspiegelt. Die Endanwender verwenden BO Webintelligence bzw. BO Desktop Intelligence, die an PMM angebunden werden (s. Abschnitt 3.3.2). BO Desktop Intelligence ist eine Client-Lösung, mit der der Anwender leicht auf Ressourcen interner oder externer Datenquellen zugreifen und diese aufbereiten, formatieren sowie darstellen kann. Für den Datenzugriff und die Datenaufbereitung wird der BO Designer⁵⁹ verwendet, der den Entwicklern die Aufbereitung der Daten im betriebswirtschaftlichen Format ermöglicht. Der Endanwender muss somit keinerlei Kenntnis von SQL- oder anderen Datenbanksprachen sowie der zugrunde liegenden Datenbankstruktur haben.

Der Zugriff auf die unternehmensweiten Informationen, den sogenannten Universen, über einen standardisierten Webbrowser wird BO Webintelligence genannt. Der Endanwender kann BO Webintelligence oder Desktop Intelligence für die Auswahl der Informationen verwenden. Die Informationen werden übersichtlich im betriebswirtschaftlichen Format dargestellt (s. Abschnitt 3.3.2). Der BusinessObjects Designer bietet die Möglichkeit, intern Universen zu erstellen (s. Abb. 3.30, S. 85). Ein Universum ist eine Datei, die folgende Eigenschaften enthält (vgl. Business Objects (2004a), S. 21):

⁵⁹ Der BO Designer ist Teil von BO Desktop Intelligence, es wird jedoch ein spezieller Entwicklerzugriff benötigt.

- Einen Verbindungsparameter für eine Datenbank-Komponente. Hierbei existieren Schnittstellen zu relationalen Datenbanksystemen über OLE DB oder ODBC, zu multidimensionalen Datenbanken, zu Excel-Dateien sowie zu Daten im XML-Format oder Webseiten. Allerdings ist pro Universum nur eine Verbindung zulässig (vgl. Business Objects (2004), S. 26). Daher sollten für verschiedene Anwendungsbereiche unterschiedliche Universen angelegt werden, um damit die Datentrennung exakter koordinieren zu können.
- Ein Datenbankschema, auf das nur die Designer-Entwickler Zugriff haben. Das Datenbankschema ist eine grafische Darstellung der Datenbankstrukturen. Es enthält die aus der Datenquelle verwendeten Tabellen und dazugehörige oder neu erstellte Joins.
- SQL-Strukturen, auch Objekte genannt, welche die aktuellen SQL-Strukturen der Datenbank, wie Spalten, Tabellen und Datenbankfunktionen, darstellen. Objekte werden anhand der Datenbankstrukturen, die in das Datenbankschema eingefügt wurden, erstellt und sind nach Klassen gruppiert. Klassen und Objekte sind sowohl für Entwickler als auch für die Endanwender sichtbar.

Laut BUSINESS OBJECTS dient die Erstellung solcher Universen dazu, technisch unerfahrenen Endanwendern eine anwenderfreundliche und leicht verständliche Benutzeroberfläche zur Verfügung zu stellen. Diese Oberfläche ermöglicht das Ausführen von Ad-hoc-Abfragen auf einer Datenbank zur Berichterstellung und Datenanalyse (vgl. Business Objects (2004a), S. 20). BO unterstützt die meisten relationalen Datenbankschemata von normierten, operativen Datenbankschemata über Stern Schemata bis hin zu Galaxien in DWH (s. Abb. 4.1, S. 110).

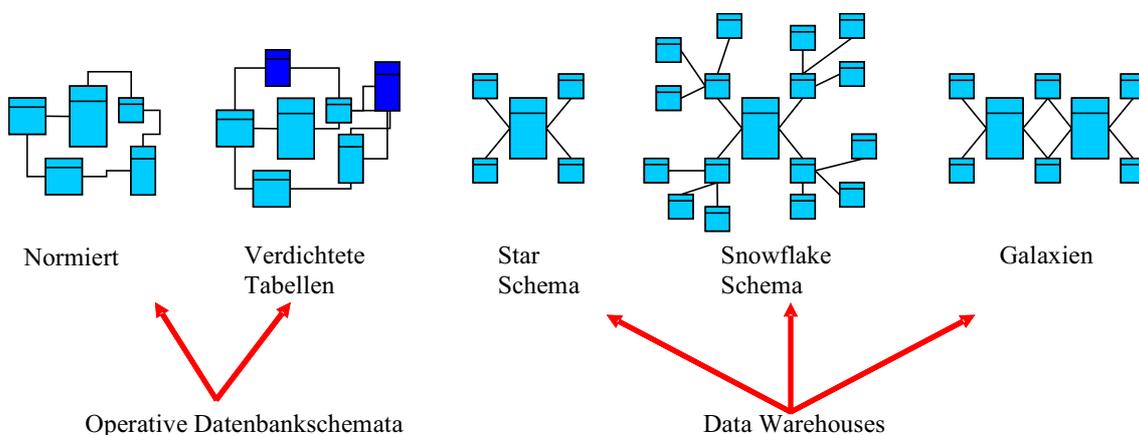


Abb. 3.19: BO Kompatible Datenbankschemata

Quelle: In Anlehnung an: Business Objects (2004a), S. 28

BO unterstützt auf diese Weise und durch das Prinzip der Universen die in 2.3.2 beschriebene virtuelle DWH Architektur. Dabei ist das DWH im Client integriert. Abfragen finden nur lesend direkt auf der operativen Datenbank statt. Durch die Verwendung mehrerer Universen ist es möglich, Strukturen für verschiedene Anwendungszwecke aufzubauen und anzubieten. Die Universen werden auf einer zentralen Domäne in einem Repository gespeichert und sind für alle Benutzer dieser Domäne über den BO Client und BO Webintelligence abrufbar.

Die Vorgehensweise zur Erstellung eines Universums sollte sich an Abb. 3.20 (s. 67) halten. Für PMM haben wir nach dieser Vorgehensweise drei Universen erstellt. Das Universum für das Projektmanagement wird als begleitendes Beispiel während der Vorgehensweise zur Universenerstellung angeführt. Die Universen zur Mitarbeiterkapazitätenplanung und zur Ressourcenplanung werden begleitend angeschnitten und im Abschluss näher beschrieben.

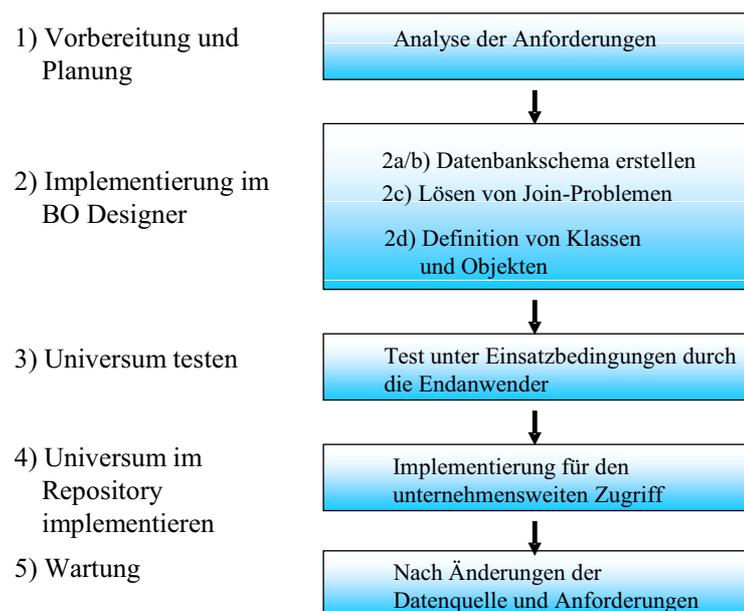


Abb. 3.20: Vorgehensweise zur Universenerstellung

Quelle: In Anlehnung an: Business Objects (2004a), S. 297

Im Folgenden wird die Vorgehensweise (Punkte 1 – 5) zur Erstellung eines Universums in Anlehnung an BUSINESS OBJECTS erläutert (vgl. Business Objects (2004a), S. 33 ff.). Als praktisches Beispiel wird auf die Erstellung des Universums zum Projektmanagement verwiesen.

1) Vorbereitung und Planung

In der ersten Phase ist es wichtig, dass sich der Entwickler mit der Zieldatenquelle und ihren Strukturen vertraut macht. Hierbei sollte er wissen, welche Daten in jeder einzelnen Tabelle der Zieldatenbank enthalten sind, welche Joins zwischen den Tabellen bestehen und welche Kardinalitäten diese beschreiben. Weiterhin muss er die Endanwender und ihre Anforderungen bzw. Anwendungszwecke genau kennen und deren Zusammenhänge verstehen. Er sollte anschließend die Daten ermitteln, die von den Endanwendern benötigt werden und diese mit ihrer Terminologie abstimmen, um sie daraufhin in einem Universum zu implementieren.

Die Planungsphase haben wir intensiv während der Entwicklung von PMM und seiner Standard-Reports durchlaufen, da wir in dieser Phase oft Rücksprache mit den Abteilungsmitgliedern gehalten haben. Zusätzlich haben wir vor Erstellung der Universen über mögliche Szenarien der Ad-hoc-Analysen und die zu verwendende Terminologie der Objekte diskutiert. Für die Analyse der vorhandenen Tabellen wurde sowohl bereits die genaue Funktionsweise von PMM (s. Abschnitt 3.2) als auch die zugrunde liegende Datenstruktur und ihre Tabellen beschrieben (s. Abschnitt 3.2.2). Nach dieser Phase kann zur Implementierung eines virtuellen DWH mit BO übergegangen werden, da, wie schon erwähnt (s. Abschnitt 3.3), von dem Aufbau eines separaten Data Warehouse aus finanziellen und zeitlichen Gründen abgesehen wurde.

2) Implementierung im BO Designer

Der BO Designer wird zur Erstellung der oben beschriebenen Universen verwendet und leitet SQL-Strukturen ab, die dem Endanwender als Klassen und Objekte präsentiert werden.

Im Weiteren wird detailliert auf die Vorgehensweise zur Erstellung eines Universums eingegangen, da hierbei gewisse Probleme zu Fehlern in der späteren Analyse führen können.

2a) Datenbankverbindung erstellen und Tabellen einfügen

Zuerst muss im Designer die Datenbankverbindung hergestellt werden (s. Abb. 3.21, S. 69). Hierbei können, wie zu Anfang der Datenbereitstellung von BO beschrieben, Schnittstellen zu verschiedenen Datenquellen definiert werden. Anschließend sind die Tabellen aus der Datenquelle auszuwählen (s. Abb. 3.21, S. 69). Es sollten nur diejenigen Tabellen ausgewählt werden, die zum einen in der Vorbereitung ausgesucht wurden und zum anderen die Datensätze enthalten, die die Endanwender für ihre Analysen benötigen.

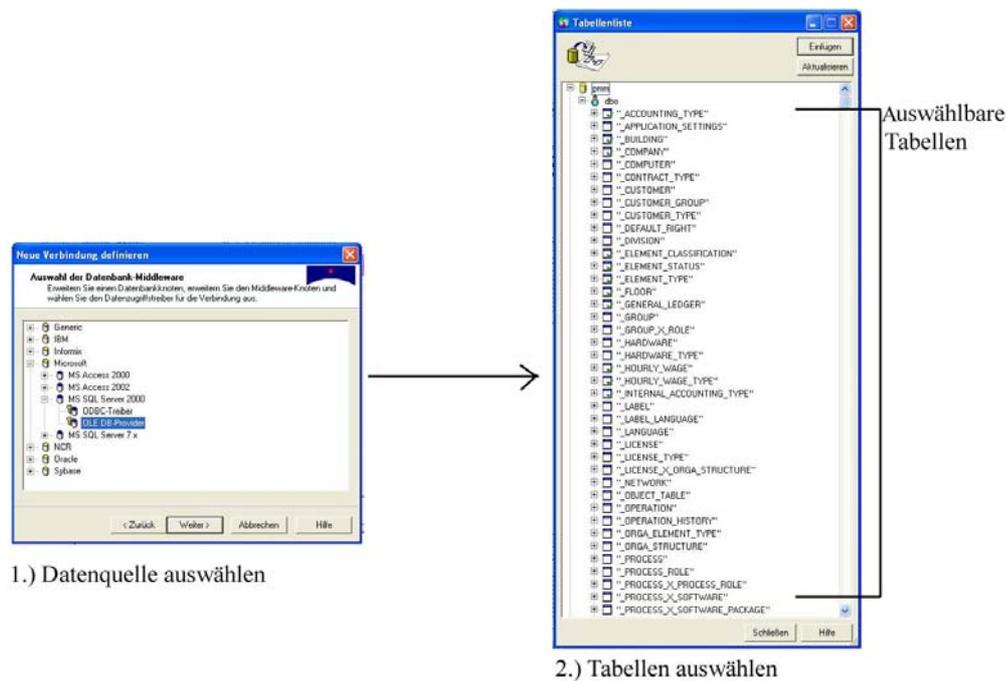


Abb. 3.21: BO Designer – Datenquelle und Tabellen auswählen

In Abb. 3.21 (s. S. 69) sind die ausgewählten Tabellen ausschnittsweise durch grün markierte Haken gekennzeichnet, die wir für das Universum „Projektmanagement“ ausgewählt haben. In der konkreten Umsetzung haben wir für das Projektmanagement alle Tabellen ausgewählt, die mit der Planung und Budgetierung der Projekte in Beziehung stehen. Darunter fallen folgende Tabellen:

_GENERAL_LEDGER
 _INTERNAL_ACCOUNTING_TYPE
 _ACCOUNTING_TYPE
 _PROJECT_STRUCTURE
 _ELEMENT_TYPE
 _ELEMENT_STATUS
 _ELEMENT_CLASSIFICATION
 _USER
 _TIME_X_PROJECT_STRUCTURE
 _PROJECT_TEAM
 _PROJECT_TEAM_X_USER

Die Tabellen `_ROOM`, `_FLOOR`, `_BUILDING` und `_COMPANY` haben wir nur für weitere Informationen über die Mitarbeiter ausgewählt. Eine komplette Übersicht über die verwendeten Tabellen und die im Designer entworfene virtuelle Datenbankstruktur bietet für das Projektmanagement bietet Abb. 3.30 (s. S. 85).

In Abb. C.1 (s. S. 130) wird die virtuelle Datenbankstruktur für die Mitarbeiterkapazitätenplanung dargestellt, hingegen in Abb. C.2 (s. S. 130) für die Ressourcenplanung.

Falls aus einigen Tabellen nur verminderte Datenmengen für die Analyse benötigt werden oder komplexe vorberechnete Werte in Tabellen abgespeichert werden sollen, können sogenannte abgeleitete Tabellen erstellt werden. Eine abgeleitete Tabelle wird durch eine SQL-Abfrage auf Universumebene definiert und kann im Designer als logische Tabelle verwendet werden (vgl. Business Objects (2004), S. 149). Nachdem die Tabellen (automatisch oder manuell) übersichtlich angeordnet worden sind, werden die benötigten Beziehungen bzw. Joins zwischen den Tabellen erstellt.

2b) Erstellen von Joins, Einrichten der Kardinalitäten und Schlüssel

Ein Join wird verwendet, um eine Beziehung zwischen zwei Tabellen herzustellen. Laut BUSINESS OBJECTS wird damit sichergestellt, dass Abfragen, die auf Daten dieser Tabellen zugreifen, korrekte Ergebnisse zurückliefern, wenn beide Tabellen in der Abfrage verwendet werden (vgl. Business Objects (2004a), S. 154).

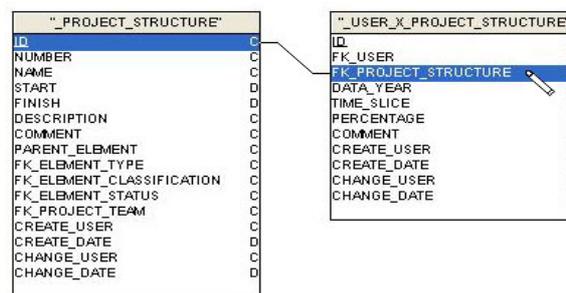


Abb. 3.22: BO Designer – Join erstellen

Joins können automatisch anhand der zugrunde liegenden Datenbank ermittelt oder manuell erstellt werden. Die automatische Ermittlung basiert auf übereinstimmenden Spaltennamen und berücksichtigt nicht die Schlüsselinformationen. Dadurch werden häufig fehlerhafte Joins erstellt. Aus diesem Grunde ist die manuelle Erstellung der Joins zu empfehlen. Bei dieser Erstellung werden die Joins zwischen zwei Tabellen entweder durch grafisches Verbinden der Attribute der Tabellen (s. Abb. 3.22, S. 70) oder durch „Extras -> Join erstellen“ definiert (s. Abb. 3.23, S. 71). Es stehen wie bei CR.NET Outer Joins und normale Joins bzw. *Equi-Joins* zur Verfügung.

Weiterhin ist es möglich, *Theta Joins* und *direkte Joins* zu erstellen. *Theta-Joins* sind benutzerdefinierte Joins, die die Beziehung zwischen zwei Tabellen auf der Basis einer beliebigen Funktion definieren (vgl. Business Objects (2004), S. 178).

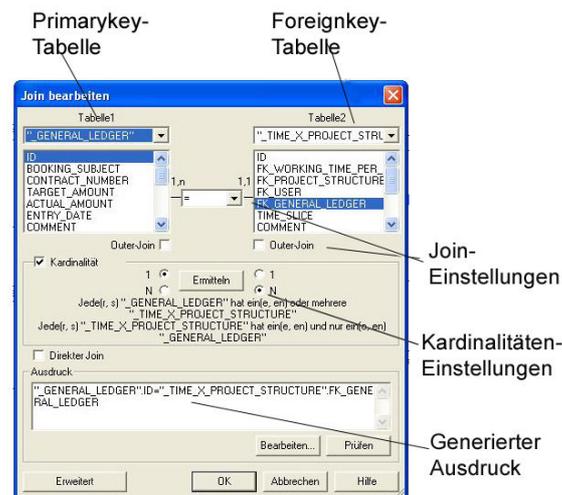


Abb. 3.23: BO Designer – Join/ Kardinalitäten definieren

Direkte Joins definieren alternative Pfade⁶⁰ zwischen Tabellen. Auf diese Weise wird die Abfrageleistung gesteigert, da längere Pfade verkürzt werden, indem für die Abfrage unnötige Tabellen überbrückt werden (vgl. Business Objects (2004), S. 192). Während oder auch nach der Erstellung der Joins müssen die Kardinalitäten festgelegt werden. Die Kardinalitäten werden entweder über das Kontextmenü „Extras -> Kardinalitäten ermitteln“ oder manuell durch Doppelklicken des Joins eingerichtet (s. Abb. 3.23, S. 71). Nach Einrichtung aller nötigen Joins und Kardinalitäten können zusätzlich die Schlüsselinformationen der einzelnen Tabellen zur Indexführung⁶¹ eingestellt werden. Dies führt erneut zu einer erhöhten Abfrageleistung, da dem Designer preisgegeben wird, welche Spalten der Tabelle Primärschlüssel und Fremdschlüssel sind. Dadurch kann er gezielt Schlüsselspalten nutzen und auf effiziente Weise Filter anwenden (vgl. Business Objects (2004), S. 319). Um die Integrität des Universums sicherstellen zu können, müssen sämtliche Joins nach ihrer Erstellung auf Probleme untersucht werden.

2c) Lösen von Join-Problemen

Durch fehlerhafte Beziehungen zwischen Fakt- und Dimensionstabellen in einer relationalen Datenbank können Join-Pfadprobleme auftreten. Ein Join-Pfad ist eine Folge von Joins, die von einer Abfrage für den Zugriff auf Daten verschiedener Tabellen verwendet werden können (vgl. Business Objects (2004), S. 217). Es können drei Arten von Join-Pfadproblemen auftreten. Sie werden als *Schleife*, *Chasm Trap* und *Fan Trap* bezeichnet. Zum besseren Verständnis werden vor der Besprechung der Join-Pfadprobleme die Begriffe Alias und Kontext definiert.

⁶⁰ Die Definition folgt während der Ausführungen zu Join-Problemen.

⁶¹ Die Indexführung ist eine Funktion, die Indexeinträge der Schlüsselspalten nutzt, um den Datenabruf zu beschleunigen. (vgl. Business Objects (2004a), S. 319).

Ein *Alias* ist eine exakte Kopie einer originalen, bereits im Schema vorhandenen Tabelle, die jedoch einen anderen Namen besitzt. Durch die Verwendung der unterschiedlichen Namen ist es möglich, die gleiche Tabelle zweimal in einer Abfrage zu verwenden (vgl. Business Objects (2004), S. 220).

Ein *Kontext* ist ein gültiger Abfragepfad bestehend aus mehreren Joins, der manuell oder automatisch festgelegt wird (vgl. Business Objects (2004a), S 226). Bei der Erstellung eines Reports kann der Endanwender bei mehreren Pfaden den korrekten Kontext wählen. Es sei denn der Kontext wird bei der SQL-Generierung direkt verfolgt, wenn nur ein Kontext für diese Abfrage vorhanden ist.

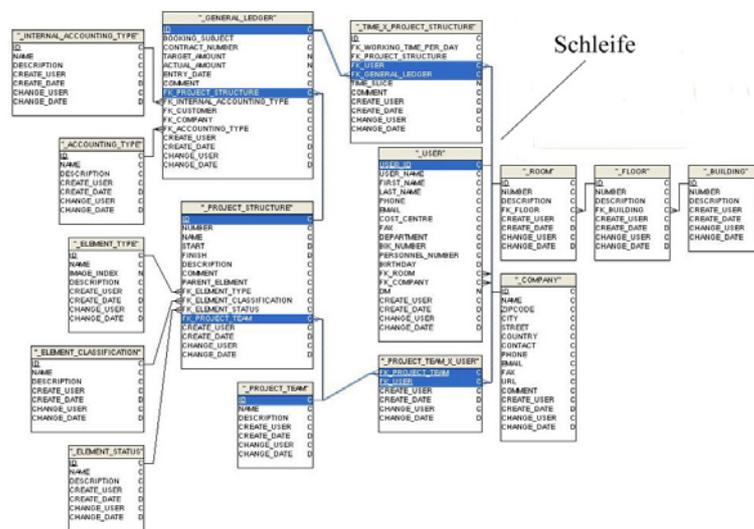
Weitergehend wird nun auf die auftretenden Join-Pfadprobleme eingegangen.

Eine *Schleife* tritt auf, wenn eine Folge von Joins einen geschlossenen Pfad zwischen mehreren Tabellen in einem Datenbankschema bildet (s. Abb. 3.24, S. 73). Daraus ergeben sich Mehrfachpfade zu Dimensionstabellen. Diese Mehrfachpfade geben bei Abfragen an das Universum zu wenig Zeilen zurück (vgl. Business Objects (2004), S. 218; S. 239). Schleifen können durch mehrere Methoden entdeckt werden. Zum einen durch die automatischen Funktionen „Aliase ermitteln“, „Kontexte ermitteln“, „Schleifen ermitteln“ oder „Integrität prüfen“ und zum anderen durch eine visuelle Schemaüberprüfung.

Zu Beginn der Erstellung des Universums für das Projektmanagement haben wir eine Schleife zwischen den folgenden Tabellen aufgedeckt:

```
_PROJECT_STRUCTURE  
_PROJECT_TEAM  
_PROJECT_TEAM_X_USER  
_USER  
_TIME_X_PROJECT_STRUCTURE  
_GENERAL_LEDGER
```

Aus Abb. 3.24 (s. S. 73) ist die Schleife ersichtlich. Die Entstehung der Schleife ist darauf zurückzuführen, dass einerseits der Zeitaufwand und die Personalkosten für die tägliche Betrachtung der Projektaktivitäten der Mitarbeiter im Hauptbuch gespeichert werden. Andererseits werden die Mitarbeiter in Projektteams eingeteilt, die anschließend Projekten zugewiesen werden. Dadurch entsteht eine Schleife, da `_USER` und `_PROJECT_STRUCTURE` in beiden Anwendungsfällen Verwendung finden.



Problem: Schleife
 Lösung: Einführung zweier Kontexte

Vorhandene Tabellen Vorhandene Joins

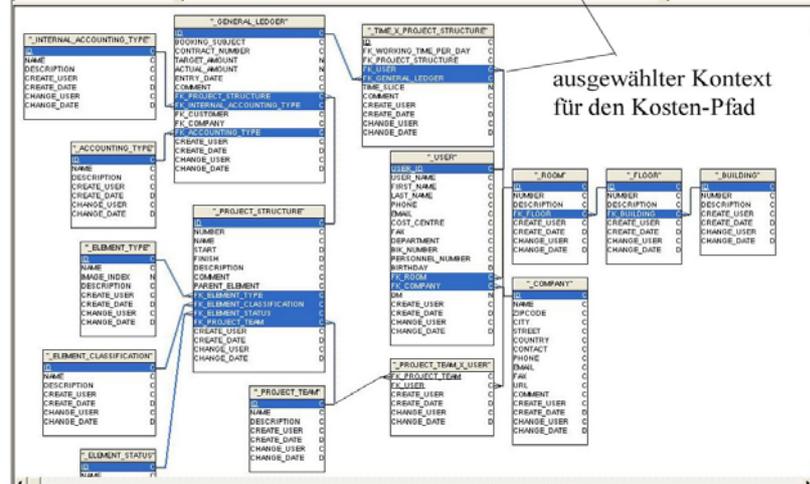


Abb. 3.24: BO Designer – Join-Pfadproblem (Schleife)

Zur Unterdrückung von Schleifen werden Aliase und Kontexte erstellt, welche die mehrfache Verwendung von Tabellen in einer Abfrage ermöglichen und einen bestimmten Pfad für die Tabellen in einer Schleife vorgeben.

Für das entwickelte Anwendungsbeispiel (s. Abb. 3.24, S. 73) haben wir jedoch kein Alias erstellen müssen, sondern es war ausreichend, zwei Kontexte zu erstellen. Durch die unterschiedlichen Kontexte werden die Schleifen unterdrückt, da zwei verschiedene Abfragepfade existieren. Der Kontext *Kosten* beinhaltet die folgenden Tabellen:

```
_GENERAL_LEDGER
_INTERNAL_ACCOUNTING_TYPE
_ACCOUNTING_TYPE
_PROJECT_STRUCTURE
_ELEMENT_TYPE
_ELEMENT_STATUS
_ELEMENT_CLASSIFICATION
_USER
_TIME_X_PROJECT_STRUCTURE
_PROJECT_TEAM
_ROOM
_FLOOR
_BUILDING
_COMPANY
```

Dieser Kontext dient der Einbeziehung der Tabellen, die für die Kostenbetrachtung notwendig sind. Der Kontext `Projektteam` beinhaltet die nachfolgenden Tabellen:

```
_USER
_PROJECT_TEAM
_PROJECT_TEAM_X_USER
```

Daher wird der zweite Kontext für Abfragen verwendet, die die organisatorische Betrachtung der Projektmitglieder benötigen. Damit haben wir die einzige Schleife innerhalb dieses Universums unterdrückt, da nun zwei verschiedene Wege bei SQL-Abfragen verwendet werden können.

Die Schleifen der übrigen Universen haben wir nach dem gleichen Verfahren entdeckt und teilweise durch zusätzliche Einführung eines Alias unterdrückt, wodurch es möglich ist, eine Tabelle mehrfach in eine SQL-Abfrage zu integrieren.

Schleifen aufzulösen oder zu unterdrücken, ist das erste Join-Pfadproblem, das gelöst werden sollte. Nachdem die Unterdrückung der Schleifen beschrieben und durchgeführt wurde, können nun Chasm Traps betrachtet werden. Ein Chasm Trap ist ein weiteres Join-Pfadproblem, das auftreten kann.

Ein *Chasm Trap* kann zwischen drei Tabellen auftreten, indem zwei N:1-Beziehungen auf eine Tabelle existieren und kein Kontext zur Trennung der übereinstimmenden Pfade vorhanden ist (s. Abb. 3.25, S. 75).

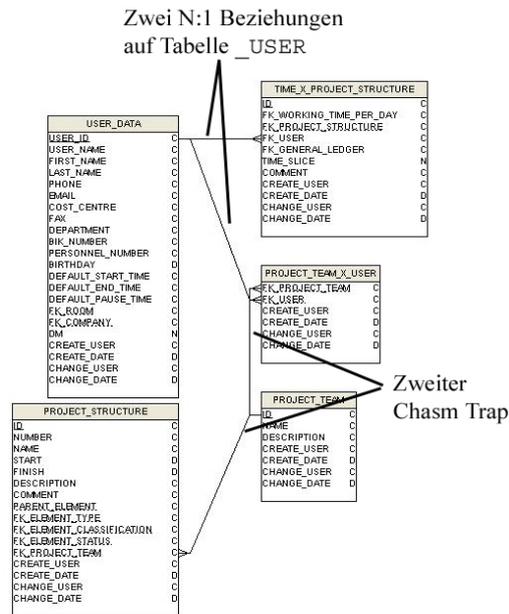


Abb. 3.25: BO Designer – Chasm Trap

Ein Chasm Trap führt bei Abfragen, die Objekte der beiden Tabellen enthalten, die sich am N-Ende der jeweiligen Joins befinden, zu einem kartesischen Produkt⁶². Somit werden mehr Zeilen als erwartet zurückgegeben, da jede Spalte der einen Tabelle mit jeder Spalte der anderen Tabelle kombiniert wird (vgl. Business Objects (2004), S. 270 f.). Sie können nur durch manuelle Analyse des Schemas entdeckt werden und müssen durch die Einführung zweier Kontexte aufgelöst werden. Weiterhin muss die Funktion „Mehrfache SQL-Anweisungen für jede Kennzahl“ im Optionsmenü des Universums ausgewählt werden.

Im Anwendungsbeispiel der Universenerstellung für das Projektmanagement trat ein Chasm Trap zwischen den nachstehenden Tabellen auf:

```

_USER
_TIME_X_PROJECT_STRUCTURE
_PROJECT_TEAM_X_USER

```

Darüber hinaus trat ein weiterer Chasm Trap zwischen folgenden Tabellen auf:

```

_PROJECT_TEAM
_PROJECT_STRUCTURE
_PROJECT_TEAM_X_USER

```

Beide Chasm Traps sind in Abb. 3.25 (s. S. 75) dargestellt. Sie wurden jedoch bereits durch die oben erwähnten Kontexte aufgelöst, da die jeweiligen Kontexte nur die

⁶² Ein kartesisches Produkt ist ein Ergebnis, bei dem alle Kombinationsmöglichkeiten sämtlicher Zeilen der in einer Abfrage enthaltenen Tabellen angezeigt werden (vgl. Business Objects (2004), S. 89).

Tabellen enthalten, die keinen Chasm Trap auslösen (s. Abb. 3.24, S. 73). Bei SQL-Abfragen wird dann der Anwender gegebenenfalls befragt, welchen Kontext er für seine Abfrage verwenden möchte. Teilweise werden allerdings zwei separate SQL-Anfragen mit jeweils einem Kontext erstellt und zusammengefügt. Dies geschieht intern durch BusinessObjects, da die Funktion „Mehrfache SQL-Anweisungen für jede Kennzahl“ ausgewählt wurde.

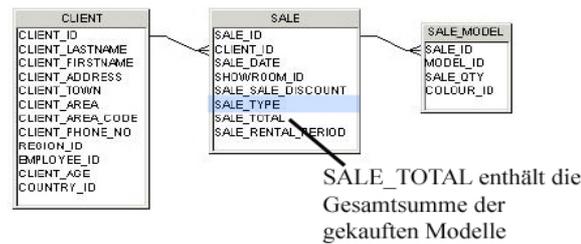
Sowohl bei der Erstellung des Universums für die Ressourcenplanung als auch bei der Erstellung des Universums für die Mitarbeiterkapazitätenplanung traten Chasm Traps auf, die wir durch die Einführung von separaten Kontexten aufgelöst haben. Als drittes Join-Pfadproblem können Fan Traps auftreten.

Ein *Fan Trap* entsteht zwischen drei Tabellen, wobei eine Tabelle durch eine 1:N-Beziehung mit einer weiteren Tabelle verknüpft ist, die ebenso durch eine 1:N-Beziehung mit einer anderen Tabelle verknüpft ist (s. Abb. 3.26, S. 77). Ein Fan Trap führt zu einem erweiterten Ergebnis, wenn eine Abfrage ausgeführt wird. Die Abfrage führt hierbei eine Aggregatfunktion in einer Tabelle aus, die sich am 1-Ende des Joins befindet, während eine am N-Ende vorhanden ist. Dies führt zu einem kartesischen Produkt und gibt zu viele Werte zurück (vgl. Business Objects (2004), S. 278 f.).

Fan Traps traten allerdings weder bei der Erstellung des Universums für das Projektmanagement oder für die Mitarbeiterkapazitätenplanung, noch bei der Erstellung des Universums für die Ressourcenplanung auf. Um dennoch Fan Traps erklären zu können, wird auf ein Beispiel aus dem Business Objects Designer Handbuch zurückgegriffen.

In dem Beispiel auf Abb. 3.26 (s. S.77) würde eine Abfrage über den Gesamtpreis (SALE_TOTAL) und die Modellnummern (MODEL_ID) der gekauften Fahrzeuge für jedes Modell den Wert von SALE_TOTAL zurückgeben, obwohl SALE_TOTAL die Kosten für alle Modelle darstellt.

Ein Fan Trap kann nur durch manuelles Analysieren des Datenbankschemas entdeckt und durch Erstellen einer Alias-Tabelle und Verwendung eines Kontextes aufgelöst werden. Der neu erstellte Kontext in Abb. 3.26 (s. S. 77) wird bei SQL-Abfragen zur Trennung der Abfragepfade verwendet. Anschließend müssen Objekte für den Alias erstellt werden, die den Anwendungsfall beschreiben. Im dargestellten Beispiel wurde das Objekt „Sale Total“ erstellt, welches die Gesamtkosten für alle Modelle darstellt. Die Funktion „Mehrfache SQL-Anweisungen für jede Kennzahl zulassen“ muss auch hier aktiviert werden, damit, wie bereits bei Chasm Traps erwähnt, für jede Kennzahl eine SQL-Abfrage erstellt und anschließend zusammengeführt wird. Dies liefert die korrekten Ergebnisse.



Lösung: - Alias Erstellung (SALE_TOTAL)
- Kontexteinführung zur Pfad-Trennung

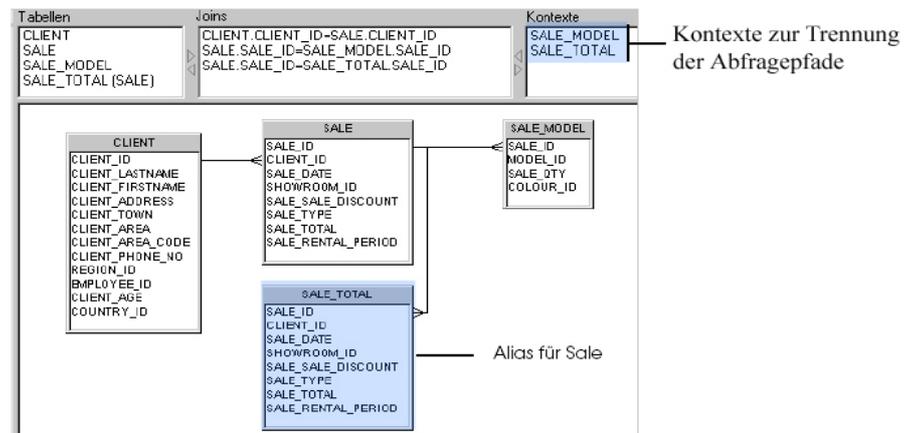


Abb. 3.26: BO Designer – Join-Pfadproblem (Fan Trap)

In Tab. 3.1 (s. S. 77) sind die verschiedenen Join-Probleme und Methoden zu ihrer Ermittlung und Lösung nochmals übersichtlich dargestellt.

Tab. 3.1: Methoden zur Ermittlung und Lösung von Join-Problemen

| Join-Problem | Ermittlungsmethode | Lösung |
|---------------------|--|--|
| Schleife | <ul style="list-style-type: none"> • Aliase ermitteln • Kontexte ermitteln • Schleifen ermitteln • Integrität prüfen • Visuelle Schemaanalyse | <ul style="list-style-type: none"> • Erstellen von Alias-Tabellen und Kontexten zur Schleifenunterdrückung |
| Chams Trap | <ul style="list-style-type: none"> • Visuelle Schemaanalyse | <ul style="list-style-type: none"> • Erstellen von Kontexten |
| Fan Trap | <ul style="list-style-type: none"> • Visuelle Schemaanalyse | <ul style="list-style-type: none"> • Erstellen einer Alias-Tabelle • Erstellen eines Kontextes für diesen Alias • Erstellung von Objekten, die diesen Alias verwenden |

Quelle: In Anlehnung an: Business Objects (2004a), S. 219

Nach erfolgreichem Auflösen der Join-Probleme, muss die Schemaintegrität geprüft werden. Eine automatische Funktion prüft hierbei ungültige Syntax in der SQL-Definition von Objekten, die Existenz von Schleifen im Schema oder in Kontexten, die Existenz von isolierten Tabellen und Joins sowie die Existenz von fehlerhaften oder fehlenden Kardinalitäten. Nach dieser Maßnahme kann davon ausgegangen werden, dass die Datenbankschemastruktur des Universums korrekt aufgebaut worden ist. Im Anschluss daran kann die Definition von Klassen und Objekten durchgeführt werden. Eine Übersicht über die komplette Datenbankschemastruktur des Universums für das Projektmanagement bietet Abb. 3.30 (s. S. 85), für die Mitarbeiterkapazitätenplanung zeigt Abb. C.1 (s. S. 130) und für die Ressourcenplanung stellt Abb. C.2 (s. S. 130) eine Übersicht dar.

2d) Definition von Klassen und Objekten

Klassen sind Container für ein oder mehrere Objekte und können zur Gruppierung dieser verwendet werden. Jedes Objekt muss einer Klasse angehören (vgl. Business Objects (2004a), S. 304). Klassen sind für den Endanwender in Form von Ordnern in einer hierarchischen Struktur sichtbar und sollen ihm die Navigation und Übersicht im Universum erleichtern (s. Abb. 3.2, S. 78).

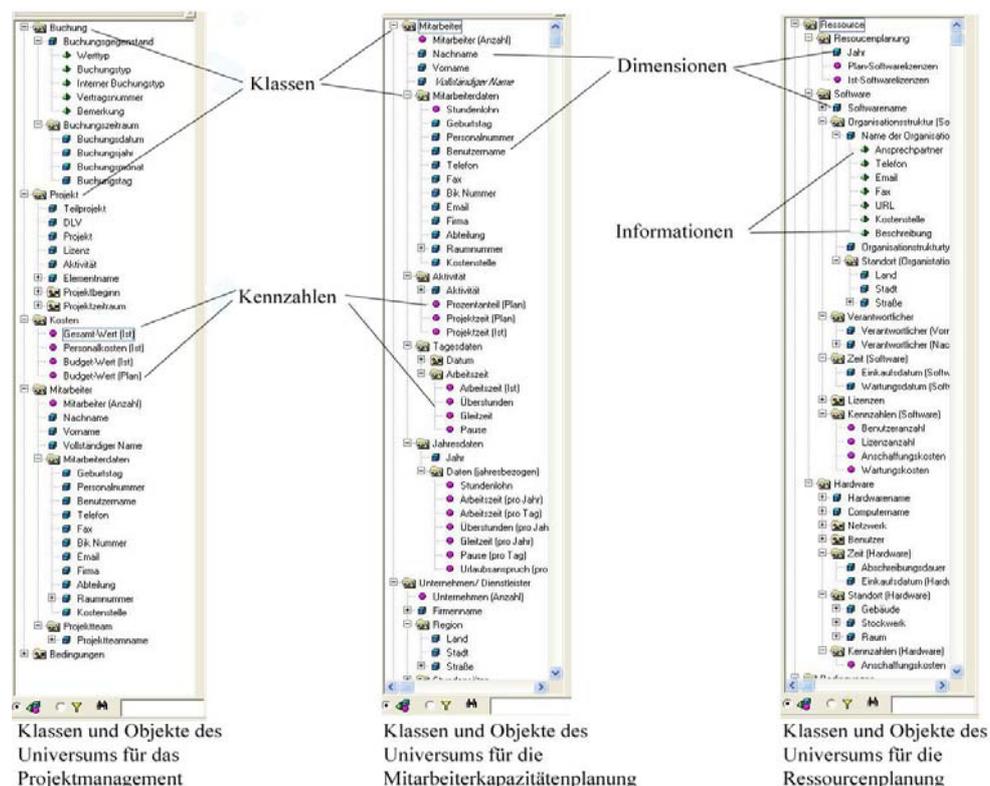


Abb. 3.27: Klassen- und Objektansichtn der Entwickler- und Endanwendersicht

Deshalb ist es sinnvoll, zusammengehörige Dimensions- und Informationsobjekte in Klassen zu gruppieren. Es ist dem Entwickler und seinen Anwendern überlassen, welche Gruppierungsstrategie und Terminologie verwendet wird.

Ein *Objekt* ist ein Universumelement, das durch eine oder mehrere Spalten einer oder mehrerer Tabellen des Universum-Datenbankschemas definiert wird. Ein Objekt kann auch eine aus mehreren Zeilen bestehende Funktion darstellen (vgl. Business Objects (2004a), S. 321). Bei der Reportgenerierung werden nach der Reporterstellung durch den Endanwender aus den definierten und verwendeten Objekten SQL-Select-Befehle abgeleitet und die Ergebnisse zurückgeliefert. Jedes Objekt wird mit einem Namen und einem Datentyp definiert (s. Abb. 3.28, S. 79). Weiterhin wird es durch einen *Select* und *Where*-Bereich, der Spalten und Tabellen des Universum-Datenbankschemas enthält, beschrieben. Liegen die Datensätze jedoch in den Tabellen nicht im korrekten Format vor oder lassen sich keine Funktionen anwenden, um das gewünschte Format zu erreichen, müssen komplexe ETL-Prozesse angewendet werden.

In Bezug auf das Universum für die Mitarbeiterkapazitätenplanung war es uns nicht möglich, die Arbeitszeiten⁶³ der Mitarbeiter sowie das dazugehörige Datum ohne ETL-Prozesse in eine korrekte Zeitdarstellung zu überführen. Wie haben daher davon abgesehen, die Zeiterfassung im Universum abzubilden. Für Reports zur Zeiterfassung müssen die Endanwender auf die in der vorliegenden Diplomarbeit erstellten Standard-Reports zurückgreifen. Für die Berechnungen der Projektaktivitäten sowie Ist- und Plan-Arbeitszeiten der Mitarbeiter für die Ad-hoc-Abfragen haben wir uns in der vorliegenden Arbeit auf die Dezimal-Darstellung beschränkt.

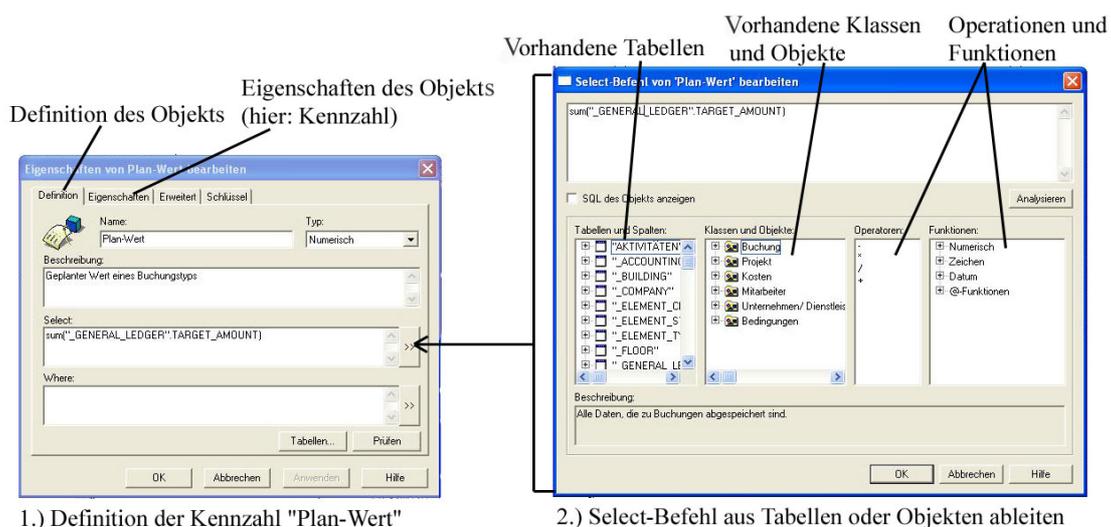


Abb. 3.28: BO Designer – Objekt erstellen

⁶³ Die Start- und Endzeiten der Tagesarbeitszeit sowie alle Zeitdatensätze sind in der Datenbank als Dezimal-Datentyp abgespeichert. Eine Umwandlung wurde in Abschnitt 3.3.1 am Beispiel eines Crystal Reports zur Zeiterfassung beschrieben.

Die Objektdefinition wird abgeschlossen, indem das Objekt als Kennzahl-, Dimensions- und Informationsobjekt abgespeichert wird.

Ein *Kennzahl*-Objekt (s. Abb. 3.27, S. 78) ist ein dynamisches Objekt oder – wie in Abschnitt 2.3.3 erwähnt – ein Fakt. Es ist abhängig von den Dimensions- und Informationsobjekten, die gemeinsam mit ihm während der Reporterstellung verwendet werden (vgl. Business Objects (2004a), S. 330). Es gibt in einem Report aggregierte Werte zurück, welche in einer zuvor festgelegten Funktion definiert werden. Standardmäßig ist zwischen folgenden Aggregatfunktionen wählbar: Summe, Anzahl, Mittelwert, Minimum, Maximum. Wenn ein Kennzahl-Objekt zusammen mit anderen Objekten während der Reporterstellung verwendet wird, leitet BusinessObjects einen `Select`-Befehl mit automatisch erstellter `Group By`-Bedingung bei der Generierung ab. Diese Generierung geht auf die Regel zurück, dass alle Elemente, die nicht in die Funktion mit einbezogen werden, in der `Group By`-Bedingung erscheinen. Dies findet statt, sobald der `Select`-Befehl eine Aggregatfunktion enthält (vgl. Business Objects (2004a), S. 331). Damit ist sichergestellt, dass die Kennzahlen im Report korrekte Ergebnisse liefern, vorausgesetzt das Universums-Datenbankschema ist korrekt.

Dimensions-Objekte (s. Abb. 3.27, S. 78) sind eine der wichtigsten Elemente während der Reporterstellung (vgl. Business Objects (2004a), S. 328). Sie stellen entweder eine oder mehrere Spalten der Datenbank dar, die eine wesentliche Rolle in der Abfrage spielen, z. B. Ort, Produkt, Mitarbeiter etc. Sie können aber auch Funktionen enthalten, die die Datensätze formatieren oder manipulieren. Diese Funktionen sind allerdings nicht so umfangreich, wie bei CR.NET. Daher entstehen Einschränkungen während der Ad-hoc-Abfrage und -Analyse

Ein *Informations*-Objekt (s. Abb. 3.27, S. 78) stellt eine detailliertere Beschreibung eines Dimensions-Objektes dar und wird diesem daher zugeordnet. Ein Informations-Objekt kann nicht ohne das dazugehörige Dimensions-Objekt in einen Report eingefügt werden (vgl. Business Objects (2004a), S. 329).

Weiterhin können *Bedingungs*-Objekte eingefügt werden, die als Einschränkungen in Form von `Where`-Klauseln an Abfragen während der Reporterstellung angehängt werden können. Damit lassen sich z. B. im Vorhinein Einschränkungen für spätere Abfragen erstellen.

Anhand von Beschreibungen für jedes Objekt, die während der Definition angegeben werden können, werden zusätzlich Meta-Daten im Repository abgelegt.

Die vorgestellten Objekte werden durch Klassen als Gruppierungselement in der hierarchischen Struktur des Universums definiert und können später vom Endanwender beliebig in den Report eingefügt werden. Für multidimensionale Analysen ist es jedoch nötig, Hierarchien zu definieren, damit Operationen wie Roll-up, Drill-down, Slice & Dice und ähnliche vom Endanwender angewendet werden können. Dazu bietet BO die Möglichkeit, entweder die Standardhierarchien zu verwenden, die anhand der Klassenhierarchie abgeleitet werden oder manuell Hierarchien zu erstellen.

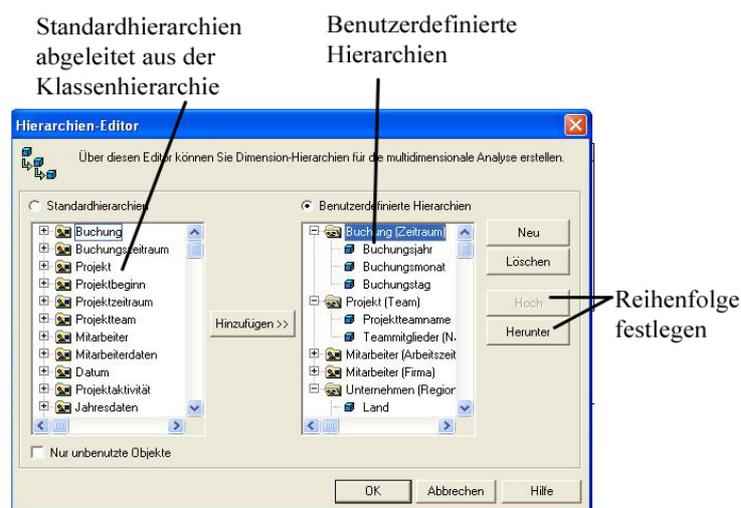


Abb. 3.29: BO Designer – Hierarchien definieren

Abb. 3.29 (s. S. 81) zeigt die manuelle Erstellung von benutzerdefinierten Hierarchien. Der Definition zufolge ist es dem Endanwender überlassen, ob er die zur Verfügung stehenden Hierarchien nutzt oder nicht. Diese freie Wahl ermöglicht die Ad-hoc-Abfrage und -Analyse nach Wünschen des Endanwenders.

Bei der Erstellung des Universums für das Projektmanagement hatten wir das Problem, dass die Projektstrukturhierarchie rekursiv in einer Tabelle (`_PROJECT_STRUCTURE`) durch eine Parent-Child-Beziehung⁶⁴ gespeichert ist (s. Tab. 3.2, S. 82).

Dadurch ist es in BO nicht möglich, die Hierarchie der einzelnen Projekte und ihrer Aktivitäten herauszulesen. Eine Möglichkeit, diese Hierarchie im BO Designer nachzubilden, ist die Erstellung von Alias-Tabellen⁶⁵. Diese Lösung zeigte uns jedoch bei Abfragen bestimmter Hierarchieebenen nur Elemente an, die bis zu dieser Ebene verzweigt waren, d. h. es wurden zu wenige Daten angezeigt. Daher ist dies eine unzureichende Lösung.

⁶⁴ Eine Tabelle enthält eine eindeutige ID und eine Parent-ID, welche die eindeutige ID seines Vaters enthält. Eine N:1-Beziehung von ID zu Parent-ID spiegelt die Parent-Child-Beziehung wieder. Somit hat jede Tabelle einen Vater und kann mehrere Kinder haben.

⁶⁵ Es müsste jedoch für jede Hierarchieebene eine Alias-Tabelle erstellt werden.

Tab. 3.2: Vereinfachte Ausgangstabelle: *PROJECT_STRUCTURE*

| <i>ID</i> | <i>Elementname</i> | <i>Parent_ID</i> | <i>Elementtyp</i> |
|-----------|----------------------------------|------------------|-------------------|
| 1 | 2006 | NULL | Jahr |
| 2 | PMM | 1 | Projekt |
| 3 | PMM – Auftragsklärung | 2 | Teilprojekt |
| 4 | PMM – Softwareplanung | 2 | Teilprojekt |
| 5 | PMM – Meetingprotokoll erstellen | 3 | Aktivität |
| 6 | PMM – UML Diagramm erstellen | 4 | Aktivität |

Eine andere Möglichkeit ist die Auflösung der Hierarchie durch einen ETL-Prozess. Dadurch werden die in der Tabelle enthaltenen Daten in eine hierarchische Tabelle geschrieben (s. Tab. 3.3, S. 82). Diese Hierarchie-Tabelle ist nach dem gleichen Prinzip wie eine Dimensionstabelle im Star Schema aufgebaut. Für jede Hierarchieebene werden alle Daten der jeweiligen Ebenen inklusive eigener Schlüssel angegeben. Es ist deutlich die Redundanz der Daten zu erkennen, die in operativen Datenbanken verhindert, jedoch in multidimensionalen, relationalen Datenbanken angestrebt wird.

Tab. 3.3: Vereinfachte Zieltabelle ohne Elementtyp: *PROJECT_STRUCTURE*

| <i>ID</i> | <i>level</i> | <i>L3</i> | <i>L2</i> | <i>L1</i> | <i>L0</i> | <i>L3_Name</i> | <i>L2_Name</i> | <i>L1_Name</i> | <i>L0_Name</i> |
|-----------|--------------|-----------|-----------|-----------|-----------|----------------|----------------|-----------------------|----------------------------------|
| 1 | 3 | 1 | 2 | 3 | 4 | 2006 | NULL | NULL | NULL |
| 2 | 2 | 1 | 2 | 3 | 4 | 2006 | PMM | NULL | NULL |
| 3 | 1 | 1 | 2 | 3 | 4 | 2006 | PMM | PMM – Auftragsklärung | NULL |
| 4 | 1 | 1 | 2 | 3 | 4 | 2006 | PMM | PMM – Softwareplanung | NULL |
| 5 | 0 | 1 | 2 | 3 | 4 | 2006 | PMM | PMM – Auftragsklärung | PMM – Meetingprotokoll erstellen |
| 6 | 0 | 1 | 2 | 3 | 4 | 2006 | PMM | PMM – Softwareplanung | PMM – UML Diagramm erstellen |

Aufgrund des Fehlens eines ETL-Tools war es uns nicht möglich, einen ETL-Prozess zu erstellen, der die Transformation durchführt. Daher wurde – nach Absprache mit den Abteilungsleitern – vorerst auf die Implementierung der Projekthierarchie verzichtet.

Das Universum enthält nun eingeschränkte Objekte (Projekt, Teilprojekt, Aktivität etc.), die einen Strukturelementtyp abbilden. Damit ist es für den Endanwender möglich, Abfragen zu jedem einzelnen Typen, jedoch nicht zu mehreren Typen durchzuführen, d. h. es können Aussagen über Kosten der Aktivitäten, der Projekte oder der Teilprojekte etc. getroffen werden, aber nicht über hierarchische Beziehungen dieser Elemente untereinander.

Dieses Problem der rekursiven Hierarchietabelle hatten wir auch bei der Universumserstellung für die Ressourcenplanung, da die Organisationsstruktur in einer ähnlichen rekursiven Tabelle abgelegt ist. Hier sind wir nach dem gleichen Prinzip verfahren. Es ist im Prinzip nur eine Ad-hoc-Abfrage über einzelne Organisationseinheiten (entweder Unternehmen oder Bereich etc.) möglich.

Ein weiteres Problem bestand darin, dass wir durch die zugrunde liegende operative Datenbankstruktur keine echten Hierarchien zur multidimensionalen Analyse abbilden konnten. Hierarchien konnten wir nur teilweise umsetzen, z. B. haben wir im Universum für die Ressourcenplanung eine Software-Hierarchie (Software -> Lizenz) und eine Standort-Hierarchie für die Organisationsstruktur (Land -> Stadt -> Straße) sowie im Universum für das Projektmanagement eine Buchungszeitraum-Hierarchie (Jahr -> Monat -> Tag) nachgebildet. Dadurch wird u. a. ein Drill-down von Software auf Lizenz bei der Ad-hoc-Analyse möglich.

Nach der Definition aller Klassen, Objekte und Hierarchien kann zum ausführlichen Test des Universums übergegangen werden.

2e) Universum testen

Nach Abschluss der Implementierung des Universums im BO Designer, muss dieses gründlich getestet werden. Dazu ist zu empfehlen, es den späteren Endanwendern zum Testen zur Verfügung zu stellen. Durch Abfragen, die die realen Einsatzbedingungen widerspiegeln, sollten Fehler in der Universumstruktur oder den Objekten auftreten. Diese Fehler sind anschließend von den Entwicklern zu prüfen und zu entfernen. Ein erneutes Ausführen des Testdurchlaufs mit den Endanwendern soll die Absicherung der Fehlerkorrektur und die korrekte Implementierung eines virtuellen DWH sicherstellen. Diese Phase haben wir mehrmals durchlaufen, um die Anforderungen der Abteilungs- und Projektleiter in den Universen abzubilden.

3) Universum mit Repository implementieren

Für den unternehmensweiten Zugriff muss das Universum im Repository des Unternehmens gespeichert werden. Das Repository ist in drei Domänen aufgeteilt, die nur für festgelegte Gruppen und Benutzer Zugriffsberechtigungen erteilen. Für die Universen ist ein Domänenbereich vorhanden, der wiederum in Unterdomänen aufgeteilt werden kann, die mit Zugriffsberechtigungen kontrolliert werden. Damit wird zusätzlich zur Integrität des Universums die Datensicherheit und Zugriffskontrolle geregelt. Jeder Mitarbeiter, der Zugriff auf die Domänen bzw. eine einzelne Domäne hat, kann die verdichteten Daten aus den Universen für seine Analysen nutzen. Daher ist es wichtig, die Universen und deren Verwendung ausführlich zu kommentieren und Beschreibungen sowie Meta-Daten abzulegen, um jedem Mitarbeiter die Nutzung zu vereinfachen. Diese Meta-Daten werden zusätzlich im Repository gespeichert. Die Speicherung der Universen im Unternehmens-Repository verfolgt den BI-Gesamtansatz, der jedoch beim betrachteten Automobilhersteller zusätzlich mit weiteren Maßnahmen angestrebt wird. Diese werden an späterer Stelle angeschnitten.

4) Wartung

Um den wachsenden Datenquellen und erweiterten Anforderungen der Endanwender gerecht zu werden, ist je nach Änderungen der Datenquelle oder der Anforderungen eine Wartung des Universums durchzuführen.

Die vorgestellte Vorgehensweise macht es möglich, aus verschiedenen internen oder externen Datenquellen Daten als Informationen für die Endanwender bzw. Entscheider aufzubereiten und zur Verfügung zu stellen.

Für PMM stellen sich diese Informationen in den drei folgenden Universen dar: „PMM-Projektmanagement“ (Entwicklersicht: Abb. 3.30, S. 85; Endanwendersicht: Abb. 3.42, S. 98), „PMM-Mitarbeiterkapazitätenplanung“ (Entwicklersicht: Abb. C.1, S. 130; Endanwendersicht: Abb. 3.42, S. 98) und „PMM-Ressourcenplanung“ (Entwicklersicht: s. Abb. C.2, S. 130; Endanwendersicht: Abb. 3.43, S. 99). Diese können mit Hilfe von BO Web- und Desktop-Intelligence aufgerufen werden. Dazu haben wir in PMM eine Schnittstelle implementiert, die zugriffsberechtigten Mitarbeitern den Zugang zu den Universen und damit zu Ad-hoc-Analysen bietet. Das Universum zum Projektmanagement haben wir für Abfragen zur in Abschnitt 3.2.1 erwähnten Planungs- und Budgetierung erstellt, um dadurch Aussagen über verschiedene Kosten der Projekte zu erhalten.

Für die Mitarbeiterkapazitätenplanung haben wir ein Universum für Abfragen über die Projekt- und Arbeitszeiten der Mitarbeiter sowie für Abfragen über die gespeicherten Unternehmen und Dienstleister erstellt.

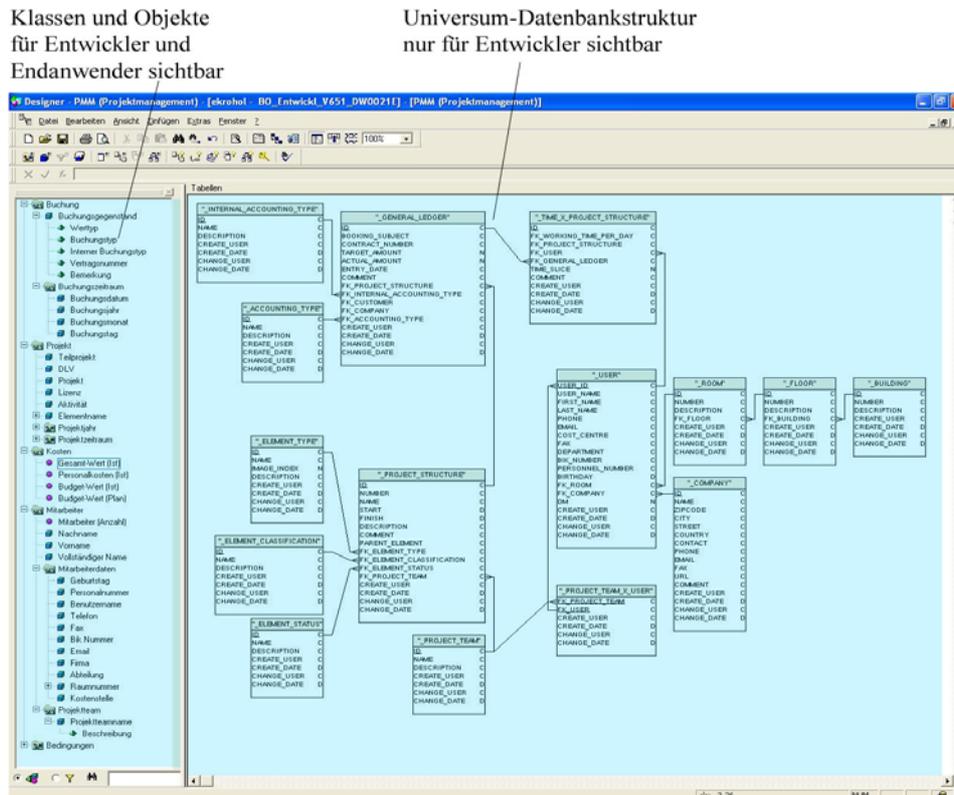


Abb. 3.30: BO Designer – Universum Projektmanagement

Das Universum zur Ressourcenplanung haben wir entwickelt, um darauf aufbauend Abfragen über die gespeicherte Hardware und Software sowie ihrer Lizenzen und den damit verbundenen Kosten zu erstellen.

Resümierend lässt sich festhalten, dass in diesem Abschnitt deutlich wurde wie die operativen Daten der PMM Datenbank durch die vorgestellten Maßnahmen entscheidungsorientiert aufbereitet wurden. Dies war erforderlich, um den Endanwendern Standard-Reports sowie für Ad-hoc-Abfragen und -Analysen zur Entscheidungsunterstützung bereitzustellen. Einerseits haben wir die Daten mit Hilfe von Manipulationen und Formattierungen in CR.NET für Standard-Reports und andererseits mit Hilfe von Universen im BO Designer für Ad-hoc-Abfragen und -Analysen vorbereitet. Im Anschluss werden die aufbereiteten Daten im Logik-Layer zu Informationen generiert, da sie in einen Bedeutungszusammenhang gebracht werden. Diese Informationen können zur effektiven Entscheidungsunterstützung verwendet werden. Daher wird im Weiteren auf den PMM-Präsentations- bzw. -Logik-Layer eingegangen, der auf die aufbereiteten Daten des Daten-Layers zurückgreift, Informationen für die Entscheider generiert und ihm den benutzerfreundlichen Zugriff zu diesen gewährt.

3.3.2 PMM-Präsentations- bzw. Logik-Layer

Verallgemeinert lässt sich sagen, dass im Logik-Layer Informationen für den Entscheider generiert und bearbeitet sowie Systeme zur Speicherung und Verteilung dieser Informationen bereitgestellt werden. Der Logik-Layer bietet zur Informationsgenerierung und systematischen Untersuchung von betriebswirtschaftlichen Zusammenhängen Systeme, die MSS, an. Bei MSS kann es sich sowohl um die in 2.4.1 erwähnten konzeptorientierten Systeme als auch um die in 2.4.2 vorgestellten generischen Basissysteme handeln.

Der Präsentations-Layer greift – dem in Kapitel 2.2 vorgestellten BI-Ordnungsrahmen entsprechend – in Form eines BI-Portals auf den Logik-Layer zu. Im Falle von PMM existiert allerdings aufgrund der Abteilungslösung kein BI-Portal. Der Präsentations-Layer von PMM ist eng mit dem Logik-Layer verknüpft und greift auf die integrierten BI-Komponenten zu. Im Logik-Layer werden Aspekte der konzeptorientierten Systeme mit Aspekten der generischen Basissysteme verknüpft und dem Entscheider (Projekt- und Abteilungsleiter) als zusammenhängende Anwendung zur Verfügung gestellt, um dadurch Informationen generieren und abrufen zu können. Daher kann dies ebenso als Präsentations-Ebene verstanden werden, da dem Endanwender der Zugriff zu den Informationen über PMM erteilt wird.

Wie bereits in 3.2 erwähnt, bilden die Grundfunktionalitäten von PMM das betriebswirtschaftliche Konzept der Planung und Budgetierung ab und unterstützen den Projekt- und Abteilungsleiter bei der Projekt- und Ressourcenplanung. Die zusätzlich integrierten BI-Komponenten bieten Funktionalitäten der generischen Basissysteme, indem Reporting- und Ad-hoc-Analyse-Möglichkeiten zur Verfügung stehen.

Im folgenden Verlauf werden die Informationsgenerierung und -darstellung der Standard-Reports und Ad-hoc-Analyse beschrieben.

Standard-Reports

In PMM haben wir – wie in 3.3.1 beschrieben – durch CR.NET vordefinierte Standard-Reports in die Anwendung implementiert. Dadurch sind diese für den Endanwender direkt aus PMM abrufbar. Einige Reports haben wir so erstellt, dass sie den Endanwender noch kleinere Interaktionsmöglichkeiten durchführen lassen, wie z. B. den Datenzeitraum für den Zeiterfassungsreport auswählen (s. Abb. 3.31, S. 87) oder die Sortierung der Hardware vorgeben (s. Abb. 3.38, S. 93).

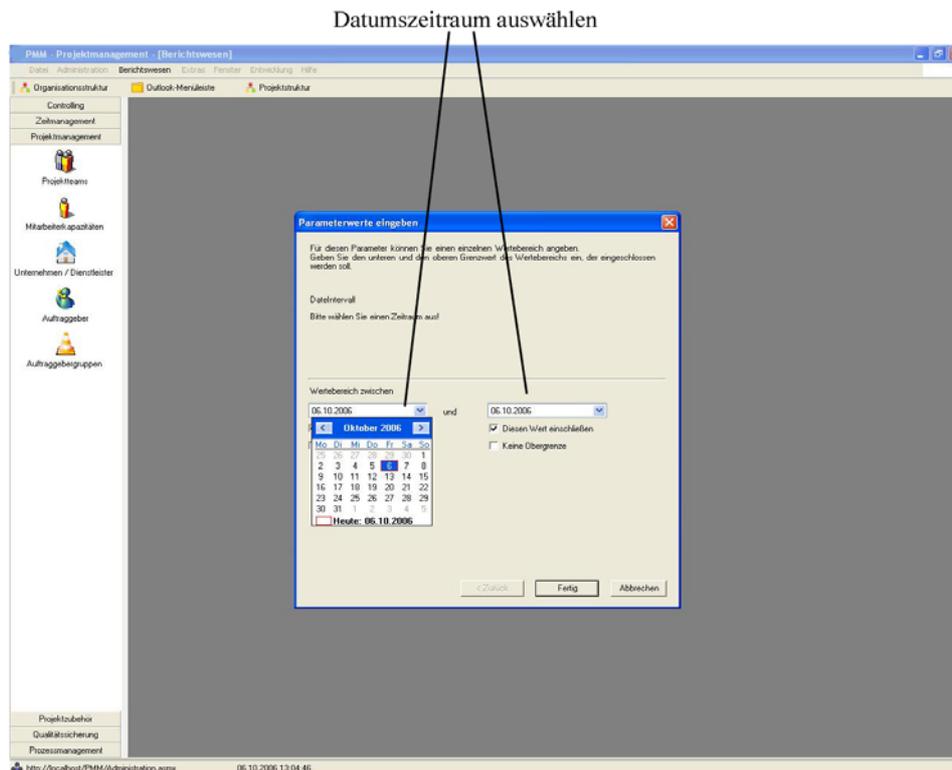


Abb. 3.31: Standard-Report – Intervallauswahl

Im PMM-Daten-Layer haben wir die Daten so aufbereitet, dass zur Laufzeit eine SQL-Anfrage generiert wird, um alle im Report enthaltenen Datensätze aus den Datenbanken zu erhalten. Die Beziehungen zwischen den Tabellen werden nach der im Verknüpfungsassistent festgelegten Weise (INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN oder FULL OUTER JOIN) durchgeführt.

Eine solche SQL-Anfrage sieht für den Zeiterfassungsreport (s. Abb. 3.33, S. 89) aus dem Anwendungsbeispiel – das im Daten-Layer (s. Abschnitt 3.3.1) vorgestellt wurde – folgendermaßen aus:

```

SELECT  _WORKING_TIME_PER_DAY.WORK_DATE,
        _WORKING_TIME_PER_DAY.START,
        _WORKING_TIME_PER_DAY.FINISH,
        _WORKING_TIME_PER_DAY.OVERTIME,
        _WORKING_TIME_PER_DAY.FLEXITIME,
        _WORKING_DAY_SHORTCUT.NAME,
        _WORKING_TIME_PER_DAY.PAUSE_TIME,
        _WORKING_TIME_PER_DAY.COMMENT, _USER.FIRST_NAME,
        _USER.LAST_NAME, _USER.USER_ID,
        _USER_DATA_PER_YEAR.DATA_YEAR,
        _USER_DATA_PER_YEAR.HOLIDAY_ENTITLEMENT
FROM    (pmm.dbo._WORKING_DAY_SHORTCUT _WORKING_DAY_SHORTCUT
        INNER JOIN (pmm.dbo._USER _USER
        INNER JOIN pmm.dbo._WORKING_TIME_PER_DAY
        _WORKING_TIME_PER_DAY
        ON _USER.USER_ID=_WORKING_TIME_PER_DAY.FK_USER)
        ON _WORKING_DAY_SHORTCUT.ID=
        _WORKING_TIME_PER_DAY.FK_WORK_SHORTCUT)
        INNER JOIN pmm.dbo._USER_DATA_PER_YEAR
        _USER_DATA_PER_YEAR
        ON _USER.USER_ID=_USER_DATA_PER_YEAR.FK_USER
ORDER BY _USER_DATA_PER_YEAR.DATA_YEAR

```

Hierbei ist zu erkennen, dass zuerst die nötigen Datensätze mit SELECT . . . FROM aus den dazugehörigen Tabellen ausgewählt und anschließend die Gruppen mit ORDER BY sortiert werden. Hier haben wir INNER JOIN verwendet, da lediglich die Datensätze verlangt werden, bei denen die Fremdschlüssel mit den Primärschlüsseln der verschiedenen Tabellen übereinstimmen:

```

_USER.USER_ID = _WORKING_TIME_PER_DAY.FK_USER
_WORKING_DAY_SHORTCUT.ID =
    _WORKING_TIME_PER_DAY.FK_WORK_SHORTCUT
_USER.USER_ID = USER_DATA_YEAR.FK_USER

```

Nach Erhalt der Datensätze werden diese zur Laufzeit an die gewünschten Formattierungen angepasst und alle Berechnungen des jeweiligen Reports durchgeführt, um diese anschließend anzeigen zu können.

Über die Funktion „Berichtswesen“ im Programmmenü kann der Endanwender einen der verfügbaren Standard-Reports auswählen und anzeigen lassen (s. Abb. 3.32, S. 89).



Abb. 3.32: PMM Berichtswesen Auswahlmü

Für die Anwendung PMM haben wir folgende Standard-Reports implementiert:

- *Zeiterfassung*: Für die übersichtliche Ausgabe der Zeiterfassung haben wir verschiedene Reports implementiert, die die täglichen Arbeitszeiten inklusive der Gleit- und Mehrarbeitszeiten der Mitarbeiter ausdrucken.

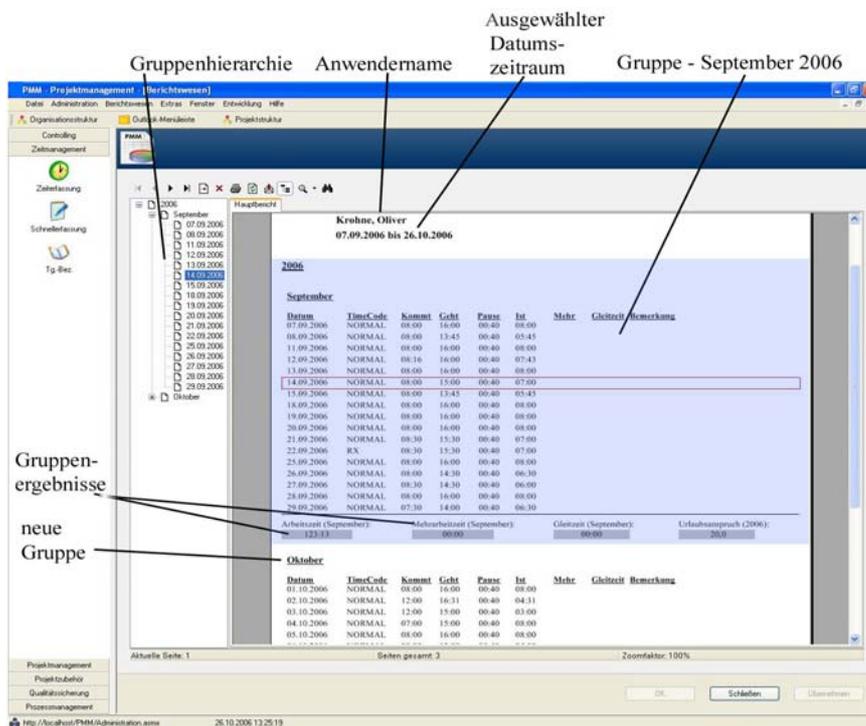


Abb. 3.33: Standard-Report – Zeiterfassung/ Intervall/ Gruppierung

Es existiert ein Report, der die kompletten Arbeitszeiten ausdrückt. Ein weiterer Report listet nur die Arbeitszeiten des letzten Monat auf (s. Abb. D.1, S. 131). Weiterhin existiert ein Report, bei dem der Endanwender ein Zeitintervall bestimmt (s. Abb. 3.31, S. 87), für das die Arbeitszeiten ausgedruckt werden sollen (s. Abb. 3.33, S. 89).

Die Reports haben wir durch die Datenaufbereitung zuerst übersichtlich nach Jahren und anschließend nach Monaten (ausgenommen der Monatsausdruck) gruppiert. Weiterhin enthalten sie dadurch am Ende jedes Monats bzw. Jahres eine Zusammenfassung und am Ende des Reports eine Gesamtzusammenfassung. Warnmeldungen zeigen an, dass die Gleitzeit oder der Urlaubsanspruch ins Negative fallen (s. Abb. 3.34, S. 90).

Warnung - negative Gleitzeit

| | | | | | |
|-------------------------|--------|-------|-------|-------|-------|
| 20.07.2006 | NORMAL | 07:15 | 16:00 | 08:45 | |
| 21.07.2006 | TU | | | | |
| 22.07.2006 | TU | | | | -2:00 |
| 26.07.2006 | NORMAL | 07:00 | 17:00 | 10:00 | |
| 27.07.2006 | NORMAL | 07:00 | 15:00 | 08:00 | |
| Arbeitszeit (Juli): | | 66:01 | | | |
| Mehrarbeitszeit (Juli): | | 00:00 | | | |
| Gleitzeit (Juli): | | | | -2:00 | |

Abb. 3.34: Standard-Report – Zeiterfassung/ Warnung

- **Projektzeiten:** Um eine Übersicht für den Abteilungs- bzw. Projektleiter zu erstellen, können Reports ausgewählt werden, die entweder die Verteilung der für das Jahr festgelegten Projektzeiten oder die bereits geleisteten Projektzeiten der Mitarbeiter anzeigen. Der Report für die jahresbezogenen Projektaktivitäten und -zeiten ist nach Jahren und Mitarbeitern gruppiert. Er zeigt an, welche Projektaktivitäten ein Mitarbeiter in einem Jahr durchführen und welchen Zeitaufwand er dafür jeweils verwenden darf. Für jeden Mitarbeiter sind die prozentualen Anteile seiner Aktivitäten in einem Kreisdiagramm und seine jahresbezogenen Daten in Listenform übersichtlich dargestellt (s. Abb. 3.35, S. 90).

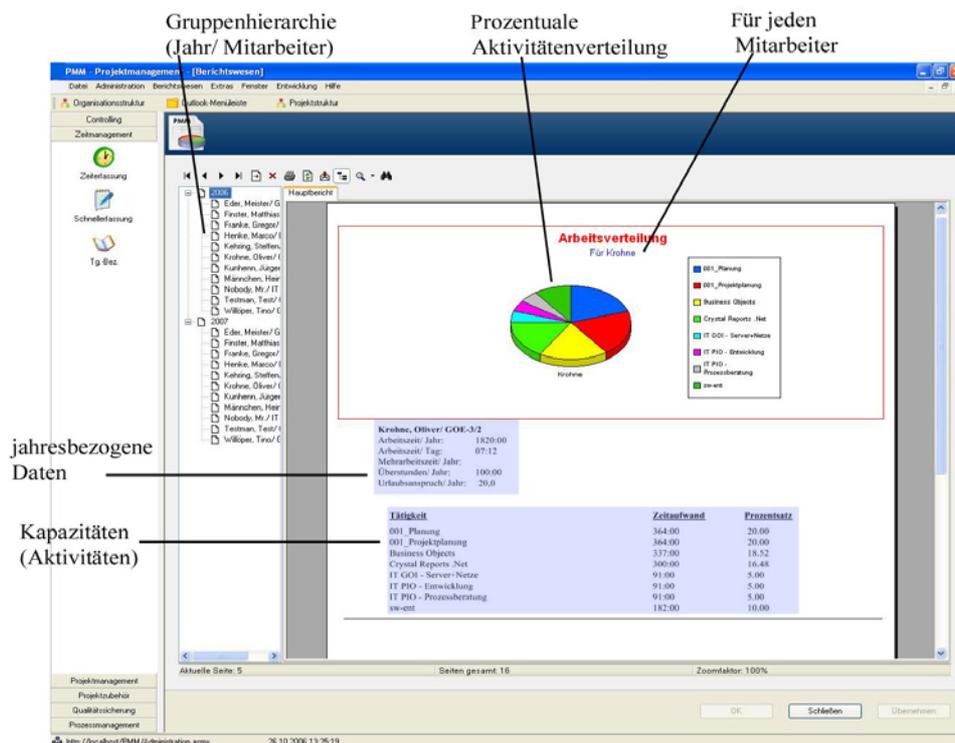


Abb. 3.35: Standard-Report – Projektarbeitszeiten der Mitarbeiter (jahresbezogen)

Im Laufe des Jahres trägt jeder Mitarbeiter seine Aktivitäten über die Zeiterfassung ein. Dadurch werden die jeweiligen Personalkosten für die einzelnen Projekte verbucht. Die tatsächlich abgeleisteten Projektaktivitäten kann der Projektleiter über einen weiteren Standard-Report abrufen, der zu jedem Mitarbeiter und für jeden Tag die Projektaktivitäten und die dafür verwendete tägliche Arbeitszeit ausdrückt (s. Abb. D.2, S. 131). Diesen Report haben wir während der Datenaufbereitung nach Mitarbeitern, Jahren, Monaten und sogar nach Tagen gruppiert, um dem Projektleiter eine schnelle Navigation in der Hierarchie zu ermöglichen. Am Ende jeder Gruppe bieten Aggregationsfunktionen (hier Summation) einen Überblick über die in dieser Gruppe angefallene Arbeits- und Projektzeit sowie die entstandenen Personalkosten.

- *Software/Abteilung*: Für die Ressourcenplanung ist es möglich, einen Report abzurufen, der eine Übersicht über alle Organisationseinheiten und die von ihnen verwendete Software enthält.

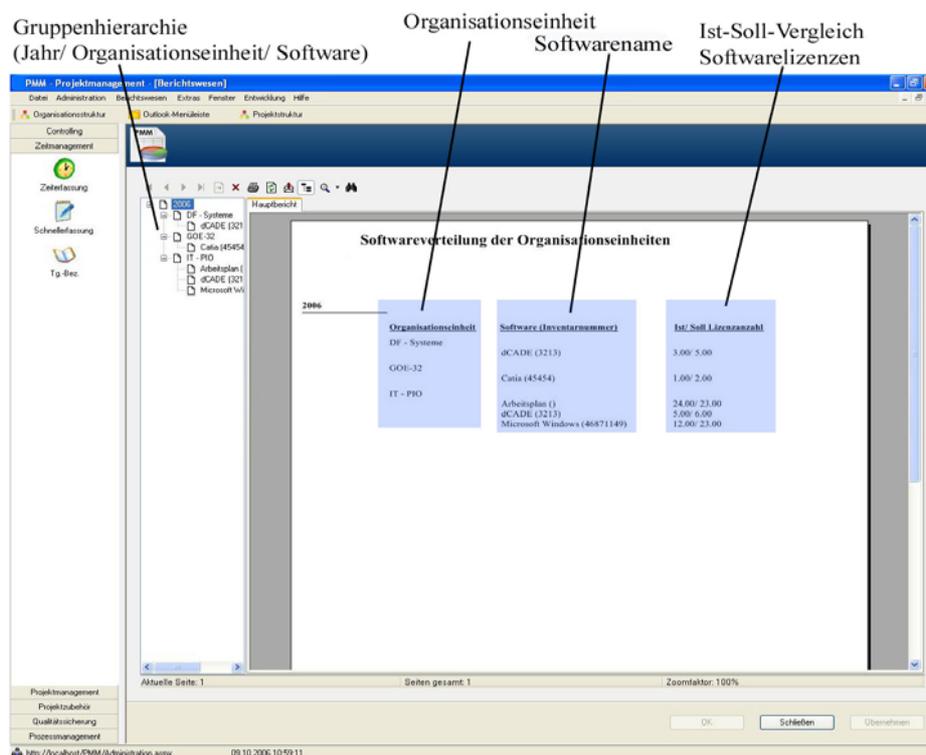


Abb. 3.36: Standard-Report – Software/ Organisationseinheit

Den Report haben wir nach Jahren, Abteilungen und nach Software gruppiert. Dadurch können zu jeder Software und jeder Abteilung die Plan- und Ist-Lizenzen angezeigt werden (s. Abb. 3.36, S. 91).

- *Softwarelizenzen*: Eine weitere Übersicht zur Ressourcenplanung bietet der Report „Software“. Er listet alle Softwareprodukte, deren Lizenzen und die damit verbundenen Anschaffungs- und Wartungskosten auf. Hierbei werden für jede Software die Gesamtkosten errechnet. Der Report ist leicht überschaubar nach Softwarenamen sortiert und gibt zu jeder Software den jeweiligen Verantwortlichen mit seinen Kontaktdaten an.

Gruppenhierarchie

Für jede Software - Lizenzkostenübersicht

Softwareübersicht

Software (Anzahl/Anzahl)

Arbeitsplan (12947/292410)
Verantwortlicher: Matthias Finster
Abteilung: OOE-3
Email: matthias.finster@...
Telefon: 22222
BIC: 23564

| Lizenznummer | Anzahl Lizenztyp | Anschaffungskosten | Wartungskosten |
|------------------------|-------------------|---------------------|--------------------|
| 12324343545 | 20/ FLOATING | 3000,00 € | 15,00 € |
| Gesamtergebnis: | Anzahl: 20 | 100'000,00 € | 15'000,00 € |

Calls (45454)
Verantwortlicher: Oliver Krohne
Abteilung: OOE-3
Email: oliver.krohne@...
Telefon: 39039
BIC:

| Lizenznummer | Anzahl Lizenztyp | Anschaffungskosten | Wartungskosten |
|------------------------|------------------|--------------------|-------------------|
| 3455 | 1/ FLOATING | 34'550,00 € | 10,00 € |
| Gesamtergebnis: | Anzahl: 1 | 34'920,00 € | 3'455,00 € |

4CADE (3213)
Verantwortlicher: Matthias Finster
Abteilung: OOE-3
Email: matthias.finster@...
Telefon: 22222
BIC: 23564

| Lizenznummer | Anzahl Lizenztyp | Anschaffungskosten | Wartungskosten |
|------------------------|------------------|---------------------|--------------------|
| 654564 | 1/ FLOATING | 20'000,00 € | 20,00 € |
| 465465 | 5/ FLOATING | 20'000,00 € | 10,00 € |
| 15131-4469-4646 | 3/ NODELOCKED | 1'000,00 € | 1'000,00 € |
| Gesamtergebnis: | Anzahl: 9 | 400'000,00 € | 59'000,00 € |

Gruppen-
ergebnisse

Abtuelle Seite: 1 Seiten gesamt: 2 Zoomfaktor: 100%

OK Schließen Übersichten

Druckvorbet Oliver.krohne.odt - OpenOffice.org Writer

Abb. 3.37: Standard-Report – Software/ Lizenzen

- *Hardware*: Der Report listet die Abteilungshardware leicht erfassbar auf. Diesbezüglich haben wir drei verschiedene Reports erstellt.

Ein Report sortiert die Hardware nach Namen, ein anderer nach Typ (Drucker, Computer, etc.) und ein Weiterer nach Abschreibungsdauer (s. Abb. 3.38, S. 93). Weiterhin werden zu jeder Hardware der Benutzer und der Standort angegeben.

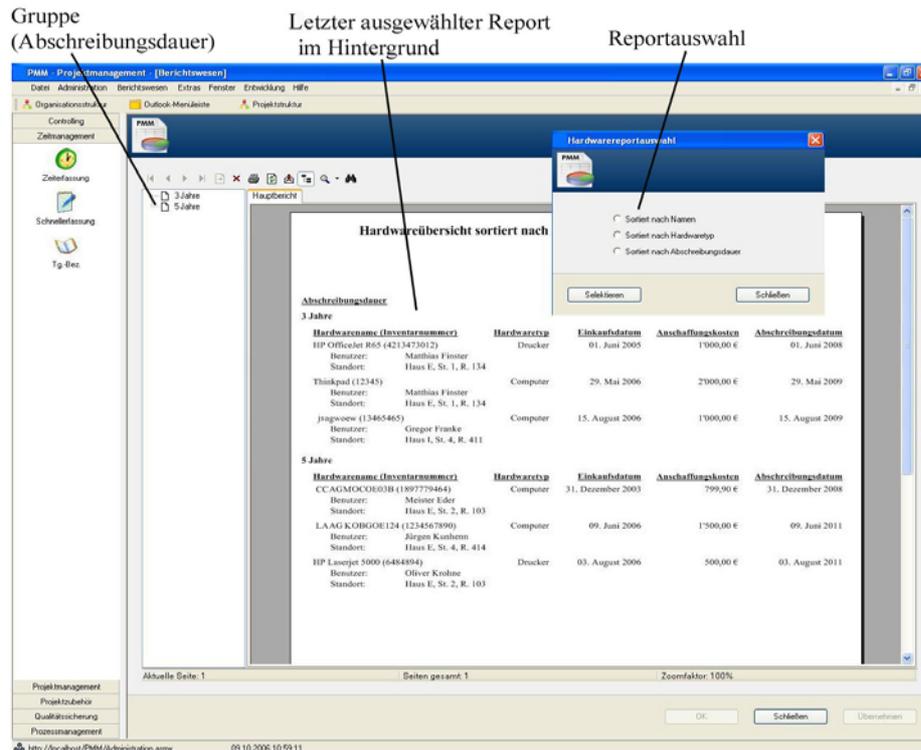


Abb. 3.38: Standard-Report – Hardware

- **Standorte:** Der Report „Standort“ bietet einen kompletten Überblick über alle Räume und die sich darin befindenden Mitarbeiter und verfügbaren Netzwerkanschlüsse.

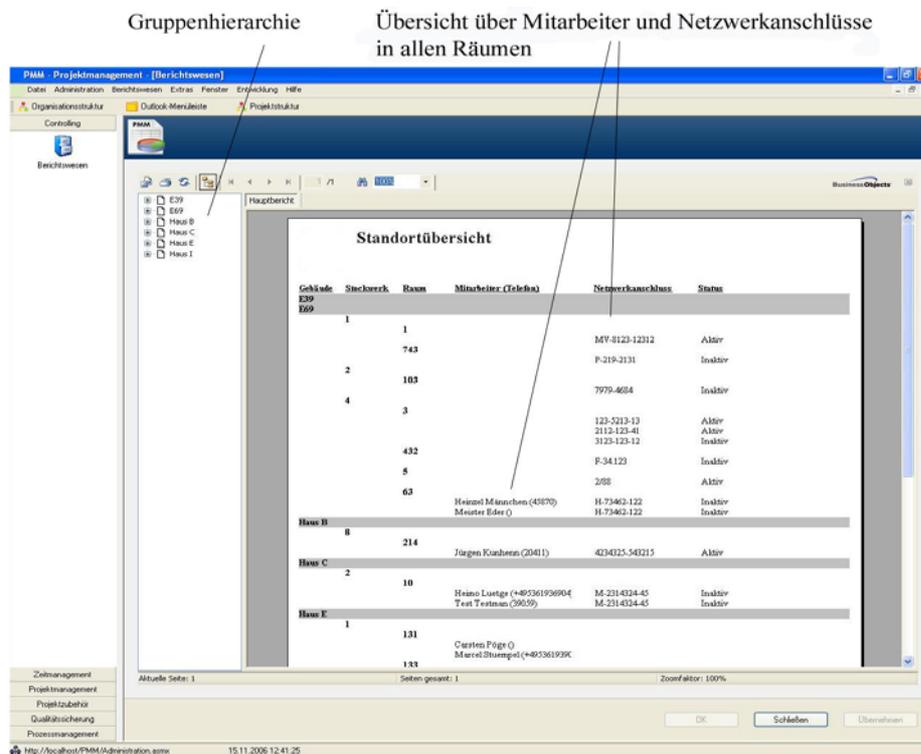


Abb. 3.39: Standard-Report – Standort

Den Report haben wir im Daten-Layer nach Gebäuden, Stockwerken und nach Räumen gruppiert, um eine Übersicht zu gewährleisten und bei Bedarf nur einzelne Gruppen anzuzeigen.

- *DLV-Status*: Dienstleistungsvereinbarungen (DLVs) sind interne Buchungssätze, die abgearbeitet werden müssen. In PMM werden diese DLVs als Aktivitäten in die Projektstruktur aufgenommen und können verschiedene Stati (In Bearbeitung, abgeschlossen etc.) annehmen. Dieser Report bietet eine einfache Listenübersicht aller noch nicht abgeschlossenen DLVs.

Für die beschriebenen Reports stehen dem Endanwender zusätzliche Funktionalitäten bereit. Die Reports, die wir nach Kategorien gruppiert haben, erhalten Drill-down-Operationen. Nach Auswahl der Gruppe durch die Drill-down Operation wird diese in einem separaten Report in dem gleichen Format angezeigt. Jeder Report bzw. Drill-down-Report kann vergrößert, gedruckt oder als E-Mail versendet werden. Weiterhin besteht die Möglichkeit, einen Report nach bestimmten Begriffen zu durchsuchen oder ihn ins MS Excel-, MS Word-, PDF-, HTML-, XML- oder RTF-Format zu exportieren.

Durch die Integration von CR.NET Standard-Reports ist es dem Endanwender möglich, Informationen über diverse Zusammenhänge der Daten zu erlangen, indem er eine leicht verständliche Darstellung erhält. Außerdem bieten sie denselben Funktionsumfang wie die ursprünglichen statischen Reports aus Peanuts und Time und erweitern sie durch dynamische Interaktionsmöglichkeiten, Drill-down-Operationen, Exportfunktionalitäten und zusätzliche Reports. Standard-Reports sind jedoch zur Analyse der kompletten betriebswirtschaftlichen Zusammenhänge nicht ausreichend und werden daher durch Ad-hoc-Analysen unterstützt.

Ad-hoc-Analysen

Neben den Standard-Reports ist es möglich, auf den Daten von PMM Ad-hoc-Abfragen und -Analysen auszuführen. Dazu haben wir die Daten im PMM-Daten-Layer derart aufbereitet, dass der Endanwender bzw. Entscheider nun mit Hilfe von BO aus diesen Daten Informationen generieren kann. Über den Button „Controlling“ in der Outlook-Menüleiste von PMM (s. Abb. 3.40, S. 95) ist es möglich, BO Webintelligence oder – falls auf dem Rechner des Endanwenders vorhanden – BO Desktop Intelligence aufzurufen.



Abb. 3.40: PMM – Controlling Auswahlbutton

Die BO Entwickler verwalten die Zugriffsrechte der Benutzer. Der Funktionsumfang ist vergleichbar mit den Funktionalitäten von BO Desktop Intelligence, ist jedoch nur für die Abfrage und Analyse von Daten vorgesehen. Nach Identifikation des Endanwenders bei BO Webintelligence oder Desktop Intelligence stehen ihm die Universen innerhalb seiner Domäne zur Verfügung. Je nach Anwendungsfall wählt er das entsprechende Universum für umfangreiche Ad-hoc-Abfragen und Analysen über „Datei -> Neu“ und mit Hilfe des Assistenten aus. Anschließend stehen ihm für die Erstellung von Abfragen alle Klassen und Objekte, die während der Universum-Entwicklung erstellt wurden, zur Verfügung.

Im Allgemeinen können beliebig viele Elemente aus der Klassenhierarchie durch einen Doppelklick in die Abfrage eingefügt werden. Kennzahl-, Dimensions- und Informationsobjekte werden in den oberen Abfragebereich eingefügt, Bedingungen in den unteren (s. Abb. 3.41, S. 96). Bedingungen werden zusätzlich durch UND oder ODER getrennt. Weiterhin ist es realisierbar, zusätzliche Restriktionen den Abfragen hinzuzufügen, die nicht im Universum definiert wurden. Dazu wird ein Objekt per Drag-and-Drop in den Bedingungsbereich gezogen. Nun werden dem Endanwender vordefinierte Funktionen (GLEICH, GRÖßER GLEICH, UNGLEICH, etc.) angeboten, um die Bedingung zu definieren. Anschließend kann ein Wert aus der Datenbank ausgewählt oder ein konkreter Wert eingetragen werden (s. Abb. 3.42, S.98).

Jedes Objekt kann weiterhin sortiert oder gruppiert werden. Bei Kennzahl-Objekten können auch Zwischen- und Endergebnisse in Form von Aggregationsfunktionen eingefügt werden. Es stehen Summen-, Anzahl-, Mittelwert-, Minimum-, Maximum- und Prozentberechnung zur Verfügung.

Für multidimensionale Analysen können Hierarchien zur Abfrage hinzugefügt werden, indem diese über das Feld „Hierarchie“ ausgewählt werden. Dies setzt jedoch voraus, dass die Hierarchien im Vorfeld im Universum durch die Entwickler definiert wurden. Für BO Webintelligence und Desktop Intelligence stehen dem Endanwender die multidimensionalen Operationen Pivotierung, Drill-down, Roll-up sowie Slice & Dice zur Verfügung. Die Pivotierung findet schon während der Erstellung der Abfrage statt, da der Datenwürfel bereits hier in eine benutzerdefinierte Position gedreht wird, indem die Reihenfolge der Objekte in der Abfrage festgelegt wird. Diese kann jedoch während der Analyse mit Slice & Dice nochmals geändert werden.

Für die im Zusammenhang mit der vorliegenden Diplomarbeit erstellten Universen ist es nicht möglich, multidimensionale Analysen entsprechend der Definition aus Abschnitt 2.4.2 durchzuführen, da wir aufgrund der operativen Datenbank keine multidimensionalen Hierarchien im BO Designer abbilden konnten. Unabhängig davon können die multidimensionalen Operationen teilweise nach Generierung des Reports angewendet werden. Dieses Verfahren wird an späterer Stelle beschrieben.

Wie in 3.3.1 erwähnt, haben wir für PMM drei Universen erstellt. Das Universum „PMM-Projektmanagement“ (s. Abb. 3.41, S. 96) wird für die komplette Steuerung, Kontrolle und Analyse der Planung und Budgetierung von laufenden und zukünftigen Projekten eingesetzt.

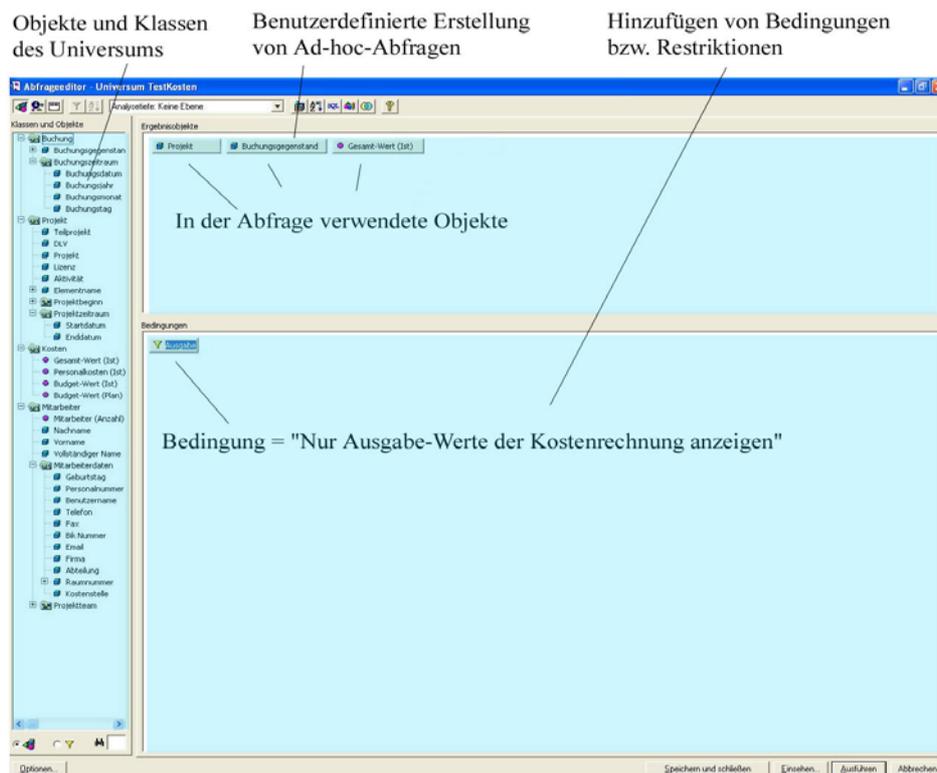


Abb. 3.41: Ad-hoc Abfragen – Projektmanagement

Die Klassen „Buchung“, „Projekt“, „Mitarbeiter“ und „Kosten“ enthalten Kennzahl- und Dimensionsobjekte, um Fragestellungen abbilden zu können. Dabei sind unter anderem Abfragen über die Ist- und Plan-Budgets der Projekte bzw. Teilprojekte etc. realisierbar, welches einen Plan-Ist-Vergleich der Projektbudgets möglich macht. Eine hierarchische Darstellung ist aufgrund der fehlenden Hierarchien nicht möglich. Es können lediglich Aussagen zu einzelnen Strukturelementen des Projektbaums getroffen werden. Ein Drill-down ist nur innerhalb der Buchungszeitraum-Hierarchie möglich, da wir diese abgebildet haben. Folglich sind aber keine multidimensionalen Analysen gemäß der Definition (s. Abschnitt 2.3.3) ausführbar. Trotzdem können Aussagen über die verursachten Personalkosten getroffen werden. Überdies können Abfragen erstellt werden, die die Gesamtkostenrechnung der auf der Datenbank gespeicherten Einnahmen, Ausgaben und dem tatsächlichen Budget widerspiegeln. Dazu können Fragestellungen wie „Ist mein Projekt bereits zu teuer?“ beantwortet werden. Wie bereits erwähnt können sich Ad-hoc-Abfragen und -Analysen nur auf einzelne Projekte, Teilprojekte, Aktivitäten etc. beziehen. Als Beispiel wird in der Ad-hoc-Abfrage aus Abb. 3.41 (s. S. 96) eine SQL-Anfrage generiert, die alle Ausgaben der Gesamtkostenrechnung nach Projekt und Buchungsgegenstand anzeigt.

Das Universum zur Mitarbeiterkapazitätenplanung (s. Abb. 3.42, S. 98) enthält Kennzahl- und Dimensionsobjekte gruppiert nach den Klassen „Mitarbeiter“ und „Unternehmen“, wobei die Klasse „Mitarbeiter“ in die weiteren Unterklassen „Mitarbeiterdaten“, „Aktivitäten“, „Tagesdaten“ und „Jahresdaten“ gegliedert ist.

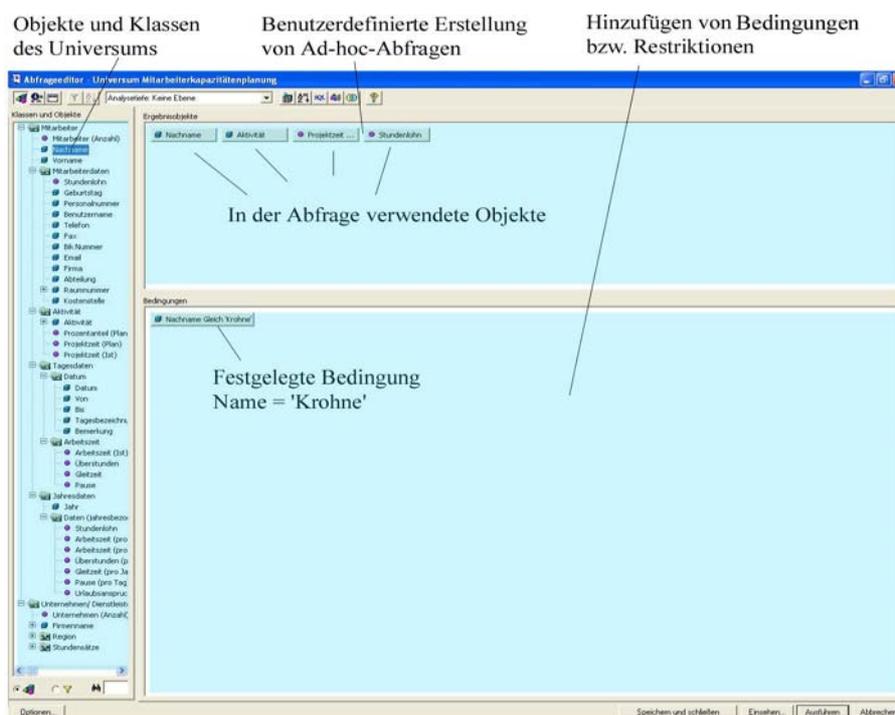


Abb. 3.42: Ad-hoc Abfragen – Mitarbeiterkapazitätenplanung

Im Universum zur Mitarbeiterkapazitätenplanung können Abfragen erstellt und analysiert werden, die Informationen über die Arbeitszeit⁶⁶ und Projektaktivitäten der Mitarbeiter zur Mitarbeiterkapazitätenplanung sowie über die gespeicherten Dienstleister geben. Im dargestellten Beispiel in Abb. 3.42 (s. S. 98) wird eine Ad-hoc-Abfrage generiert, die für den Mitarbeiter „Krohne“ den Stundenlohn und seine Aktivitäten mit den dafür verwendeten Projektzeiten anzeigen soll.

Die Kennzahl- und Dimensionsobjekte im Universum „PMM-Ressourcenplanung“ (s. Abb. 3.43, S. 99) haben wir nach den jeweiligen Ressourcen („Hardware“ oder „Software“) gegliedert.

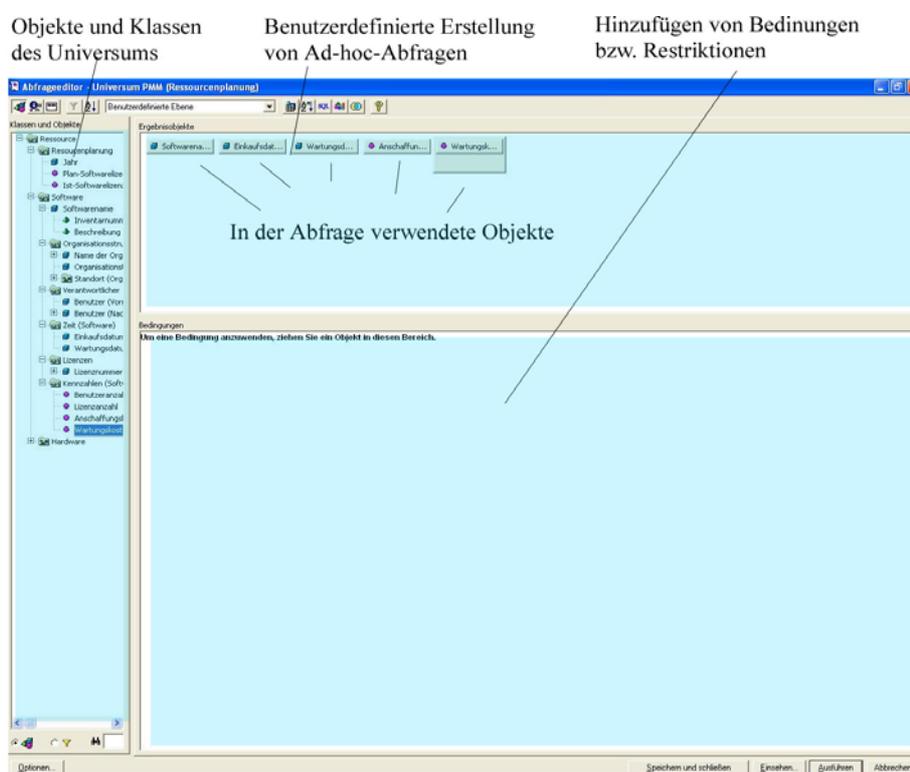


Abb. 3.43: Ad-hoc-Abfragen – Ressourcenplanung

Das Universum wird zur Überwachung und Durchführung der Ressourcenplanung von Software, deren Lizenzen und der abteilungsinternen Hardware verwendet. Für die Ressourcenplanung können zu einzelnen Organisationsstrukturelementen⁶⁷ ein Ist- und Plan-Vergleich der benötigten Softwareprodukte erstellt werden. Die Organisationsstrukturelemente können anhand der Standort-Hierarchie analysiert werden.

⁶⁶ Wie in Abschnitt 3.3.1 beschrieben, sind hier die Zeiten nur in Dezimaldarstellung abrufbar.

⁶⁷ Aufgrund der fehlenden Hierarchie können auch hier nur einzelne Elementtypen (z. B. Unternehmen oder Bereich) abgefragt werden.

Zusätzliche Aussagen können getroffen werden, indem Daten über die Anschaffungs- und Wartungskosten der Software abgefragt werden können. Die Ad-hoc-Abfrage, die in Abb. 3.43 (s. S. 99) dargestellt ist, generiert eine SQL-Anfrage, welche für jede Software das Anschaffungs- und Wartungsdatum sowie die dazugehörigen Kosten anzeigen soll. Infolge der erstellten Hierarchie für „Software -> Lizenz“ ist hier nach Erstellung des Reports ein Drill-down auf die Lizenz-Ebene möglich.

Nach erfolgreichem Erstellen der Abfrage wird der Report durch den „Ausführen“-Button generiert. Jedes Objekt in der Abfrage ist verknüpft mit einem SQL-Select-Befehl. Bei der Ausführung der Abfrage wird für jedes Objekt der Select-Befehl mit optionaler Where-Klausel als Anfrage an die Datenbank generiert und dem Endanwender als Report präsentiert. Im unten abgebildeten Beispiel (s. , S.) haben wir eine Abfrage erstellt, die für jedes Projekt, den Buchungsgegenstand (z. B. Personalkosten) und die damit verbundenen Ist-Werte der Gesamtkostenrechnung anzeigen soll. Die Gesamtkostenrechnung verrechnet das genehmigte Budget mit allen aufgetretenen Einnahmen und Ausgaben.

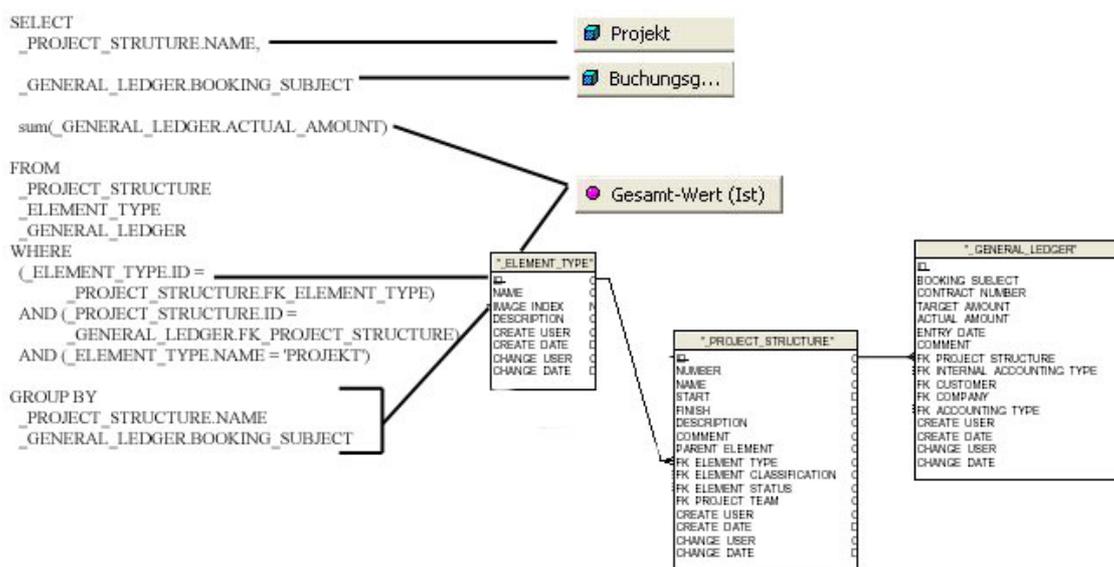


Abb. 3.44: SQL-Generierung der Ad-hoc-Abfrage

Quelle: Vom Verfasser erstelltes Beispiel in Anlehnung an: Business Objects (2004a), S. 27

Diese SQL-Abfrage verdeutlicht noch einmal die Verwendung der Group-By-Bedingung zur Aggregation der Kennzahl-Objekte.

Ein generierter Report kann durch zahlreiche Formatierungen angepasst und durch die oben erwähnten Aggregationsfunktionen innerhalb der Slice & Dice-Analyse ergänzt werden. Die fertiggestellten Reports sind durch die oben beschriebenen multi-dimensionalen Analysefunktionen deutlicher interpretierbar. Dies setzt allerdings voraus, dass die verfügbaren Hierarchien durch die Funktion „Hierarchien“ ausgewählt

wurden. Drill-down und Roll-up lässt den Endanwender in einem Report zwischen den Hierarchien hin und her springen, um detailliertere oder gröbere Daten zu erhalten.

In Abb. 3.45 (s. S. 101) sind zunächst alle Softwarekosten nach Software geordnet. Anschließend haben wir einen Drill-down auf die Software „PMM“ durchgeführt. Dies hat zur Folge, dass alle Lizenzen von PMM und die dazugehörigen Kosten angezeigt werden.

Softwarekosten

| Software | Anschaffungskosten | Wartungskosten |
|-------------------------|--------------------|----------------|
| Arbeitsplan | 100.000,00 | 15.000,00 |
| Catia | 34.530,00 | 3.453,00 |
| dCADE | 400.000,00 | 59.000,00 |
| Microsoft Visual Studio | 150.000,00 | 15.000,00 |
| Microsoft Windows | 39.251,00 | 1.950,00 |
| PMM | 84.430,00 | 20.039,00 |
| QualityServiceCenter | Lizenznummer | 2.753,50 |
| | 813.536,00 | 117.195,50 |

Normaler Bericht

Zeigt die nächste Ebene an

Drill-Filter

PMM

Drill-down auf PMM

| Lizenznummer | Anschaffungskosten | Wartungskosten |
|--------------|--------------------|----------------|
| 1616494464 | 5.000,00 | 2.000,00 |
| 4353345 | 3.330,00 | 10.989,00 |
| 64564 | 11.100,00 | 5.550,00 |
| 76231598345 | 65.000,00 | 1.500,00 |
| | 84.430,00 | 20.039,00 |

Abb. 3.45: Ad-hoc-Abfragen – Drill-down

Slice & Dice zerschneidet den Datenwürfel in benutzerdefinierte Ebenen und bricht die Ebenen auf konkrete Kombinationen von Dimensionselementen herunter (s. Abb. 3.46, S. 101). Wie im unten abgebildeten Beispiel zu sehen, haben wir durch Slice & Dice die Spalten neu geordnet. Dies führt bei der erneuten Abfrage zu einer anderen Sichtweise der abgefragten Daten.

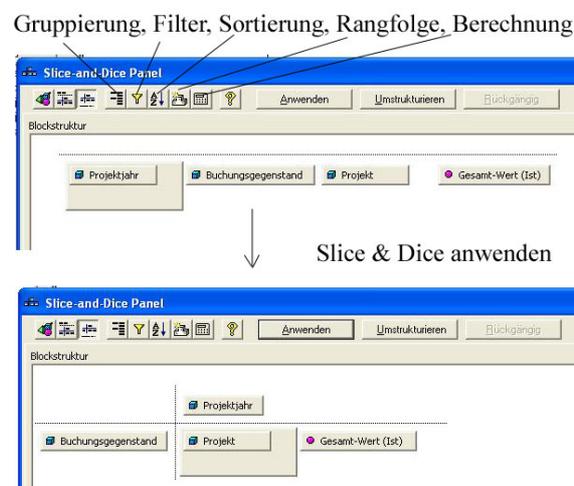


Abb. 3.46: Ad-hoc-Abfragen – Slice & Dice

Umfangreichere Reports können erstellt werden, indem Datensätze aus anderen Datenquellen oder anderen Universen in einen Report eingefügt werden. Dies geschieht entweder durch einen Unterbericht oder durch einfaches Hinzufügen einer neuen Tabelle in den Report, die eine andere Abfrage enthält. Ferner können Diagramme wie in CR.NET zur Darstellung der Zusammenhänge in den Reports verwendet werden. Diese können ebenso durch Drill-down, Roll-up und Slice & Dice analysiert werden. Auf weitere Einzelheiten der Ad-hoc-Abfrage in BO wird aufgrund des Umfangs in der vorliegenden Diplomarbeit verzichtet.

Die erstellten Reports werden im Unternehmens-Repository gespeichert und damit anderen Endanwendern zur Verfügung gestellt. Jeder Endanwender des Unternehmens, der Zugriff zu der Domäne des gespeicherten Universums hat, kann es für seine Analysezwecke verwenden. Hinter dieser Methode verbirgt sich das Anliegen, BI als integrierten, IT-basierten Gesamtansatz für das betrachtete Unternehmen zu sehen und die Informationsverbreitung im gesamten Unternehmen – wie es in Kapitel 2 beschrieben wurde – zu verfolgen. Allerdings basiert der Ansatz der Universen-Repositories nicht auf einem oder mehreren unternehmensweiten DWH und somit hält er sich nicht an die heutigen wissenschaftlichen Erkenntnisse.

Diesbezüglich wird bei dem in dieser Arbeit betrachteten Automobilhersteller derzeit eine BI-Standardisierung und -Strategie durchgeführt, bei der die verwendeten Daten der BI-Insel-Lösungen aus den einzelnen Abteilungen in sogenannten Gravitationszentren⁶⁸ verdichtet gespeichert werden. Anschließend werden diese anwendungsneutral für spezielle Data Marts vorbereitet, die über ein BI-Portal angesteuert werden können. Die Universen aus BO werden dann im Weiteren über das BI-Portal abrufbar sein und werden die Daten aus den Data Marts verwenden. Die BI-Architektur ist vergleichbar mit der in 2.3.2 beschriebenen ODS erweiterten DWH Architektur. Diese BI-Strategie verfolgt den BI-Ansatz gemäß der Definition in 2.2 sowie den unternehmensweiten Zugriff über ein BI-Portal.

Da sich die BI-Infrastruktur zurzeit im Aufbau befindet, wird in der vorliegenden Diplomarbeit nicht näher auf die unternehmensweite Umsetzung eines BI-Ansatzes oder DWH eingegangen und im Folgenden die BI-Abteilungslösung behandelt.

Durch die vorgestellten Standard-Reports und Ad-hoc-Analysen besitzen die Entscheider nun ein Hilfsmittel zur effektiven Entscheidungsunterstützung. Im Vergleich zur Ausgangssituation (s. Abschnitt 3.1.2) benötigen die Abteilungs- und Projektleiter

⁶⁸ Die Verantwortlichen für die BI-Standardisierung und -Strategie bezeichnen eine Kombination aus ODS und DWH als Gravitationszentrum. Im Laufe der Standardisierung soll es zunächst mehrere Gravitationszentren geben, die nach erfolgreicher Umsetzung untereinander verknüpft werden.

nicht mehr verschiedene Systeme, sondern lediglich ein System zur Projekt- und Ressourcenplanung. Darüber hinaus stellt es komfortable BI-Komponenten für die Entscheidungsunterstützung bereit. In Abschnitt 3.1.2 verdeutlichte das Ablaufdiagramm (s. Abb. 3.1, S. 36) die enorme Prozessabfolge zur Erstellung eines einfachen Standard-Reports. Durch die Entwicklung von PMM und vor allem durch die Integration der BI-Komponenten wird diese Prozessabfolge nun minimiert und zusätzlich können sowohl zahlreiche anwendungsfallorientierte Standard-Reports als auch beliebige Ad-hoc-Reports bzw. -Analyse durchgeführt werden. Diese optimierte Prozessabfolge veranschaulicht Abb. 3.47 (s. S. 103).

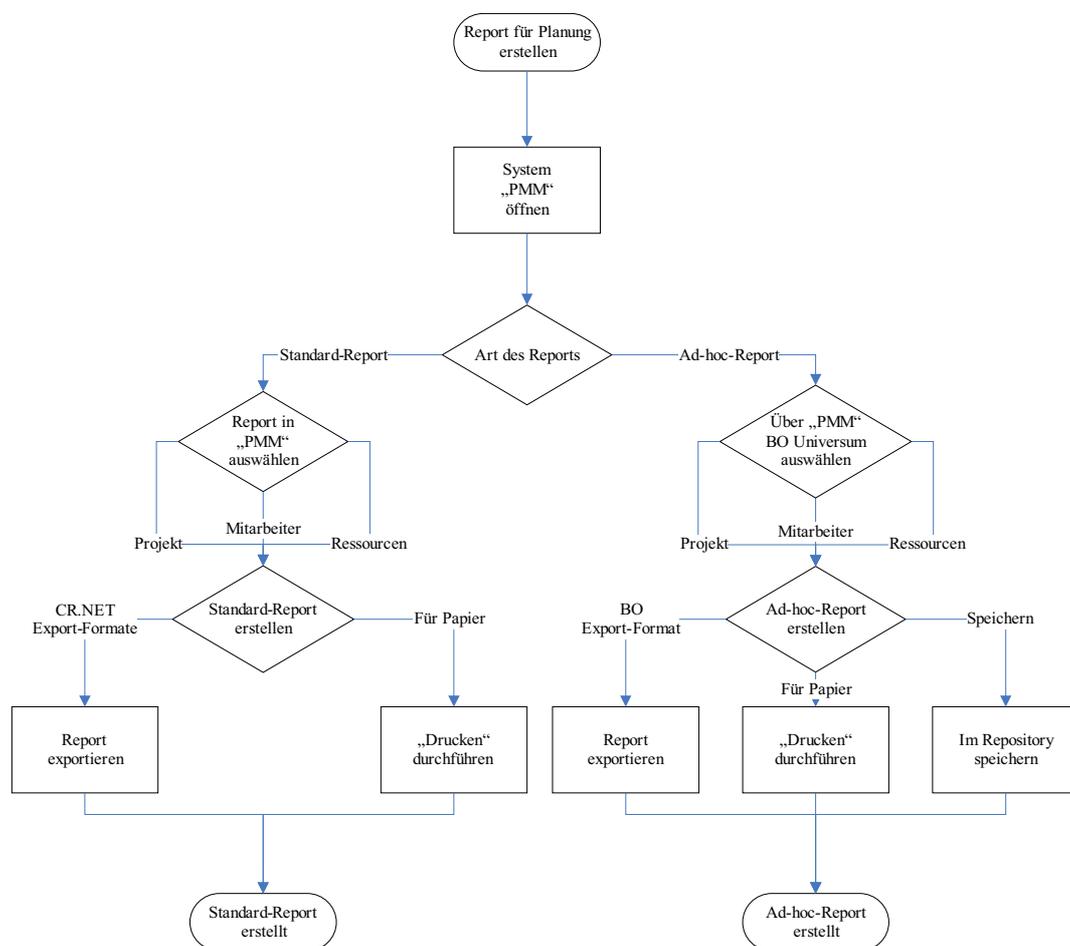


Abb. 3.47: Ablaufdiagramm – Reporterstellung mit BI-Komponenten

Aus dieser Abbildung wird ersichtlich, dass nur noch ein System zur Reporterstellung unabhängig seines Anwendungszwecks aufgerufen werden muss. Zudem kann nachvollzogen werden, dass verschiedene Möglichkeiten bestehen, Reports für die effektive Entscheidungsunterstützung zur Projekt- und Ressourcenplanung zu erstellen.

3.4 Resümee

Vorabgehend wurde erläutert, dass durch die Entwicklung von PMM die Geschäftsprozesse bezüglich der Projekt- und Ressourcenplanung der betrachteten IT-Abteilung effizienter gestaltet wurden. Denn sowohl die mühsame Aufbereitung von Excel-Tabellen zur Aufstellung des Bebauungsplans und der Forecast-Planung zur Lizenzverwaltung als auch für die Finanzplanung der Software entfällt. Weiterhin wird durch die Verwendung von PMM nur ein System zur Finanz- und Kapazitätenplanung der Projekte und Personalkosten eingesetzt. Dadurch kann PMM sowohl zur Projektplanung als auch zur Ressourcenplanung verwendet werden.

Darüber hinaus haben wir die Entscheidungsunterstützung erheblich durch die zusätzliche Integration von BI-Komponenten in PMM zur Projekt- und Ressourcenplanung verbessert. Die Integration der BI-Komponenten erreicht einen Mehrwert, da nun nicht nur statische Berichte abrufbar sind, sondern dynamische Standard-Reports sowie Ad-hoc-Abfragen und -Analysen durchführbar sind. Dies haben wir im Anwendungsbeispiel umgesetzt, indem die Daten im Daten-Layer mit Hilfe von CR.NET durch Manipulationen und Formatierungen der Datensätze sowie durch Universen im BO Designer aufbereitet wurden. Anschließend können im Präsentations- bzw. Logik-Layer von PMM Informationen aus den operativen Daten des Daten-Layers durch Standard-Reports sowie Ad-hoc-Abfragen und -Analysen generiert werden. Die Standard-Reports sind im Front-End integriert und werden nach Auswahl eines Reports in PMM übersichtlich dargestellt. Ad-hoc-Abfragen und -Analysen kann ein Abteilungs- bzw. Projektleiter durch BO Webintelligence oder Desktop Intelligence anwenderfreundlich durchführen. Für verschiedene Anwendungsfälle, wie Projektmanagement, Mitarbeiterkapazitätenplanung oder Ressourcenplanung stehen ihm drei Universen zur Auswahl bereit. Diese Verwendung von BI führt zu einer qualitativen Verbesserung der Entscheidungsunterstützung während der Projekt- und Ressourcenplanung.

Allerdings muss mit Funktionseinschränkungen gerechnet werden, da zum einen größtenteils keine Hierarchien vorhanden sind und dadurch keine multidimensionale Analyse möglich ist. Zum anderen liegen die Daten teilweise nicht im gewünschten Format vor.

Dennoch stehen den Abteilungs- und Projektleitern durch PMM sowie durch BI eine Vielzahl von Informationen aus den verfügbaren Daten zur Verfügung. Diese können die Abteilungs- und Projektleiter zur Entscheidungsunterstützung für die Planung, Leitung und Kontrolle sowohl der Kosten als auch der Kapazitäten ihrer Projekte und Ressourcen nutzen. Daraus ergibt sich in Bezug auf die Ausgangsfrage, Optimierung der Projekt- und Ressourcenplanung bezüglich der Entscheidungsunterstützung, eine

Reduzierung des Zeitaufwandes und der Kosten sowie die Steigerung der Planungsqualität während des operativen Betriebs. Insgesamt kann daher festgehalten werden, dass der Einsatz von BI innerhalb der Projekt- und Ressourcenplanung die bisherigen Geschäftsprozesse der Digitalen Fabrik- und Fertigungsplanung erheblich effizienter gestaltet.

4 Optimierung der Entscheidungsunterstützung

Die vorangegangenen Kapitel dienen der Einführung in den Themenkomplex Business Intelligence und den damit verbundenen Ansätzen und Systemen sowie zur Vorstellung eines Beispiels aus der Praxis. Anhand des Anwendungsbeispiels wird sichtbar, dass Theorie und Praxis häufig verschiedene Wege gehen. Dennoch wird der Gedanke, BI als integrierten, unternehmensspezifischen, IT-basierten Gesamtansatz zu verstehen, im Auge behalten. Denn die erstellten Universen werden im Unternehmensrepository gespeichert und zusätzlich wird gegenwärtig eine BI-Strategie zur Standardisierung und Aufbau einer unternehmensweiten BI-Architektur verwirklicht. Nach erfolgreicher Umsetzung wird diese Strategie den Entscheidungsträgern oder Entscheidern, die Zugriff zum Unternehmens-BI-Portal erhalten, die Vorteile von BI bereitstellen. Zurzeit ist dies jedoch noch nicht umgesetzt und daher entstehen BI-Insel-Lösungen, um den Projekt- und Abteilungsleiter ebenso die Vorteile von BI zur Verfügung zu stellen.

Innerhalb des vierten Kapitels werden die Schwachstellen des praktischen Teils der BI-Einführung aufgezeigt und anschließend Gründe für die Einführung eines Data Warehouses genannt. Abschließend wird ein Data Warehouse Konzept entwickelt, das auf die Anforderungen und Voraussetzungen des praktischen Beispiels zugeschnitten ist. Diese Vorgehensweise ist damit zu begründen, dass bereits im praktischen Teil (s. Abschnitt 3.3.2) Funktionseinschränkungen festzustellen waren. Eine grundlegende Analyse der Schwachstellen und Überlegungen zur Problemlösung kann die Projekt- und Ressourcenplanung zusätzlich effektiver gestalten.

4.1 Potenzialanalyse

Im Allgemeinen dient die Potenzialanalyse der Analyse eines Unternehmens mit dem Ziel, die Stärken und Schwächen sowie die verfügbaren Ressourcen aufzuzeigen. Außerdem sollen sogenannte Potenzialfelder erschlossen werden, deren Optimierung die größten Verbesserungsmöglichkeiten bietet. Dabei dient das Aufzeigen der Schwachpunkte und der Handlungsspielräume in den Potenzialfeldern sowie einer Abschätzung dieser, der Leistungsverbesserung und Kostensenkung (vgl. Feder Consulting (2006)). In diesem Fall sollen mit Hilfe der Potenzialanalyse Schwachstellen und Verbesserungsmöglichkeiten der BI-Integration zur weiteren Verwendung von PMM in der Digitalen Fabrik- und Fertigungsplanung aufgezeigt werden.

Durch die Neuentwicklung und Einführung von PMM ist bereits jetzt ein Mehrwert im Ablauf der Geschäftsprozesse zu verzeichnen. Die mühselige Aufbereitung und Planung mittels Excel-Tabellen oder der verschiedenen Datenbanken und Systeme sind komplett entfallen.

Zusätzlich bietet PMM aufgrund des Aufbaus der Projektstruktur für das Projektmanagement und der Organisationsstruktur für die Ressourcenplanung eine detaillierte Planung und Leitung der täglichen Abläufe. Darüber hinaus verbessern die integrierten BI-Komponenten die Reporting- und Analysemöglichkeiten der anfallenden Kosten, verfügbaren Kapazitäten und Ressourcen. Dadurch wird eine verbesserte Entscheidungsunterstützung zur Projekt- und Ressourcenplanung hinsichtlich Finanzen und Kapazitäten erzielt. Jedoch existieren Potenzialfelder bezüglich der BI-Komponenten, dessen Optimierung eine Verbesserung der Informationsgenerierung für Reporting- und Analysefunktionalitäten gewährleisten kann. Mit Potenzialfeldern ist die Datenbereitstellung für die Standard-Reports und die Ad-hoc-Analyse in PMM gemeint, worauf im kommenden Abschnitt detaillierter eingegangen wird.

4.1.1 Schwachstellen der Datenbereitstellung für Standard-Reports

In Kapitel 3.3.1 wurde ausführlich beschrieben, wie die operativen Daten mit Hilfe von CR.NET aufbereitet wurden. Eine Middleware Komponente des Reporting-Systems setzt direkt auf die operative Datenbank auf und stellt die gewünschten Tabellen und Datensätze zur Auswahl bereit. Es waren für jeden Report zahlreiche Formelfelder zur Umwandlung und Manipulation der Formate, der Datensätze sowie der Datentypen nötig.

Zur Verwendung der gewünschten Datensätze in einem Report treten durch die Beziehungen zwischen unterschiedlichen Tabellen übermäßig viele Joins auf. Dies ist darauf zurückzuführen, dass durch die Datenstruktur der operativen Datenbank teilweise zahlreiche Tabellen miteinander in Beziehung stehen. Diese übermäßige Verwendung von Joins ruft erhebliche Mängel der Performance hervor und führt zu beträchtlichen Wartezeiten während der Reportgenerierung. Dies liegt darin begründet, dass die eigentliche Selektion der Datensätze auf den Client-Rechnern stattfindet und auf die Prozessorleistung des Servers verzichtet werden muss (vgl. Schinzer/Bange (1999), S. 51). Zusätzlich werden die Datensätze nach Erhalt der SQL-Abfrage zur Laufzeit, d. h. während der Report-Abfrage, an die gewünschten Formatierungen angepasst und alle Berechnungen für den jeweiligen Report durchgeführt. Dies hat ebenso große Einflüsse auf die Performance und die Wartezeit.

Durch diese Ressourcenbelastung entsteht während der Reportgenerierung eine zu hohe Zeitspanne bis der gewünschte Report fertig gedruckt auf dem Bildschirm angezeigt wird. In Zukunft können sich die Wartezeiten weiter erhöhen, da durch den operativen Betrieb ein starker Datenzuwachs stattfinden wird.

Lägen die Tabellen bereits in der gewünschten Struktur (ohne übermäßiges Auftreten von Joins), die Datensätze und Datentypen im gewünschten Format sowie die Berechnungen in den Tabellen vor, kann dies eine positive Auswirkung auf die Performance und eine Verkürzung der Reporterstellungszeit bewirken.

4.1.2 Schwachstellen der Datenbereitstellung für Ad-hoc-Analyse

Eine weitere Schwachstelle stellt die Datenbereitstellung in BO dar. Das Problem ist hierbei nicht das Prinzip der Universen, sondern der Zugriff der Universen auf eine transaktionsorientierte Datenbank und somit die virtuelle Data Warehouse Architektur. Dies belegen auch die Ausführungen von SCHINZER und BANGE belegen. Die Vorteile dieser Architekturvariante liegen in der kostengünstigen Anschaffung und Einführung. Außerdem müssen mit Ausnahme des Middleware-Werkzeugs keine DWH-Datenbanken erworben werden. Es kann zudem auf eine aufwendige Entwicklung einer DWH Infrastruktur verzichtet werden (s. Abschnitt 2.3.2).

Das wirkt sich jedoch nachteilig auf die Performance aus, da die Auswahl der Tabellen, Datensätze und ihrer Verknüpfung durch Joins auf den Client-Rechnern stattfindet. Weiterhin wird bei komplexen Abfragen die operative Datenbank enorm beeinträchtigt, da große Datenmengen anfallen und mehrere Benutzer gleichzeitig auf die Daten der operativen Datenbank zugreifen. Zusätzliche Nachteile entstehen, da keine aggregierten Werte im Vorhinein auf dem Server gespeichert werden können, sondern diese bei jeder Benutzung zur Laufzeit neu berechnet werden müssen. Dadurch wird die Leistungsfähigkeit der Client-Rechner gehemmt, was Wartezeiten zur Folge hat (vgl. Schinzer/Bange (1999), S. 50 f.).

Ein weiteres Problem bei der Verwendung von BusinessObjects stellt die Tatsache dar, dass die Datensätze in den Tabellen nicht im gewünschten Format vorhanden sind und BusinessObjects keine vielfältigen Manipulationen der ausgewählten Daten zulässt. Es müssten komplexe ETL-Prozesse zur Transformation der Daten angewendet werden, um die gewünschten Ergebnisse zu erhalten oder es muss auf Funktionalitäten verzichtet werden (wie in 3.3.1 beschrieben unter anderem auf die Ad-hoc-Abfrage der Zeiterfassung). Ferner war es mit BusinessObjects nicht möglich, managementrelevante Hierarchien wie z. B. die Projektstruktur, Organisationsstruktur oder eine Zeitdimension

abzuleiten. Dies ist darauf zurückzuführen, dass wir weder die Hierarchien aus einer rekursiven Tabelle auslesen konnten, noch dass die Datenbankstruktur es zulässt, andere relevante Hierarchien abzubilden. Eine Ausnahme stellen die Buchungszeitraum-Hierarchie, die Software-Hierarchie und die Standort-Hierarchie der Organisationsstruktur dar. Echte multidimensionale Hierarchien könnten nur durch den Einsatz von komplexen ETL-Prozessen und der Erstellung angepasster Tabellen implementiert werden. Aufgrund fehlender ETL-Software wurde vorerst darauf verzichtet.

Angesichts der fehlenden Hierarchien sind multidimensionale Operationen wie Drill-down nur teilweise anwendbar. Somit ist es nicht möglich, umfangreiche multidimensionale Analysen zu führen, wodurch der Entscheider in seinen Analysemöglichkeiten eingeschränkt wird.

Diese Problematik zeigt, dass es notwendig ist, durch ETL-Prozesse neue, den entscheidungsunterstützenden Anforderungen, angepasste Tabellen zu erstellen.

Ebenso wie bei der Datenbereitstellung durch CR.NET kann eine Verbesserung der Performance durch eine optimierte Datenbankstruktur und korrekte Datenformate erreicht werden. Ferner entfallen die Einschränkungen bei der Ad-hoc-Abfrage, wenn die Daten im gewünschten Format vorliegen.

Zusammenfassend werden die Schwachstellen der Datenbereitstellung im Folgenden aufgelistet:

- Performanceverschlechterung durch Selektion der Datensätze und anschließende Verknüpfung der Tabellen durch Joins mittels der Client-Rechner
- Performanceverschlechterung durch übermäßig viele Joins
- Performanceverschlechterung durch Manipulationen und Berechnungen der Datensätze während der Laufzeit
- Performanceverschlechterung durch Abfragen erhöhter Datenmengen auf einer operativen Datenbank
- Performanceverschlechterung durch das Fehlen aggregierter Werte auf der Datenbank
- Funktionseinschränkung der Ad-hoc-Abfrage und Analyse durch inkompatible Tabellen oder Datenformate sowie durch die operative Datenbankstruktur

- Funktionseinschränkung durch fehlende multidimensionale Hierarchien, daher ist keine multidimensionale Analyse entsprechend der Definition möglich
- Funktionseinschränkungen durch fehlende Historisierung
- Beeinflussung des operativen Betriebs durch Ressourcenbelastung

Durch diese Schwachstellen findet keine optimale Entscheidungsunterstützung statt. Dies verlangt nach einer Optimierung der Datenqualität, um die starke Beeinflussung der Entscheidungsunterstützung und der Performance zu verhindern.

4.1.3 Optimierung

Aus den Ausführungen der vorangegangenen Abschnitte wird ersichtlich, dass das Problem bei der Verwendung einer operativen Datenbank als Grundlage für Reporting- und Analysefunktionalitäten liegt. In operativen Datenbanken werden die Daten während der Benutzung der Datenbank abgefragt und verändert. Diese Eigenschaft wird durch ein Transaktionskonzept kontrolliert, damit keine Fehler auftreten (vgl. Messerschmidt/Schweinsberg (2003), S. 28 f.). Weiterhin sind die Transaktionen darauf ausgerichtet, die Daten aus dem operativen Betrieb zu einem bestimmten Zeitpunkt ideal zu verarbeiten. Daher ist eine operative Datenbank transaktionsorientiert und performanceoptimiert in Bezug auf die zeitpunktgenaue Verarbeitung anfallender, geschäftsbezogener Daten aus dem Wertschöpfungsprozess (vgl. Kemper/Finger(1999), S. 78). Durch diese transaktionsorientierte Ausrichtung sind die Tabellen oft in einer sehr komplexen, normalisierten Struktur angeordnet, um schnelle Transaktionen im täglichen Geschäft zu gewährleisten. Dadurch kommt es bei komplexen betriebswirtschaftlichen Abfragen eines MSS zur Verwendung von übermäßig vielen Joins zwischen den Tabellen, da die gewünschten Datensätze in zahlreichen verschiedenen Tabellen vorliegen. Dies führt zu Wartezeiten bei der Abfrage. Weiterhin können durch die normalisierte Datenbankstruktur meist keine managementrelevanten Hierarchien abgebildet werden.

In einem DWH hingegen liegen multidimensionale Datenbankstrukturen zugrunde, die betriebswirtschaftlichen Fragestellungen nachkommen. Die Datensätze liegen bereits hinsichtlich ihrer Zusammengehörigkeit gruppiert in einer Dimensionstabelle vor. Somit enthalten Abfragen weniger komplexe Joins und es können Hierarchien abgebildet werden, die für betriebswirtschaftliche Fragestellungen unentbehrlich sind.

Darüber hinaus ist eine operative Datenbank funktionsorientiert und nicht wie eine dispositive Datenbank zur Entscheidungsunterstützung themenorientiert aufgebaut, wodurch es nötig ist die Daten intern umzustrukturieren. Dies bezieht sich auch auf die Formate und Datentypen, die für eine schnelle Abwicklung der Geschäftsprozesse optimiert sind und nicht für Berechnungen oder Darstellungen zur Entscheidungsunterstützung. Dies geschieht wiederum auf Kosten der Performance oder von Funktionseinschränkungen. Ferner bezieht sich die oben erwähnte Transaktionsorientierung und Performanceoptimierung vor allem auf geringe Datenmengen pro Transaktion (vgl. Hothuis (1999), S. 41). Bei betriebswirtschaftlichen Fragestellungen hingegen kommt es zu komplexen Abfragen und Berechnungen großer Datenmengen, wodurch es bei Abfragen zu einer starken Ressourcenbelastung auf einer operativen Datenbank führt. Dies beeinträchtigt nicht nur die Entscheidungsunterstützung, sondern auch das operative Geschäft (vgl. Kemper/Finger (1999), S. 78). Die Ressourcenbelastung steigt exponentiell zur Komplexität der Abfragen auf einer operativen Datenbank. Hingegen verändert sich die Ressourcenbelastung in Bezug auf die Komplexität bei Abfragen auf eine dispositive Datenbank kaum (s. , S.).

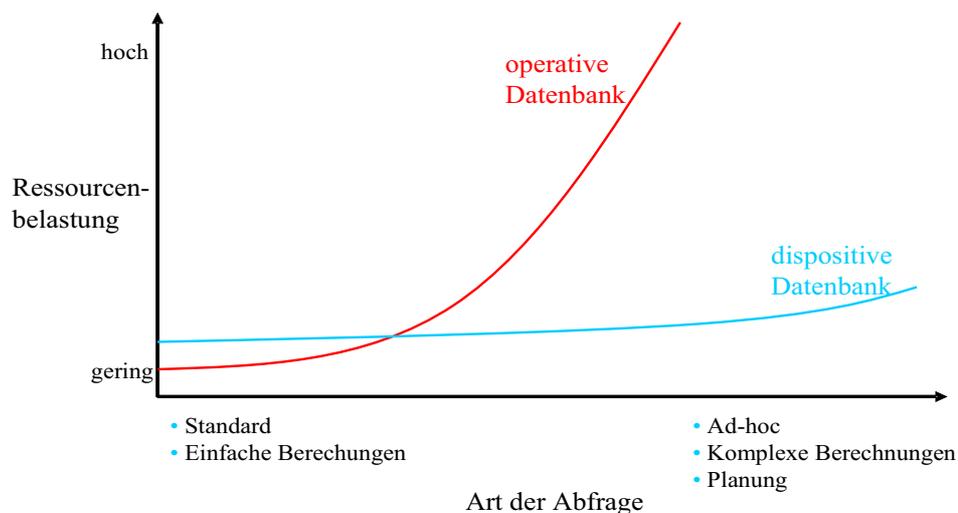


Abb. 4.1: Ressourcenbelastung bei Abfragen auf operative und dispositive Datenbestände

Quelle: Darstellung des Verfassers in Anlehnung an: Ortiz (2005), S. 20

Für betriebswirtschaftliche Fragestellungen ist weiterhin eine historische Betrachtung der Datenbestände unerlässlich, um Verdichtungen der aktuellen und historischen Daten bereitzustellen (vgl. Kemper/Finger (1999), S. 79). Durch PMM ist zwar eine jahresbezogene Betrachtung der Projekte und Mitarbeiterkapazitäten möglich, jedoch spiegelt dies den Ist-Zustand der aktuellen Planung wieder und kann täglich geändert werden. Zudem liefern Abfragen auf einer operativen Datenbank aufgrund der sich ständig ändernden Daten keine reproduzierbaren Ergebnisse (vgl. Bauer/Winterkamp (1996), S. 50). Eine historische Speicherung in einem DWH lässt eine echte Historie der aufbe-

reiteten Daten zu. Dies ermöglicht Trendanalysen für zukünftige Projekte, was, wie in Abschnitt 3.3.2 erwähnt, zurzeit nicht möglich ist.

Abschließend soll noch einmal betont werden, dass die vorangegangenen Erkenntnisse verdeutlichen, dass mit Performance- und Funktionseinschränkungen gerechnet werden muss, wenn für Reporting- und Analyse-Systeme eine operative Datenbank zugrunde liegt. Wie bereits in Abschnitt 2.3.1 erwähnt und ab Abschnitt 4.1 gezeigt sind operative Daten nicht für Analysezwecke ausgerichtet, sondern für die Unterstützung der Geschäftsprozesse. Dadurch ist es zu empfehlen, eine dispositive Datenhaltung in Form eines DWH aufzubauen, um den gewünschten Ansprüchen in Bezug auf die Funktionalität und Performance gerecht zu werden. In der nachfolgenden Tab. 4.1 (s. S. 111) sind die Unterschiede zwischen einer operativen und einer dispositiven Datenbank noch einmal übersichtlich dargestellt.

Tab. 4.1: *Operative vs. dispositive Datenbanken*

| | <i>Operative Datenbank</i> | <i>Dispositive Datenbank</i> |
|---------------------------------------|--|--|
| Ziel | Abwicklung der Geschäftsprozesse | Informationen für die Entscheidungsunterstützung |
| Inhalt der Daten | anwendungsbezogen, funktionsorientiert | themenbezogen (nach Produkten, Märkten, ...) |
| Art der Daten | transaktionsorientiert, detaillierte Geschäftsvorfalldaten | aggregierte, transformierte Daten |
| Datenstruktur | flache, nicht hierarchische Tabellen (meist 3NF) | multidimensionale Strukturen (denormalisiert) |
| Datenintegration | kaum aus anderen Anwendungen integriert | Möglichkeit der Datenintegration aus einer Menge von Anwendungen |
| Zeitraumbezug | aktuell, zeitnah | gegenwärtig und zukünftig, Historiebetrachtung |
| Datenvolumen pro Transaktion/ Abfrage | klein | sehr umfangreich |

Mohr (2005), S. 20; Holthuis (1999), S. 41; Kemper et al. (2004), S. 14
 Quelle: Zusammenfassung des Verfassers in Anlehnung an:

Aus der Tabelle geht hervor, dass eine dispositive Datenbank (ein DWH) exakt auf die Ansprüche einer entscheidungsunterstützenden Abfrage und Analyse der Datenbestände ausgerichtet ist, da die benötigten multidimensionalen Datenbankstrukturen und die gewünschten Datenformate und Datentypen in einem von den operativen Datenbanken losgelösten DWH bereitgestellt werden. Die multidimensionalen Strukturen verhindern ebenso die Verwendung von übermäßig vielen komplexen Joins, da die Tabellen durch

diese Struktur auf eine gewisse Anzahl von Joins beschränkt werden. Dies ist darauf zurückzuführen, dass die Daten bereits zusammengefasst in Dimensionstabellen vorliegen. Durch die optimierte Struktur sowie die einfachen Datenformate wird das Antwortverhalten extrem verbessert und die historisch gespeicherten Daten lassen erst eine langfristige Erkennung von Trends mittels Zeitreihenanalyse zu (vgl. Kurz (1999), S. 75). Zusätzlich können die verlangten multidimensionalen Analysen mit Hilfe der Hierarchien durchgeführt werden. Des Weiteren wird das tägliche Geschäft nicht durch die Ressourcenbelastung der betriebswirtschaftlichen Abfragen beeinträchtigt. Zusätzlich können Daten aus verschiedenen Datenquellen mit einbezogen werden, um den Entscheidern weitere Informationen bereitzustellen. Im Allgemeinen verbessern DWH basierte Planungssysteme den Planungsprozess (vgl. Bauer/Günzel (2001), S. 22) und verbessern die Entscheidungsgrundlage (vgl. Mucksch/Behme (2000), S. 70). Dies führt zu einem branchenspezifischen Nutzenpotenzial, wodurch DWH Lösungen überwiegend für die Erreichung der strategischen Ziele und zur Steigerung der Wettbewerbsfähigkeit des Unternehmens eingesetzt werden (vgl. Martin (1999), S. 19). Diese Faktoren rechtfertigen die umfangreiche Entwicklung eines Data Warehouses. Daher wird in den folgenden Abschnitten ein Data Warehouse Konzept als Realisierungsansatz vorgestellt. Dieser Ansatz stellt, in Bezug auf die Ausgangsfrage der vorliegenden Diplomarbeit, einen optimalen Nutzen der Entscheidungsunterstützung bezüglich Zeit, Kosten und Qualität dar.

4.2 Entwicklung eines Data Warehouse Konzeptes

In den weiteren Abschnitten wird in Anlehnung an das Rahmenkonzept zum Aufbau eines DWH aus Abschnitt 2.3.2 ein DWH Konzept entworfen. Bis zum Aufbau eines DWH müssen die Phasen Datenselektion, Datenmodellierung, Datenextraktion, Datentransformation und Datenimport durchgeführt werden. Zum Erhalt eines DWH müssen kontinuierlich der Betrieb und die Pflege gewährleistet werden.

Im Anschluss an die erste Phase, Datenselektion, wird in der Phase, Datenmodellierung, aufgezeigt, welche Datenbankstrukturen einem DWH in Bezug auf eine Projekt- und Ressourcenplanung zugrunde liegen müssen, um keine Einschränkungen während der Ad-hoc-Abfrage und -Analyse hinnehmen zu müssen. Weiterhin wird im Anschluss erläutert, welche DWH Architekturvariante für eine Projekt- und Ressourcenplanung mit BO und PMM als MSS aufgebaut werden sollte. Dieses DWH Konzept soll als Optimierungsansatz zur effizienten Steuerung, Kontrolle und Analyse der Projekt- und Ressourcenplanung dienen.

4.2.1 Datenselektion

Während der Datenselektion werden alle Daten, die für den Endanwender nötig sind, um eine effektive Entscheidungsunterstützung zu gewährleisten, aus den operativen Datenbanken in einem Fachkonzept abgebildet. Am Ende der Datenselektion steht ein semantisches Datenmodell, das die Grundlage für die Entwicklung der logischen DWH Struktur bietet. Zur Modellierung wird das mERM verwendet, da es die Erfordernisse auf einfache und übersichtliche Weise darstellt.

Im Falle der Optimierung der Projekt- und Ressourcenplanung in der betrachteten Abteilung wurde bereits in den vorangegangenen Abschnitten (s. Abschnitt 3.2.1 und 3.2.2) aufgezeigt, welche Daten wofür verwendet und benötigt werden. In der derzeitigen Lage werden die Daten nur aus einer Datenbank zusammengefasst. Für das semantische Datenmodell werden die in der PMM-Datenbank vorliegenden Daten berücksichtigt, jedoch Vorschläge zur Optimierung der Datenqualität gegeben. Daraus leiten wir folgende drei Faktrelationen und sechs Dimensionen ab (s. Abb. 4.2, S. 113):

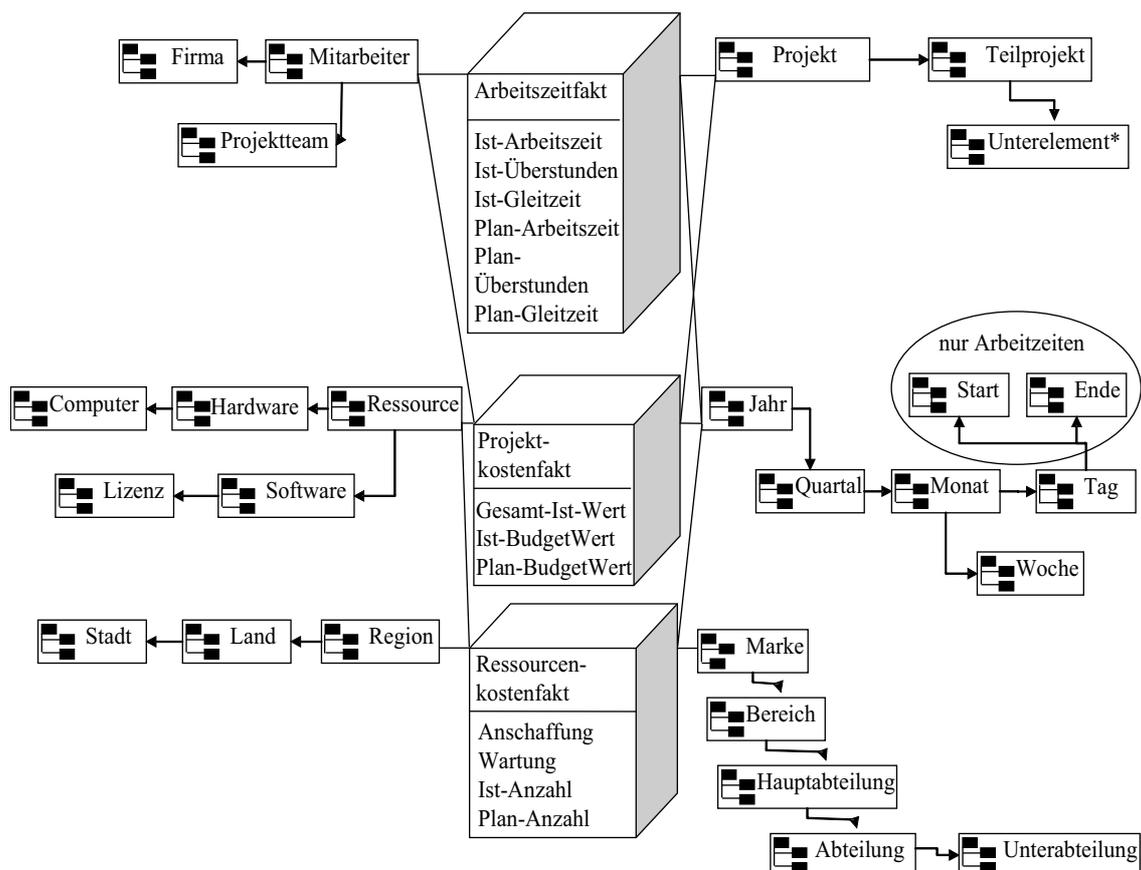


Abb. 4.2: Semantisches Data Warehouse Datenmodell

- *Arbeitszeitfakt*: Diese Faktrelation soll alle Zeiten, die in Bezug auf die Arbeitszeit gespeichert werden enthalten (s. Abb. 4.2, S. 113). Diese Zeiten werden als Plan- und Ist-Werte abgelegt, so dass ein Ist-Plan-Vergleich über die Arbeit- und Projektzeiten möglich ist. Die Faktrelation *Arbeitszeitfakt* steht mit den Dimensionen *Mitarbeiter*, *Projekt* und *Zeit* in Beziehung. Wobei *Zeit* die historische Zeitdimension des DWH widerspiegelt. Die Zeitdimension sollte für die Faktrelation *Arbeitsfakt* bis auf die Start- und Endzeit der Tage herunter gebrochen werden können, wenn es sich um die Tagesarbeitszeit handelt. Dadurch ist eine detaillierte Auswertung der Tagesarbeitszeit für die Zeiterfassung möglich. Die Hierarchieebene *Unterelement* der Dimension *Projekt* kann in Bezug auf PMM entweder eine Aktivität, eine DLV oder ein Prozess sein.
- *Projektkostenfakt*: Ziel dieser Faktrelation (s. Abb. 4.2, S. 113) ist eine Übersicht über die entstanden Kosten und erlangten Einnahmen bei vergangenen, laufenden und zukünftigen Projekten. Dazu werden, in Anlehnung an die Ausführungen zu PMM, der Gesamtkosten-Ist-Wert und die Ist- und Plan-Werte der Budgets gespeichert. Im Laufe der Zeit wird anhand der gespeicherten Daten eine Trendanalyse möglich sein, da aufgrund des DWH eine historische Speicherung stattfindet. Weiterhin kann anhand der Dimensionen *Projekt*, *Mitarbeiter* und *Ressourcen* nachvollzogen werden, welche Ursache ein bestimmter Wert hat.
- *Ressourcenkostenfakt*: Diese Faktrelation (s. Abb. 4.2, S. 113) soll die Ressourcenplanung unterstützen und eine Vielzahl von multidimensionalen Analysemöglichkeiten erlauben. Dazu werden die Anschaffungskosten, Wartungskosten sowie die aktuelle und geplante Anzahl der Ressourcen abgespeichert und aufbereitet. Die Faktrelation *Ressourcenkostenfakt* steht mit den Dimensionen *Ressource*, *Zeit*, *Ort* und *Organisationsstruktur* in Beziehung. Dadurch können Abfragen realisiert werden, die die Kosten für eine bestimmte Ressource anhand von bestimmten Standorten oder Unternehmensbereichen anzeigen.

Anhand dieser drei Faktrelationen und ihrer Dimensionen sind alle nötigen Informationen zur effektiven Analyse und Entscheidungsunterstützung einer Projekt- und Ressourcenplanung vorhanden.

Anschließend wird nun konkreter auf die verwendeten Daten eingegangen, indem dieses semantische Datenmodell zu einem logischen Data Warehouse Modell verfeinert wird.

4.2.2 Datenmodellierung

Anhand des semantischen Datenmodells wird in dieser Phase ein logisches Datenmodell entworfen, das die Grundlage für die spätere Implementierung bietet. Da, wie in Abschnitt 2.3.4 erwähnt, prinzipiell davon ausgegangen wird, dass dem Data Warehouse eine relationale Datenbank zugrunde liegt, werden die Verwandten des Star Schemas zur Datenmodellierung herangezogen. Dabei ist jedoch abzuwägen, welche Art des Star Schemas im vorliegenden Fall Verwendung findet.

Das *klassische Star Schema* ist ein einfaches, intuitives Datenmodell mit einer geringen Anzahl von physischen Data-Warehouse-Tabellen. Weiterhin werden bei Abfragen kaum Join-Operationen ausgeführt, welches positive Einflüsse auf die Performance hat. Jedoch existieren Nachteile wie Redundanz innerhalb der Dimensionstabellen und starke Performanceprobleme durch das schlechte Antwortverhalten bei sehr großen Dimensionstabellen (vgl. Kurz (1999), S. 159; Hahne (1999), S. 157 f.). Darüber hinaus spricht HAHNE ein weiteres Problem an, welches, ergänzend zu den oben erwähnten, das klassische Star Schema nur für kleinere und mittelgroße Datenmengen empfehlenswert macht. Das Problem entsteht demnach durch die level-Attribut-Umsetzung, da diese nicht nur bei jeder Abfrage an das Data Warehouse mitgeliefert werden müssen, sondern bei Änderungen in der Dimensionshierarchie enormen Aufwand sowohl am Data Warehouse Modell als auch an den Abfragen verursachen (vgl. Hahne (1999), S. 158). Dieses Problem kann durch den Modellierungsansatz nach dem Fact Constellation Schema behoben werden.

Bei dem *Fact Constellation* Schema wird jedes Aggregat in einer eigenen Fakttablelle gespeichert wird. Dadurch wird eine Performanceverbesserung bei Abfragen erzielt, weil die Fakttabellen kleiner sind (vgl. Bauer/Günzel (2001), S. 205). Dem wirkt entgegen, dass bei einer Abfrage, in der gleichzeitig Aggregate und einzelne Faktdaten auftreten, die Komplexität erhöht wird und durch die starke Zunahme an Fakttabellen bei jeder weiteren Dimension das Schema komplexer wird. Insgesamt ist das Fact Constellation Schema dem klassischen Star Schema nur vorzuziehen, wenn diese sehr große Fakttabellen beinhaltet (vgl. Hahne (1999), S. 160).

Eine weitere Alternative, um auf die level-Attribute des klassischen Star Schemas verzichten zu können, stellt das Snowflake Schema dar. Das *Snowflake Schema* bietet die Vorteile des Fact Constellation Schemas, zudem hat es teilweise deutlich verkürzte Zugriffszeiten (vgl. Holthuis (1999), S. 203). Außerdem existieren keine redundanten Einträge in den Dimensionstabellen (vgl. Kurz (1999), S. 166). Jedoch steigt aufgrund der großen Anzahl an Tabellen die Komplexität. Entweder müssen bei Abfragen eine erhöhte Anzahl von Join-Operationen oder komplexe SQL-Befehle ausgeführt werden.

Dies erhöht die Fehleranfälligkeit und lässt bei der Wartung oder Änderung von Hierarchien den Aufwand extrem steigen (vgl. Kurz (1999), S. 166; Kemper et al. (2004), S. 64).

In Anlehnung an BAUER und GÜNZEL ist das klassische Star Schema als Ansatz zur multidimensionalen logischen Modellierung den anderen Varianten auch im vorliegenden Fall vorzuziehen. Demzufolge werden Abfragen meist auf höheren Hierarchiestufen definiert, welches durch die Struktur des klassischen Star Schemas zu schnelleren Antwortzeiten führt. Weiterhin ist das Datenvolumen der Hierarchieebenen im Gegensatz zur Größe der Fakttablelle meist geringer, wodurch die denormalisierten Dimensionstabellen zu keiner großen Datenvolumenerhöhung führen. Des Weiteren treten Änderungen an den Dimensionstabellen seltener auf als an den Fakttabellen und verringern somit das Auftreten von Fehlern.

Die level-Problematik kann durch Abfragen der Aggregate zur Laufzeit vernachlässigt werden (vgl. Bauer/Günzel (2001), S. 203). Jedoch ist zu bemerken, dass im vorliegenden Anwendungsfall level-Attribute verwendet werden, da teilweise Aggregate als vorberechnete Werte im DWH gespeichert werden sollen, um die Antwortzeiten zu verringern.

Abschließend lässt sich festhalten, dass das klassische Star Schema Vorteile bietet wie die einfache Struktur, die einfache, flexible Darstellung der Hierarchieebenen und die effiziente Anfrageverarbeitung innerhalb der Dimensionen. Die angeführten Vorteile lassen das Star Schema zu einem anerkannten Modell für die multidimensionale, logische Modellierung eines Data Warehouses auf relationaler Basis werden (vgl. Bauer/Günzel (2001), S. 203; Hahne (1999), S. 169).

Nach diesen Ausführungen und der Aussage, dass das klassische Star Schema für kleinere und mittelgroße Datenmengen den anderen Varianten vorzuziehen ist, soll es weiterführend die Grundlage für das logische Data Warehouse Datenmodell bilden. Es sollen jedoch mehrere Analysezwecke nach dem Prinzip der Data Marts abgebildet werden (Mitarbeiterkapazitäten, Projekt- und Ressourcenplanung). Des Weiteren wird zur Konsistenzsicherstellung empfohlen (s. Abschnitt 2.3.4), dass sich Fakttabellen strukturidentische Dimensionen teilen. Daher entwerfen wir eine Galaxie als logisches Data Warehouse Datenmodell. Abb. 4.3 (s. S. 117) stellt die entworfene Galaxie dar. Es ist zu erkennen, dass drei Fakttabellen (Arbeitszeitfakt, Projektkostenfakt, Ressourcenkostenfakt) dem semantischen Modell entsprechend verwendet werden. Weiterhin werden die Hierarchien der Dimensionen in jeweils einer Dimensionstabelle abgelegt.

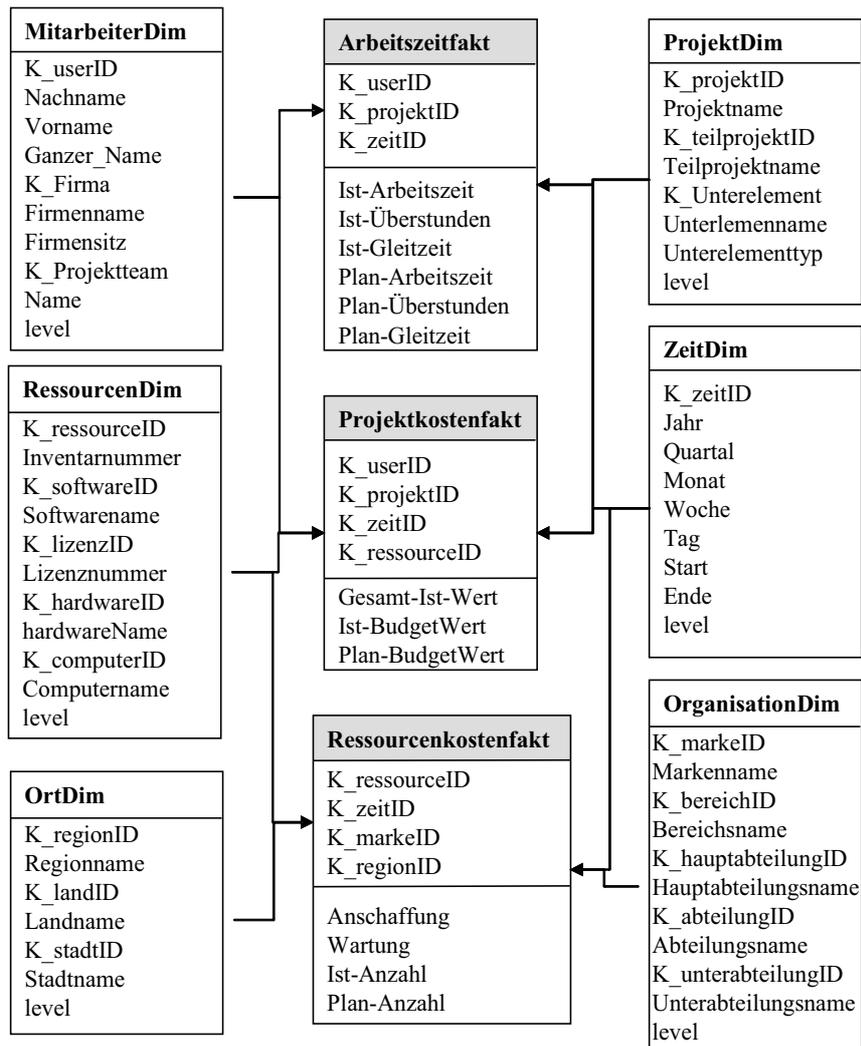


Abb. 4.3: Logisches Data Warehouse Datenmodell

Die level-Attribute werden verwendet, damit die Hierarchieebenen unterschieden werden können und keine Mehrfachzählungen der Aggregate vorkommen. Der untersten Hierarchieebene wird das level 0 zugewiesen. Die Aggregate spiegeln sich anhand von NULL-Werten innerhalb der Dimensionen wider. Tab. 4.2 (s. S. 118) zeigt beispielhaft eine stark vereinfachte Zeitdimensionstabelle. Anhand der NULL-Werte bei Jahr einerseits und bei Jahr und Monat andererseits, sind die Aggregate für die Jahressumme und die Gesamtsumme erkennbar. Anhand des logischen Data Warehouse Datenmodells können die ETL-Prozesse zur Datenextraktion, -transformation und -speicherung im DWH entworfen werden.

Tab. 4.2: Dimensionstabelle: *Zeit*

| <i>K_zeitID</i> | <i>Jahr</i> | <i>Monat</i> | <i>level</i> |
|-----------------|-------------|--------------|--------------|
| 1 | 2006 | Januar | 0 |
| 2 | 2006 | Februar | 0 |
| 3 | 2006 | März | 0 |
| 4 | 2006 | April | 0 |
| 5 | 2006 | Mai | 0 |
| 6 | 2006 | Juni | 0 |
| 7 | 2006 | Juli | 0 |
| 8 | 2006 | August | 0 |
| 9 | 2006 | September | 0 |
| 10 | 2006 | Oktober | 0 |
| 11 | 2006 | November | 0 |
| 12 | 2006 | Dezember | 0 |
| 13 | 2006 | NULL | 1 |
| 14 | NULL | NULL | 2 |

In Anlehnung an: Hahne (1999), S. 155

Insgesamt lässt sich festhalten, dass nach dem Entwurf des semantischen und logischen DWH Datenmodells, die Grundlage für die Implementierung eines DWH gelegt wurde. Mit einem DWH allein kann jedoch noch keine effektive Entscheidungsunterstützung gewährleistet werden, da dies ausschließlich den Daten-Layer repräsentiert. Daher ist es notwendig, eine komplette DWH Architektur aufzubauen, die einen Daten-Layer als Datenspeicher, einen Logik-Layer zur Informationsgenerierung und einen Präsentations-Layer für den anwenderfreundlichen Zugriff beinhaltet.

Im folgenden Abschnitt wird darauf eingegangen, welche DWH Architektur unter Berücksichtigung der Verwendung von BO und PMM als MSS verfolgt werden soll, um eine Optimierung der Entscheidungsunterstützung zur Projekt- und Ressourcenplanung gewährleisten zu können.

4.2.3 Data Warehouse Architektur

Für Planungs- und Kontrollsysteme besteht nicht die Dringlichkeit, hochaktuelle Daten im DWH bereitzustellen (vgl. Kemper et al. (2004), S. 86). Daher ziehen wir für das DWH zur Projekt- und Ressourcenplanung das klassische Data Warehousing heran, wonach die Daten in zyklischen Abständen – einen Tag, eine Woche oder einen Monat – durch ETL-Prozesse in das DWH geladen werden. Im Konkreten ist es sinnvoll, die Daten in Tageszyklen in das DWH zu überführen. Somit entsteht durch die zyklischen Erweiterungen eine echte Historisierung im DWH, wodurch Trendanalysen ermöglicht werden.

Des Weiteren lassen die Voraussetzungen des betrachteten Automobilherstellers und der Entwurf einer Galaxie nur ein DWH auf relationaler Basis zu. Damit verfolgen wir als Client-Server-Umsetzungsvariante einen *Relational OLAP*⁶⁹-Ansatz. Beim ROLAP werden die Verwandten des Star Schemas zur Erreichung der multidimensionalen Strukturen eingesetzt. Damit wird eine hohe Stabilität und Sicherheit bei großen Datenmengen und einer Vielzahl von Benutzern sichergestellt (vgl. Kemper et al. (2004), S. 100).

In Bezug auf die DWH Architektur wurde bereits während der Potenzialanalyse verdeutlicht (s. Abschnitt 4.1.2), dass eine *virtuelle DWH Architektur* für die Umsetzung nicht in Frage kommt. Aber es sollen jedoch die übrigen Architekturvarianten gegeneinander abgewogen werden, um für eine bestmögliche Entscheidungsunterstützung zur Projekt- und Ressourcenplanung zu gewährleisten.

Bei der *zentralen DWH Architektur* wird die Performance gegenüber dem virtuellen DWH verbessert, da Anfrageberechnungen und Auswertungen auf dem Server stattfinden und nur die tatsächlich benötigten Daten an den Client übermittelt werden. Weiterhin entfallen die zusätzliche Speicherung des Datenmodells und die Synchronisation mit den OLTP-Systemen, da die lokalen Schemata der Front-End-Werkzeuge entfallen (vgl. Kemper et al. (2004)). Eine Speicherung aller dispositiven Daten in einem DWH hat jedoch zur Folge, dass mehrere Anwendungsfälle abgebildet werden müssen. Dies kann ohne Trennung der Anwendungsfälle zu einer Unübersichtlichkeit der betriebswirtschaftlichen Sicht und somit zur Demotivation der Mitarbeiter führen.

Eine weitere Variante stellt eine *dezentrale DWH Architektur* aus verschiedenen Data Marts dar, da der Grundgedanke von Data Marts die größtenteils unabhängige Haltung von Abteilungsdaten vorsieht (vgl. Schinzer/Bange (1999), S. 52). Letzteres bestätigt die Ableitung von drei Fakttabellen im semantischen und logischen DWH Datenmodell,

⁶⁹ Beim relational OLAP sorgt eine Drei-Stufen-Architektur für den Zugriff von Client-Werkzeugen auf relationale Datenbanken mit einer multidimensionalen Struktur (vgl. Schinzer/Bange (1999), S. 57).

die für jeden Anwendungsfall speziell aufbereitete Daten liefern. Jedoch müssen bei der Aufbereitung in Data Marts die mehrfach verwendeten Dimensionstabellen redundant in den Data Marts gehalten werden. Dies bedeutet einen zusätzlichen Aufwand während der ETL-Phase und während der späteren Wartung.

Ferner wurde in bereits eine komplexere DWH Architekturvariante detailliert vorgestellt, die *ODS erweiterte DWH Architektur*. Der Vorteil dieser Architektur liegt in der komplexen und gemeinsamen Bereitstellung von sowohl hochaktuellen, detaillierten, unternehmensweiten, subjektorientierten als auch abteilungsorientierten dispositiven Daten. Allerdings wurde zu Beginn dieses Abschnitts beschlossen, dass auf eine hochaktuelle DWH-Variante verzichtet werden soll, da dies zur Planung und Kontrolle nicht unbedingt nötig ist. Daher ist im vorliegenden Fall davon abzuraten, einen ODS in die DWH Architektur zur Projekt- und Ressourcenplanung zu implementieren.

Die Bereitstellung aller dispositiven Daten in einem C-DWH und die erneute Verdichtung und anwendungsfallorientierte Aufbereitung in Data Marts ist allerdings durchaus vorteilhaft. Unter diesem Aspekt und unter der Berücksichtigung, dass BO als MSS Verwendung finden sollte, empfehlen wir, eine C-DWH basierte Architektur aufzubauen und diese an den BI-Ordnungsrahmen aus Abschnitt 2.2 anzugleichen (s. Abb. 4.4, S. 121).

Demnach können die Daten aus der operativen Datenbank von PMM durch ETL-Prozesse in die in 4.2.2 entworfene multidimensionale Struktur einer Galaxie (s. Abschnitt 4.2.2) überführt und im Daten-Layer als dispositive Daten bereitgestellt werden.

Der Daten-Layer soll die Vorbereitung der Standard-Reports in CR.NET und die Bereitstellung von anwendungsfallorientierten Data Marts durch Universen in BO für die Ad-hoc-Abfrage und -Analyse bereitstellen. Daher ist es als Vorteil zu sehen, dass die Daten für jeden speziellen Analysezweck (Mitarbeiterkapazitäten, Projekt- oder Ressourcenplanung) unabhängig voneinander aufbereitet werden und anschließend innerhalb der Universen komplexe Funktionen und Berechnungen durchgeführt werden können. Anhand von Abb. 4.4 (s. S. 121) ist ersichtlich, dass sowohl BO als auch CR.NET die Daten aus dem C-DWH beziehen, um sie anschließend für den Logik-Layer bereitzustellen.

Anschließend sollen im Logik-Layer Informationen aus den aufbereiteten Daten des Daten-Layers generiert werden. Zusätzlich sollen Standard-Reports direkt aus dem PMM-Front-End aufgerufen werden. Des Weiteren sollen Ad-hoc-Abfragen und -Analysen über BO Webintelligence und Desktop Intelligence ermöglicht werden. Durch die zugrunde liegende multidimensionale Struktur und der Eigenschaften von BO

ist es dann auch möglich, OLAP-Analysen durchzuführen. Damit sind alle in BO vorhandenen OLAP-Operationen uneingeschränkt verwendbar. Dies sind im Konkreten Pivotierung, Drill-down, Roll-up sowie Slice & Dice.

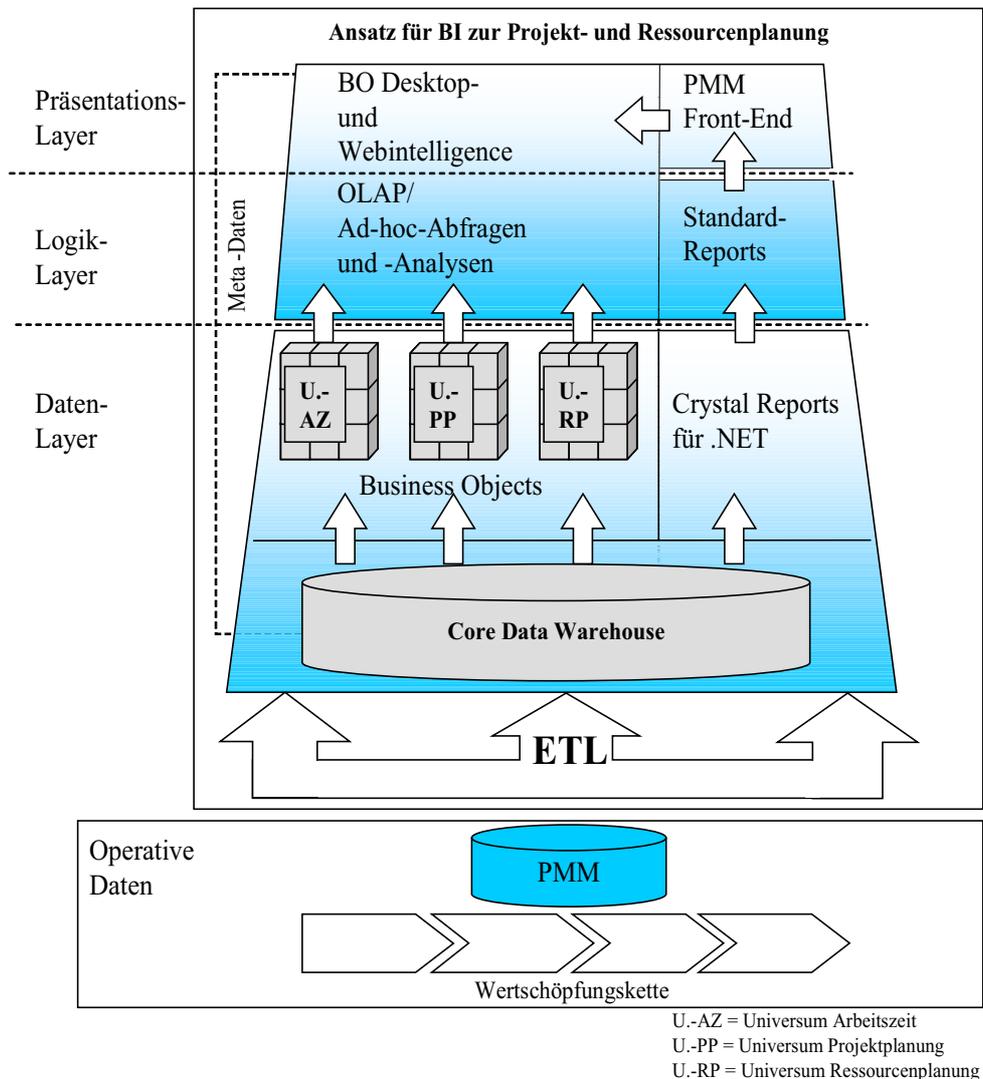


Abb. 4.4: Ansatz für BI zur Projekt- und Ressourcenplanung

Der Präsentations-Layer wird mit dem Logik-Layer eng verknüpft, da das PMM-Front-End sowohl die integrierten Standard-Reports als auch die Schnittstelle zur Endanwendersicht von BO bereitstellt. Dadurch gelangen die Endanwender auf benutzerfreundliche Weise an Informationen zur Entscheidungsunterstützung. PMM kann weiterhin als Front-End verwendet werden und vereinigt somit wiederum die konzeptorientierten Systeme mit den generischen Basissystemen, wie es bereits in Abschnitt 3.2.1 der Fall war. Diese Architekturvariante verfolgt das Prinzip einer isolierten und anwendungsfallorientierten, dispositiven Datenbasis und vereint alle in Abschnitt 4.1.3 genannten Vorteile einer dispositiven Datenbank. Damit kann sie optimal zur Entscheidungsunterstützung während der Projekt- und Ressourcenplanung beitragen.

In den vorangegangenen Abschnitten wurde zunächst ein semantisches und darauf aufbauend ein logisches DWH Datenmodell in den Phasen Datenselektion und Datenmodellierung entworfen. Das logische DWH Datenmodell bildet die Grundlage für die spätere Implementierung. Weiterführend wurde die Architektur in Anlehnung an den Ordnungsrahmen aus Abschnitt 2.2 in drei Schichten unterteilt. Der Daten-Layer basiert auf einem C-DWH, das Data Marts für OLAP-Analysen und CR.NET für Standard-Reports beliefert. Darauf aufbauend können mit Hilfe der Komponenten im Logik-Layer Informationen generiert werden. Durch das PMM-Front-End im Präsentations-Layer sind die Informationen übersichtlich und anwenderfreundlich abrufbar. Diese Ausführungen sind konzeptioneller Natur, daher müssen weitere Phasen zum Aufbau eines DWH durchlaufen werden. Erst nach dem kompletten Aufbau eines DWH kann eine effiziente Entscheidungsunterstützung gewährleistet werden.

Im Folgenden wird auf die weitere Vorgehensweise zur Einführung dieser Architekturvariante inklusive des DWH eingegangen.

4.2.4 Weitere Vorgehensweise zur Einführung eines Data Warehouse

Bis zur Inbetriebnahme der DWH Architektur, die im vorangegangenen Abschnitt (s. 4.2.3) entworfen wurde, sind noch weitere, komplexe Phasen zum Aufbau eines DWH zu durchlaufen. Dazu gehören die Datenextraktion, Datentransformation und der Datenimport sowie die kontinuierlichen Phasen Betrieb und Pflege.

Abgesehen von den zu administrierenden Hard- und Softwareressourcen ist eine Phase von ganz besonderer Bedeutung: der ETL-Prozess, der die Datenextraktion, -transformation und den Datenimport umfasst. Für die Phase des ETL-Prozesses stehen meist komplexe ETL-Tools zur Verfügung, um den Entwicklern diesen Prozessschritt zu vereinfachen. ETL-Tools beinhalten unter anderem bereits Algorithmen zur Datenselektion, Datentypkonvertierung und zur Hierarchieerstellung aus operativen Tabellen. Letzteres wird als „Hierarchie Flattening“ bezeichnet. Für die Phasen des ETL-Prozesses der Filterung, Harmonisierung, Verdichtung und Anreicherung werden ETL-Tools verwendet, um komplexe Transformationen durchzuführen und die dispositiven Daten anschließend in das DWH zu laden. Die ETL-Prozesse werden nach Umsetzung automatisiert und täglich ausgeführt. Erst nach der Datenextraktion, -transformation und -speicherung im C-DWH stehen diese verdichtet als dispositive Daten zur Verfügung. Anschließend müssen in CR.NET die Standard-Reports an die neue Datenbasis angepasst werden. Es wird nun jedoch einfacher gelingen, die gewünschten Formate und Berechnungen zu erhalten, da diese bereits im C-DWH vorhanden sein

werden. Die Universenerstellung im BO Designer hält sich weiterhin an das Vorgehensmodell aus Abschnitt 3.3.1. Es solle wie in Abschnitt 4.2.3 vorgeschlagen für jeden Anwendungsfall ein Universum erstellt werden.

Nach erfolgreicher Umsetzung der Universen und Neuerstellung der Standard-Reports müssen diese mit dem PMM-Front-End verbunden werden, um den anwenderfreundlichen Zugriff zu gewährleisten. Anschließend kann auf eine optimierte, dispositive Datenbasis zur Steuerung, Kontrolle und Analyse der Projekt- und Ressourcenplanung zurückgegriffen werden.

Die Phasen, Betrieb und Pflege müssen nach Umsetzung des DWH Konzeptes kontinuierlich ausgeführt werden, um die Entscheidungsunterstützung in Zukunft zu gewährleisten.

In Bezug auf die BI-Strategie des betrachteten Automobilherstellers können die dispositiven Daten im Verlaufe der Standardisierung direkt in die Gravitationszentren geladen, harmonisiert und für Data Marts aufbereitet werden. Da sich zurzeit die Umsetzung der BI-Strategie in der Vorbereitung befindet, wird auf die Vorteile einer unternehmensweiten BI-Strategie noch verzichtet werden müssen.

Dennoch kann die Umsetzung des entworfenen DWH Konzeptes die Problematik der erkannten Schwachstellen lösen. Denn durch die DWH basierte Architektur können die Performanceprobleme, wie unter anderem übermäßig viele Joins, Berechnungen zur Laufzeit oder Belastung des operativen Betriebs, entfallen. Weiterhin können dem Entscheider mehr Funktionalitäten in Bezug auf die Analyse der Datenbestände zur Verfügung stehen, da aufgrund der multidimensionalen Struktur nun auch komplexe OLAP-Analysen möglich sein werden. Dies führt zu einer verbesserten Entscheidungsunterstützung und zu effizienteren Geschäftsprozessen. Daraus entstehen zunächst nur in der IT-Abteilung Wissensvorsprünge, die allerdings nach Umsetzung der unternehmensweiten BI-Strategie zu Wettbewerbsvorteilen des Unternehmens auf dem globalen Markt führen können.

4.3 Resümee

Zu Beginn des vierten Kapitels wurden Schwachstellen der BI-Komponenten zur Projekt- und Ressourcenplanung des praktischen Beispiels durch eine Potenzialanalyse aufgezeigt. Anschließend wurde durch die Analyse der vorhandenen Potenzialfelder deutlich, dass eine operative Datenbank als Grundlage für Reporting- und Analysefunktionalitäten nicht ausreichend ist. Nach Gegenüberstellung der Eigenschaften von operativen und dispositiven Datenbanken wurde der Ansatz bestärkt, ein DWH als dispositive Datenbasis aufzubauen, das die Grundlage für die Reporting- und Analysefunktionalitäten zur Entscheidungsunterstützung darstellt. Dies kann eine verbesserte Planung zur Folge haben und dadurch Zeit und Kosten einsparen.

Weiterführend wurde zunächst ein semantisches DWH Modell in Form eines mERM entworfen, welches das Fachkonzept der Projekt- und Ressourcenplanung abbildet und die nötigen Informationen übersichtlich darstellt. Darauf aufbauend wurde ein logisches DWH Modell in Form einer Galaxie entworfen, das die Grundlage für die spätere Umsetzung der relationalen DWH Datenbankstruktur bieten kann. Im darauf folgenden Abschnitt wurde eine DWH Architektur entworfen, die zu einer optimierten Entscheidungsunterstützung einer Projekt- und Ressourcenplanung dienen kann. Dies leitet sich aus der Verbesserung der Datenqualität aufgrund der Speicherung der aufbereiteten Daten in einem separaten DWH ab. Die vorgeschlagene Architektur setzt sich im Konkreten aus drei Schichten zusammen, und zwar aus einem Daten-Layer und einer engen Verknüpfung zwischen Logik- und Präsentations-Layer. Die dispositiven Daten sollen durch ETL-Prozesse aus der derzeit einzigen operativen Datenbank in ein separates C-DWH im Daten-Layer geladen werden. Das C-DWH soll anschließend drei durch Universen in BO – anwendungsfallorientiert aufgebaute Data Marts – und CR.NET speisen. Im darüber liegenden Logik-Layer sollen Informationen durch übersichtliche Standard-Reports und OLAP-Analysen generiert werden. Diese Informationen sollen anwenderfreundlich durch das PMM-Front-End und die Schnittstelle zu BO im Präsentations-Layer abrufbar sein.

Weiterhin zeigte die weitere Vorgehensweise zur Einführung dieser Architekturvariante auf, dass noch einige wichtige Phasen, darunter die ETL-Umsetzung, durchlaufen werden müssen, um auf eine optimierte DWH Architektur zur Entscheidungsunterstützung der Projekt- und Ressourcenplanung zurückgreifen zu können.

Insgesamt lässt sich festhalten, dass bereits durch die Integration der BI-Komponenten in PMM die Projekt- und Ressourcenplanung der Digitalen Fabrik- und Fertigungsplanung effizienter gestaltet wurde, indem die Qualität der Entscheidungsunterstützung verbessert wurde. Daher kann der Zeitaufwand während Entscheidungen des täglichen

Geschäfts reduziert und Kosten eingespart werden. Darüber hinaus kann eine Umsetzung des BI-Ansatzes und die Einführung einer DWH Architektur – wie ab Abschnitt 4.2 beschrieben – die Entscheidungsunterstützung zusätzlich hinsichtlich Zeit, Kosten und Qualität optimieren. Dies liegt darin begründet, dass die Verbesserung der Datenqualität bezüglich der Datenbereitstellung sowie die Erhöhung der Performance und die Funktionserweiterung zu einer enormen Steigerung der Reporting- und Analysefunktionalitäten führen können. Dadurch kann sich die Entscheidungsunterstützung wesentlich effektiver gestalten lassen und wird nach Durchführung weiterer Maßnahmen bezüglich der BI-Strategie zu Wettbewerbsvorteilen des Unternehmens auf dem weltweiten Markt führen.

5 Schlussbetrachtung

Zu Beginn dieser Diplomarbeit wurden die aktuellen Entwicklungen im Bereich Business Intelligence vorgestellt. Weiterführend wurden die damit verbundenen Konzepte, Technologien und Systeme als integrierter, unternehmensspezifischer, IT-basierter Gesamtansatz zur betrieblichen Entscheidungsunterstützung beschrieben.

In den aktuellen Forschungen entwickelt sich die Tendenz, eine unternehmensweite BI-Strategie zu verfolgen. Diese unternehmensweite Betrachtung wurde jedoch bei der Integration von BI-Komponenten an einem praktischen Beispiel aufgrund der gegebenen Voraussetzungen nicht betrachtet. Es wurde aus finanziellen, zeitlichen und organisatorischen Gründen eine BI-Abteilungslösung entworfen und umgesetzt. Dabei handelt es sich um die Integration von BI-Komponenten zur Entscheidungsunterstützung der Projekt- und Ressourcenplanung innerhalb der Digitalen Fabrik- und Fertigungsplanung. Es wurde gezeigt, dass die Umsetzung und Verwendung der BI-Komponenten zu einer enormen Verbesserung der Entscheidungsunterstützung beiträgt. Zurückblickend auf die Ausgangssituation waren nur statische Auswertungen zur Entscheidungsunterstützung möglich. Nach Integration der BI-Komponenten in die Projektmanagement-Anwendung sind nun sowohl teilweise dynamische Standard-Reports als auch Ad-hoc-Abfragen und Analyse möglich. Diese Möglichkeiten zur Entscheidungsunterstützung verbessern somit die Planungsqualität. Dadurch wird sich der Zeitaufwand des Prozesses zur Entscheidungsfindung im Wesentlichen reduzieren und demnach werden Kosten eingespart.

Jedoch ist zu betonen, dass Optimierungspotenziale auch innerhalb der entwickelten BI-Lösung vorliegen, welche im Abschluss dieser Arbeit aufgezeigt wurden. Die Implementierung des vorgestellten Data Warehouse Ansatzes kann eine zusätzliche Steigerung der Reporting- und Analysefunktionalitäten zur Entscheidungsunterstützung bewirken. Dies resultiert aus der Steigerung der Datenqualität, der Optimierung der Performance und der verbesserten Kostenbetrachtungen sowie der Funktionserweiterungen der Analyse.

Zukünftig sollte beim betrachteten Automobilhersteller der vorgestellte DWH Ansatz umgesetzt und anschließend in die unternehmensweite BI-Strategie aufgenommen werden. Dies kann zusätzlich die Entscheidungsunterstützung des täglichen Geschäfts verbessern.

Abschließend lässt sich festhalten, dass im Zentrum von Business Intelligence stets der Wunsch steht, Wirkungszusammenhänge und Mechanismen, die für das eigene Geschäft relevant sind, durch Analysen vorhandener Datenbestände zu entdecken und besser verstehen zu können, um die Geschäftsprozesse effizienter gestalten zu können (vgl. Gluchowski (2001), S. 15). Deshalb muss, unterstützt durch die aktuellen Forschungen und Entwicklungen, ein integrierter, unternehmensspezifischer, IT-basierter Gesamtansatz im gesamten Unternehmen verfolgt werden, um mit Hilfe von BI aus allen verfügbaren Datenbeständen entscheidungsunterstützende Informationen abzuleiten. Daraus entsteht Wissen über die unternehmensinternen und -externen Informationen und über die zukünftigen Entwicklungen. Dieses Wissen wird nicht nur in einzelnen Abteilungen, sondern im gesamten Unternehmen genutzt und zur Erlangung von Wettbewerbsvorteilen eingesetzt. Damit ist und bleibt ein Unternehmen in Zukunft durch den Einsatz von Business Intelligence konkurrenzfähig und erfolgreich.

Anhang

A. Ressourcenplanung in Microsoft Excel

| Systemname | Presswerk | | Karobau | | Lack | | Montage | | Fördertechnik u. Beläuer | | Werkzeug- und Vorrichtungsbau | | Aggregate | | Konsumplanung | | Kunststoffteile | | Logistik | | Ebau/Infrastruktur | | Sonsige | | Fertigung/Produktion | | | |
|----------------------------------|-----------|------|---------|------|------|------|---------|------|--------------------------|------|-------------------------------|------|-----------|------|---------------|------|-----------------|------|----------|------|--------------------|------|---------|------|----------------------|------|----|----|
| | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | ist | soil | | |
| 10 AP | 25 | 23 | 16 | 14 | 10 | 9 | 160 | 30 | 0 | 0 | 0 | 0 | 23 | 15 | 8 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 881 | ## | 50 | 50 | |
| 46 CATIA | 16 | 7 | 11 | 7 | 10 | 10 | 43 | 34 | 1 | 1 | 161 | 158 | 26 | 26 | 20 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 485 | 522 | 0 | 0 | |
| 50 CATIA | 18 | 7 | 10 | 7 | 11 | 10 | 37 | 34 | 1 | 1 | 160 | 158 | 26 | 26 | 21 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 501 | 455 | 7 | 0 | |
| 52 CATIA | 17 | 7 | 9 | 7 | 0 | 0 | 20 | 18 | 1 | 1 | 158 | 158 | 0 | 0 | 21 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 430 | 539 | 0 | 0 | |
| 63 DMU | 9 | 9 | 8 | 8 | 12 | 12 | 78 | 78 | 0 | 0 | 10 | 10 | 1 | 3 | 36 | 41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 406 | 484 | 0 | 0 | |
| 149 KYS | 22 | 21 | 25 | 15 | 3 | 0 | 262 | 228 | 0 | 0 | 568 | 550 | 100 | 100 | 56 | 55 | 0 | 0 | 0 | 0 | 8 | 0 | 1 | 155 | 161 | 0 | 0 | |
| 150 KYS | 8 | 6 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 551 | 550 | 1 | 1 | 55 | 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 437 | 437 | 0 | 0 | |
| 151 KYS | 48 | 46 | 164 | 161 | 112 | 108 | 436 | 383 | 35 | 35 | 967 | 550 | 85 | 85 | 55 | 55 | 0 | 0 | 0 | 0 | 28 | 22 | 1 | ## | ## | 30 | 30 | |
| 221 Product-View | 22 | 0 | 31 | 0 | 23 | 0 | 172 | 0 | 33 | 0 | 171 | 120 | 82 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 501 | 573 | 5 | 5 |
| 279 ZG-View | 52 | 0 | 248 | 0 | 209 | 0 | 436 | 0 | 42 | 0 | 30 | 0 | 137 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 187 | 0 | 35 | 0 | ## | ## | 70 | 70 |
| 282 AP - gesamt | 25 | 23 | 16 | 14 | 10 | 9 | 193 | 61 | 0 | 0 | 0 | 0 | 23 | 15 | 8 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 892 | ## | 50 | 50 | |
| 283 AutoCAD - gesamt | 0 | 0 | 8 | 0 | 4 | 0 | 19 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 0 | 34 | 5 | 1 | 0 |
| 284 CATIA - gesamt | 18 | 7 | 15 | 11 | 11 | 10 | 44 | 34 | 1 | 1 | 162 | 159 | 26 | 26 | 21 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 717 | 676 | 7 | 0 |
| 285 COMOS ET - gesamt | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 74 |
| 286 JCADE / eM-Stamping - gesamt | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 |
| 287 DMU - gesamt | 9 | 9 | 8 | 8 | 12 | 12 | 78 | 78 | 0 | 0 | 13 | 13 | 1 | 3 | 36 | 41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 426 | 504 | 0 | 0 |
| 288 DPE - gesamt | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 137 | 147 | 0 | 0 |
| 289 DPM - gesamt | 0 | 0 | 0 | 0 | 4 | 8 | 26 | 27 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 | 126 | 0 | 0 |
| 290 eM-Designer - gesamt | 0 | 0 | 17 | 0 | 0 | 0 | 6 | 5 | 0 | 0 | 19 | 31 | 3 | 3 | 9 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 51 | 501 | 0 | 0 |
| 291 eM-Engineer - gesamt | 0 | 0 | 4 | 0 | 1 | 0 | 1 | 0 | 0 | 13 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 292 eM-Machining - gesamt | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 7 | 5 | 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 28 | 0 | 0 |

Abb. A.1: Bebauungsplanung in Microsoft Excel

| Kernprozesse und -systeme der DF | | Verantwortlich | Aufgabe | 2006 | 2007 | 2008 | 2009 | 2010 | Gesamt |
|----------------------------------|---|----------------|--------------------|------|------|------|------|-------|--------|
| 18 | Thema: DF Prozesse | IT P/ID | Prozessberatung | 0 | 0 | 0 | 0 | 2.200 | 220 |
| 19 | | IT P/ID | Anwendungsberatung | 0 | 0 | 0 | 0 | 1.150 | 115 |
| 20 | | FB | Eigenleistung | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | | FB | Fremdleistung | 0 | 0 | 0 | 0 | 480 | 48 |
| 22 | | | | 0 | 0 | 0 | 0 | 480 | 48 |
| 23 | | | | 0 | 0 | 0 | 0 | 480 | 48 |
| 24 | Kern DF | | | 218 | 218 | 73 | 73 | 0 | 2.230 |
| 25 | | IT P/ID | Entwicklung | 180 | 180 | 60 | 60 | 0 | 250 |
| 26 | | IT GOI | Server-Netz | 0 | 0 | 0 | 0 | 0 | 500 |
| 27 | | IT GOM | SV-Wartung | 0 | 0 | 0 | 0 | 0 | 50 |
| 28 | | IT GOM | Betrieb | 0 | 0 | 0 | 0 | 0 | 1.000 |
| 29 | | FB | Eigenleistung | 38 | 38 | 13 | 13 | 0 | 480 |
| 30 | | FB | Fremdleistung | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | | FB | SV-Invest | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | | FB | HV-Invest | 0 | 0 | 0 | 0 | 0 | 0 |
| 33 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 34 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 35 | Unterst. Prozesse u. Systeme -> Entwicklung | | | 45 | 45 | 15 | 15 | 0 | 470 |
| 36 | Thema: Prozesse und Schnittstellen | | | 45 | 45 | 15 | 15 | 0 | 470 |
| 37 | | IT P/ID | Prozessberatung | 0 | 0 | 0 | 0 | 0 | 150 |
| 38 | | IT P/ID | Eigenleistung | 45 | 45 | 15 | 15 | 0 | 0 |
| 39 | | IT GOI | Server-Netz | 0 | 0 | 0 | 0 | 0 | 20 |
| 40 | | IT GOM | Betrieb | 0 | 0 | 0 | 0 | 0 | 150 |
| 41 | | FB | Eigenleistung | 0 | 0 | 0 | 0 | 0 | 120 |
| 42 | | FB | Fremdleistung | 0 | 0 | 0 | 0 | 0 | 12 |
| 43 | Unterst. u. Systeme -> Serie | | | 5 | 5 | 2 | 2 | 0 | 30 |
| 44 | Thema: Prozesse und Schnittstellen | | | 5 | 5 | 2 | 2 | 0 | 30 |
| 45 | | IT P/ID | Prozessberatung | 0 | 0 | 0 | 0 | 0 | 15 |
| 46 | | IT P/ID | Eigenleistung | 5 | 5 | 2 | 2 | 0 | 15 |
| 47 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 48 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 49 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 50 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 51 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 52 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 53 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 54 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 55 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 56 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 57 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 58 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 59 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 60 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 61 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 62 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 63 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 64 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 65 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 66 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 67 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 68 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 69 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 70 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 71 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 72 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 73 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 74 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 75 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 76 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 77 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 78 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 79 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 80 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 81 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 82 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 83 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 84 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 85 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 86 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 87 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 88 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 89 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 90 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 91 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 92 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 93 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 94 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 95 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 96 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 97 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 98 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 99 | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 100 | | | | 0 | 0 | 0 | 0 | 0 | 0 |

Abb. A.2: Softwareanschaffungs- und Softwarewartungskostenplanung in MS Excel

B. Finanz- und Kapazitätenplanung in Microsoft Access

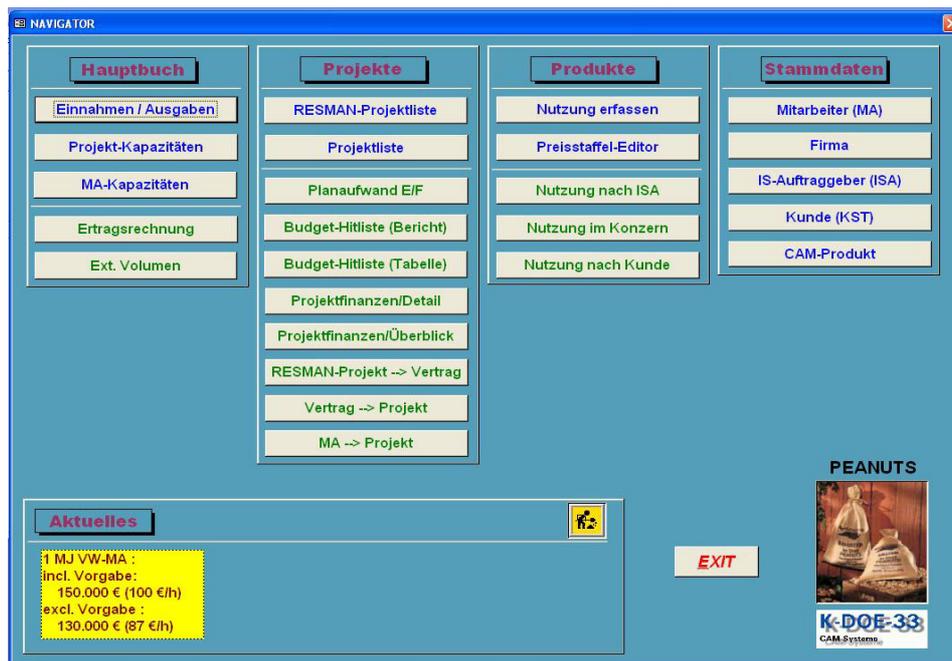


Abb. B.1: Programm „Peanuts“ zur Finanz- und Kapazitätenplanung

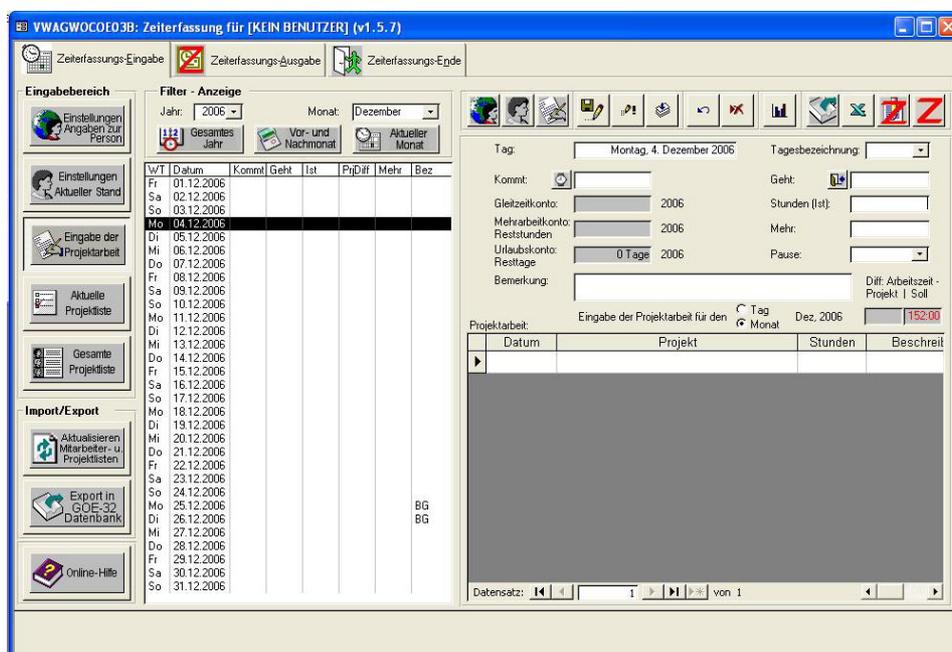


Abb. B.2: Programm „Time“ zur Zeiterfassung

C. BusinessObjects Universen

Klassen und Objekte für Entwickler und Endanwender sichtbar

Universum-Datenbankschema nur für Entwickler sichtbar

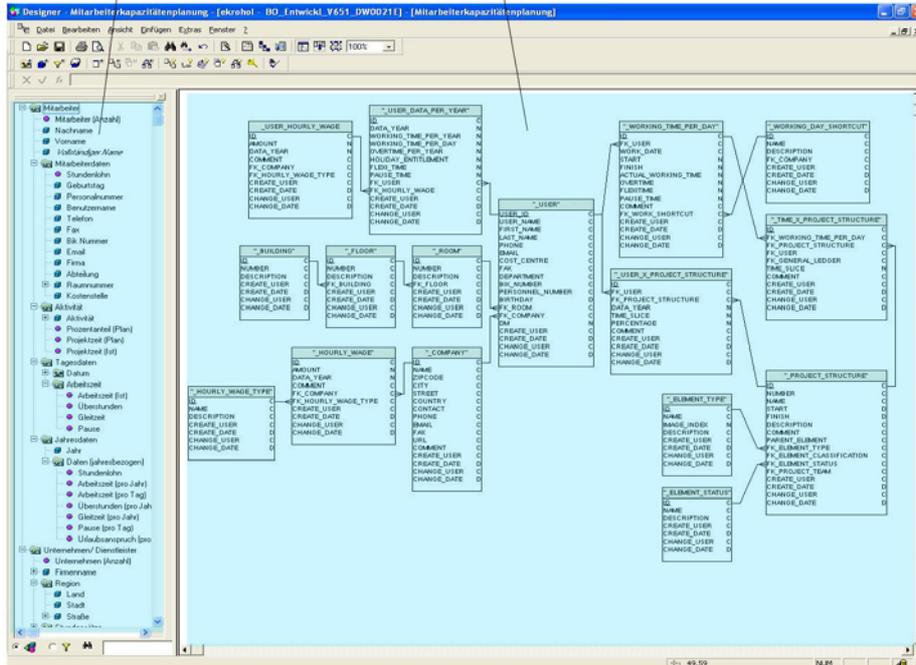


Abb. C.1: BO Designer – Universum Mitarbeiterkapazitätenplanung

Klassen und Objekte für Entwickler und Endanwender sichtbar

Universum-Datenbankstruktur nur für Entwickler sichtbar

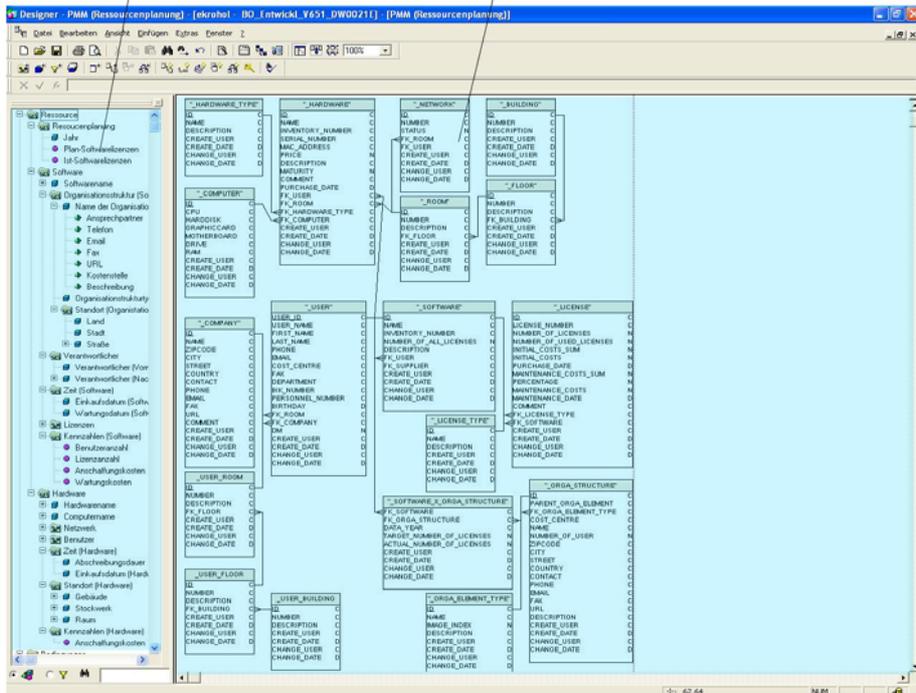


Abb. C.2: BO Designer – Universum Ressourcenplanung

D. PMM Standard-Reports

Anwendername (Krohne, Oliver)

Monatsname (letzter Monat) (September 2006)

Komplette Zeiterfassung (Table with columns: Datum, TimeCode, Kommt, Geht, Pause, Ist, Mehr, Gleitzeit, Bemerkung)

Gruppen-ergebnisse (Summary statistics: Arbeitszeit (September): 150:28, Mehrarbeitszeit (September): 00:00, Gleitzeit (September): 00:00, Urlaubanspruch (2006): 20,0)

Abb. D.1: Standard-Report – Letzter Monat

Gruppenhierarchie (Mitarbeiter/ Jahr/ Monat) (Navigation tree on the left)

Täglich berichtete Aktivitäten für gewählten Mitarbeiter (Table with columns: Datum, Aktivität, Zeitaufwand (Stdn:min))

Verwendete Arbeitsstunden (Summary statistics: Summe der Arbeitszeit (Oktober): 214:53, Summe der Projektarbeitszeit (Oktober): 143:24, Kosten (Oktober): 9638,97 €)

Abb. D.2: Standard-Report – Projektarbeitszeiten der Mitarbeiter (tagesbezogen)

Literaturverzeichnis

- Anandarajan, M.; Anandarajan, A.; Srinivasan, C. A. (2004): Business Intelligence Techniques. Berlin et al.
- Bange, C. (2003): Business Intelligence – Systeme und Anwendungen: Werkzeuge und Technologien für die Unternehmenssteuerung. Würzburg.
- Bange, C. (2006): Werkzeuge für Business Intelligence. In: Heilmann et al. (2006), S. 63-73.
- Bange, C.; Schinzer, H. D. (2006): Data Warehouse und Business Intelligence: Grundlagen entscheidungsorientierter Informationssysteme.
[http://www.competencesite.de/bisysteme.nsf/BBE71828E145670BC125695400483F6B/\\$File/grundlagen%20-%20dw%20und%20bi-systeme.pdf](http://www.competencesite.de/bisysteme.nsf/BBE71828E145670BC125695400483F6B/$File/grundlagen%20-%20dw%20und%20bi-systeme.pdf).
 22. September 2006.
- Bauer, A.; Günzel, H. (Hrsg.) (2001): Data Warehouse: Architektur, Entwicklung und Anwendung. Heidelberg.
- Bauer, S.; Winterkamp, T. (1996): Relationales OLAP versus Mehrdimensionale Datenbanken. In: Hanning (1996), S. 45-53.
- Bissantz, N.; Hagedorn, J.; Mertens, P. (2000): Data Mining. In: Mucksch/Behme (2000), S. 377-407.
- Business Objects (Hrsg.) (2004a): Designerhandbuch. o. O.
- Business Objects (Hrsg.) (2004b): BusinessObjects-Anwenderhandbuch: Berichterstellungstechniken und Formatierung. o. O.
- Business Objects (Hrsg.) (2004c): BusinessObjects-Handbuch für erweiterte Abfrageerstellungstechniken. o. O.
- Business Objects (Hrsg.) (2004d): BusinessObjects-Anwenderhandbuch: Datenzugriff und -analyse. o. O.
- Business Objects (Hrsg.) (2004e): WebIntelligence für Einsteiger. o. O.
- Business Objects (Hrsg.) (2006): BusinessObjects XI Release 2.
<http://www.businessobjects.de/products/businessobjectsexi/default.asp>.
 03. Oktober 2006.
- Chamoni, P.; Gluchowski, P. (1997): Organisatorische Aspekte bleiben meist außen vor – Entwicklungstendenzen und Perspektiven der OLAP-Technologie. Computerwoche Fokus, o.Jg., Ausgabe 2, S. 24-26.
- Chamoni, P.; Gluchowski, P. (Hrsg.) (1998): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining. Berlin, u.a.
- Chamoni, P.; Gluchowski, P. (Hrsg.) (1999a): Analytische Informationssysteme: Data Warehouse, On-Line Analytical Processing, Data Mining. 2., neubearbeitete Auflage. Berlin, u.a.
- Chamoni, P.; Gluchowski, P. (1999b): Entwicklungslinien und Architekturkonzepte des On-Line Analytical Processing. In: Chamoni/Gluchowski (1999), S. 262-280.
- Chen, P.P.-S.(1976): The Entity Relationship Model Toward a Unified View of Data. In: ACM Transactions on Database Systems, Band 1, Nr. 1, S. 9-36.
- Codd, E. (1970): A relational model for large shared data banks. In: Communications of the ACM, Band 13, Nr. 6, S. 377-387.

- Codd, E. F.; Codd, S. B.; Salley, C. T. (1993): Providing OLAP to User-Analysts: An IT Mandate.
http://dev.hyperion.com/resource_library/white_papers/providing_olap_to_user_analysts.pdf. 22.08.2006.
- DIN (Hrsg.) (1972): DIN 44300: Informationsverarbeitung. Bonn.
- DIN (Hrsg.) (1987): DIN 69901: Grundbegriffe des Projektmanagements. Bonn.
- Drosdowski, G.(Hrsg.); Köster, R.; Müller, W.; Scholze-Stubenrecht, W.; Trunk, M. (1982): Duden Fremdwörterbuch, 4., neu bearbeitete und erweiterte Auflage. Mannheim, u.a.
- Feder Consulting(2006): Feder Consulting - Potentialanalyse.
<http://federconsulting.com/produkte/potentialanalyse.pdf>.
Stand: 2. November 2006.
- Gluchowski, P. (2001): Business Intelligence: Konzepte, Technologien und Einsatzbereiche. In: Hildebrand, K (2001), S. 7-15.
- Gluchowski, P.; Gabriel, R.; Chamoni, P. (1997): Management Support Systeme – Computergestützte Informationssysteme für Führungskräfte und Entscheidungsträger. Berlin, u.a.
- Gluchowski, P.; Kemper, H.-G. (2006): Quo Vadis Business Intelligence: Aktuelle Konzepte und Entwicklungstrends. BI-Spektrum, 1. Jg., Nr. 1, S. 12-19.
- Gronau, N. (Hrsg.); Dilz, S.; Kalisch, A. (2004): Anwendungen und Systeme für das Wissensmanagement – ein aktueller Überblick. Berlin.
- Grothe, M. (1999): Aufbau von Business Intelligence: Entwicklung einer softwaregestützten Controlling-Kompetenz bei o.tel.o. Düsseldorf.
- Grothe, M.; Gentsch, P. (2000): Business Intelligence – Aus Informationen Wettbewerbsvorteile gewinnen. München.
- Gutenberg, E. (1983): Grundlagen der Betriebswirtschaftslehre, Band 1: Die Produktion. 24. Auflage. Berlin, u.a.
- Hahne, M. (1999): Logische Datenmodellierung für das Data Warehouse: Bestandteile und Varianten des Star Schemas. In: Chamoni/Gluchowski (1999), S. 145-169.
- Hahne, M. (2002): Transformation mehrdimensionaler Datenmodelle. In: von Maur/Winter (2002), S. 399-420.
- Hanning, U. (Hrsg.) (1996): Data Warehouse und Managementinformationssysteme. Stuttgart.
- Hansen, H. R. (2002): Wirtschaftsinformatik I, 6., neubearbeitete und stark erweiterte Auflage. Stuttgart, Jena.
- Hansen, H. R.; Neumann, G. (2001): Wirtschaftsinformatik I. Grundlagen betrieblicher Informationsverarbeitung, 8. Auflage. Stuttgart.
- Heilmann, H.; Kemper, H.-G.; Baars, H. (2006): Business & Competitive Intelligence. HMD – Praxis der Wirtschaftsinformatik, 43. Jg., Heft 247. Heidelberg.
- Henke, M. (2006): Lastenheft für das Projektmanagementtool „PMM“ Version 0.1. Wolfsburg.
- Herden, H. (2001): Eine Entwurfsmethodik für Data Warehouses. Dissertation, Carl von Ossietzky Universität. Oldenburg.

- Herden, H.; Harren, A. (1999): MML und mUML – Sprache und Werkzeug zur Unterstützung des konzeptuellen Data Warehouse-Designs. 2. Workshop Data Mining und Data Warehousing als Grundlage moderner entscheidungsunterstützender Systeme (DMDW99), o.Jg., S. 57-67.
- Heuer, A.; Saake, G. (2000): Datenbanken: Konzepte und Sprachen. 2., aktualisierte und erweiterte Auflage. Bonn.
- Hildebrand, K. (Hrsg.) (2001): Business Intelligence. HMD – Praxis der Wirtschaftsinformatik, 38. Jg., Heft 222. Heidelberg.
- Holthuis, J. (1999): Der Aufbau von Data Warehouse-Systemen: Konzeption – Datenmodellierung – Vorgehen. 2., überarbeitete und aktualisierte Auflage. Wiesbaden.
- Holthuis, J. (2000): Grundüberlegungen für die Modellierung einer Data Warehouse-Datenbasis. In: Mucksch/Behme (2000), S. 149-186.
- Horváth, P. (2003). Controlling. 9. Auflage. München.
- Humm, B.; Wietek, F. (2005): Architektur von Data Warehouses und Business Intelligence Systemen. Informatik Spektrum, o. Jg., 23. Februar 2005, S. 3-14.
- Inmon, W. H., Imhoff, C.; Battas, G. (1992): Building the Data Warehouse. New York et al. John Wiley & Sons.
- Inmon, W. H., Imhoff, C.; Battas, G. (1996): Building the Data Warehouse. Second Edition. New York et al.
- Inmon, W. H. (1999): Building the Operational Data Store. New York et al.
- Inmon, W. H.; Hackathorn, R. D. (1994): Using the Data Warehouse. New York et al.
- Jarosch, H. (2002): Datenbankentwurf – Eine beispielorientierte Einführung für Studenten und Praktiker. Braunschweig, Wiesbaden.
- Jung, R. (2001): Gestaltung einer datenintegrierenden Architektur für dispositive und operative Zwecke. In: Hildebrand (2001), S. 29-37.
- Kahaner, L. (1997): Competitive Intelligence: How to gather, analyze, and use information to move your business to the top. New York.
- Kemper, H.-G.; Finger, R. (1999): Datentransformation im Data Warehouse: Konzeptionelle Überlegungen zur Filterung, Harmonisierung, Verdichtung und Anreicherung operationaler Datenbestände. In: Chamoni/Gluchowski (1999), S. 77-94.
- Kemper, H.-G.; Mehanna, W.; Unger, C. (2004): Business Intelligence – Grundlagen und praktische Anwendungen: eine Einführung in die IT-basierte Managementunterstützung. Wiesbaden.
- Kemper, H.-G.; Baars, H. (2006): Business Intelligence und Competitive Intelligence: IT-basierte Managementunterstützung und markt-/wettbewerbsorientierte Anwendungen. In: Heilmann et al. (2006). S. 7-20.
- Koschnick, W. J. (1996): Management – Enzyklopädisches Lexikon. Berlin, New York.
- Klaus, A.; Voigt, K. (2006): Presseinformation 28-06/ 28. Juli 2006: Führendes Analyseunternehmen erklärt Business Objects abermals zum marktführenden Anbieter von Business Intelligence Lösungen.
http://www.businessobjects.de/news/pi_06_07_28.asp#. 01. Oktober 2006.

- Krallmann, H.; Frank, H.; Gronau, N. (1999): Systemanalyse im Unternehmen. 3., völlig überarbeitete und erweiterte Auflage. München, Wien.
- Kurz, A. (1999): Data Warehousing Enabling Technology. Bonn. MITP-Verlag.
- Litke, H.-D. (2004): Projektmanagement: Methoden, Techniken, Verhaltensweisen, Evolutionäres Projektmanagement. 4. Auflage. München, Wien.
- Martin, W. (Hrsg.) (1998): Data Warehousing: Data Mining – OLAP. Bonn.
- McAmis, D. (2004): Professional Crystal Reports for Visual Studio .NET. 2nd Edition. Indiana.
- Mertens, P. (Hrsg.); Back, A.; Becker, J.; König, W.; Krallmann, H.; Rieger, B.; Scheer, A.-W.; Seibt, D.; Stahlknecht, P.; Strunz, H.; Thome, R.; Wedekind, H. (1997): Lexikon der Wirtschaftsinformatik. 3. Auflage. Berlin, u.a.
- Mertens, P.; Giese, J. (2002): Integrierte Informationsverarbeitung 2. 9. Auflage. Wiesbaden.
- Messerschmidt, H.; Schweinsberg, K. (2003): OLAP mit dem SQL Server: Eine Einführung in Theorie und Praxis. 1. Auflage. Heidelberg.
- Microsoft (Hrsg.) (2006): Was sind MDAC, DA SDK, ODBC, OLE DB, ADO, RDS und ADO/ MD? <http://support.microsoft.com/kb/190463/DE/>. 1. Dezember 2006.
- Mischke, J. (2006): Druck mich: Von der Datenbank zum fertigen Bericht. iX, o. Jg. Heft 7, S. 112-116.
- Mohr, M. (2005): HCC-Einführungsschulung zum SAP Business Information Warehouse. Hallbergmoos bei München.
- Mucksch, H. (1999): Das Data Warehouse als Datenbasis analytische Informationssysteme: Architektur und Komponenten. In: Chamoni/Gluchowski (1999), S. 171-189.
- Mucksch, H.; Behme, W. (Hrsg.) (1998): Das Data Warehouse-Konzept: Architektur – Datenmodelle – Anwendungen. 3. Auflage. Wiesbaden.
- Mucksch, H.; Behme, W. (Hrsg.) (2000): Das Data Warehouse-Konzept: Architektur – Datenmodelle – Anwendungen. 4. Auflage. Wiesbaden.
- Mucksch, H.; Behme, W. (2000): Das Data Warehouse-Konzept als Basis einer unternehmensweiten Informationslogistik. In: Mucksch/Behme (2000), S. 3-80.
- North, K. (1998): Wissensorientierte Unternehmensführung: Wertschöpfung durch Wissen. Wiesbaden.
- Ortiz, C. (2005): OLAP-Workshop. Wolfsburg.
- Ortner, E. (1999): Repository Systems. Teil 1: Mehrstufigkeit und Entwicklungsumgebung. Informatik Spektrum, o. Jg., Volume 22, Number 4, S. 235-251.
- Österle, H. (2000): Geschäftsmodell des Informationszeitalters. In: Österle/Winter (2000), S. 21-42.
- Österle, H.; Winter, R. (Hrsg.) (2000): Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters. Berlin, u.a.
- Oestereich, B.; Hruschka, P.; Josuttis, N.; Kocher, H.; Krasemann, H.; Reinhold, M. (1999): Erfolgreich mit Objektorientierung – Vorgehensmodelle und

- Managementpraktiken für die objektorientierte Softwareentwicklung.
München, Oldenburg.
- Peck, G. (2003a): Crystal Reports professional. Bonn.
- Peck, G. (2003b): The Complete Reference: Crystal Reports 9. New York et al.
- Pendse, N. (2006): What is OLAP? - FASMI. <http://www.olapreport.com/fasmi.htm>.
20. September 2006.
- Pendse, N.; Creeth, R. (1995): The OLAP Report: Succeeding with On-Line Analytical Processing. Business Intelligence, o. Jg., Nr. 1, S. 50-55.
- Poe, V. (1996): Building a Data Warehouse for Decision Support. Upper Saddle River.
- Probst, G.; Raub, S.; Romhardt, K. (2003): Wissen managen. 4. Auflage.
Frankfurt/Main.
- Sapia, C.; Blaschka, M.; Höfling, G.; Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. Advances in Database Technologies Lecture Notes in Computer Science, o. Jg., Vol 1552, S. 105-116.
- Scott Morten, M. S. (1983): State of the Art in Research in Management Support Systems. Vortrag im Rahmen des Colloquiums on Informations Systems, MIT, Massachusetts.
- Schinzler, H. D.; Bange, C.; Wehner, J. (1997): Management mit Maus und Tastatur: Business Intelligence- und Data Warehouse-Werkzeuge. Wiesbaden.
- Schinzler, H. D.; Bange, C. (1999): Werkzeuge zum Aufbau analytischer Informationssysteme: Marktübersicht. In: Charmoni/Gluchowski (1999), S. 45-74.
- Schirp, G. (2001): Anforderungsanalyse im Data-Warehouse-Projekt: Ein Erfahrungsbericht aus der Praxis. In: Hildebrand (2001), S. 81-87.
- Seufert, A.; Schiefer, J. (2005): Enhanced Business Intelligence – Supporting Business Processes with Real-Time Business Analytics. In: Proceedings of the 16th International Workshop on Database and Expert Systems Applications. o. O.
- Seufert, A.; Lehmann, P. (2006): Business Intelligence – Status quo und zukünftige Entwicklungen. In: Heilmann et al. (2006): S.21-31.
- Volkswagen AG (Hrsg.) (2006): BI-Strategie des Volkswagen Konzerns. Wolfsburg.
März 2006.
- Von Maur, E.; Winter, R. (Hrsg.) (2002): Vom Data Warehouse zum Corporate Knowledge Center. Heidelberg.
- Von Maur, E.; Winter, R. (Hrsg.) (2003): Data Warehouse Management: Das St. Galler Konzept zur ganzheitlichen Gestaltung der Informationslogistik. Berlin, u.a.
- Watson, H. J.; Wixom, B. H.; Hoffer, J. A.; Anderson-Lehrman, R.; Reynolds, A. M. (2006): Real Time Business Intelligence: Best Practices At Continental Airlines. Information Systems Management Journal, o. Jg., Heft Winter, S. 7-18.
- Wedekind, H. (1997): Datenmodell. In: Mertens et al. (1997), S. 118-120.
- Zornes, A. (1994): Re-Engineering „Data Jailhouses“ into „Data Warehouses“. Next Generation Decision Support: Meta Group Inc., o. Jg., S. 17.

Abschließende Erklärung

Ich versichere hiermit, da ich die vorliegende Diplomarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, den 12. Dezember 2006