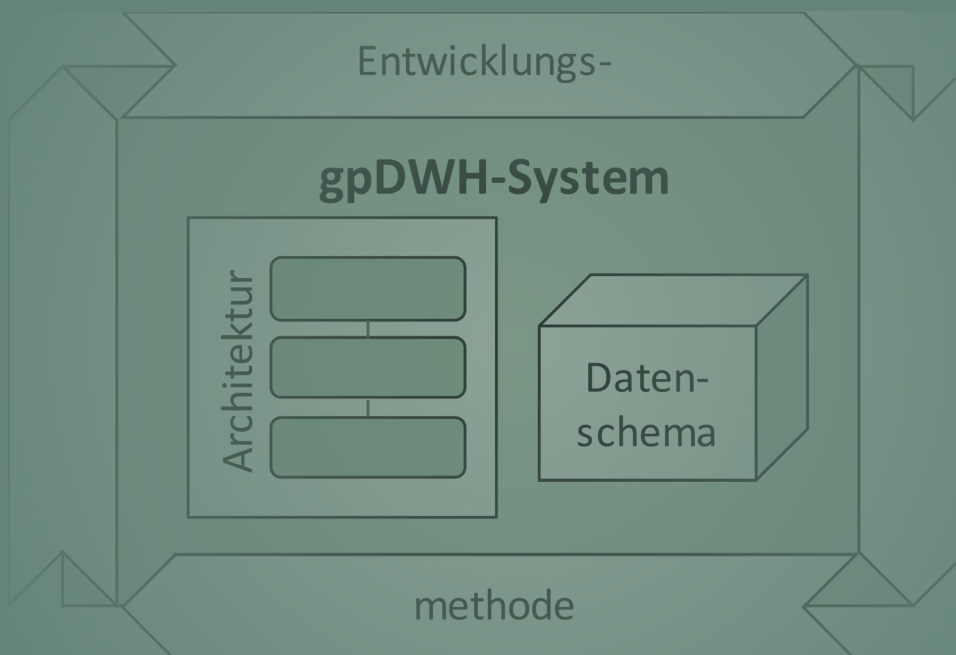


Konstruktion geschäftsprozessorientierter operativ-dispositiver Komponentenarchitekturen

Thomas Benker



University
of Bamberg
Press

27 Schriften aus der Fakultät Wirtschaftsinformatik
und Angewandte Informatik der Otto-Friedrich-
Universität Bamberg

Contributions of the Faculty Information Systems
and Applied Computer Sciences of the
Otto-Friedrich-University Bamberg

Schriften aus der Fakultät Wirtschaftsinformatik
und Angewandte Informatik der Otto-Friedrich-
Universität Bamberg

Contributions of the Faculty Information Systems
and Applied Computer Sciences of the
Otto-Friedrich-University Bamberg

Band 27

Konstruktion geschäftsprozessorientierter operativ-dispositiver Komponentenarchitekturen

Thomas Benker

Bibliographische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Informationen sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Diese Arbeit hat der Fakultät Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg als Dissertation vorgelegen.

1. Gutachter: Prof. Dr. Elmar J. Sinz

2. Gutachter: Prof. Dr. Sven Overhage

Tag der mündlichen Prüfung: 11.12.2017

Dieses Werk ist als freie Onlineversion über den Hochschulschriften-Server (OPUS; <http://www.opus-bayern.de/uni-bamberg/>) der Universitätsbibliothek Bamberg erreichbar. Kopien und Ausdrücke dürfen nur zum privaten und sonstigen eigenen Gebrauch angefertigt werden.

Herstellung und Druck: docupoint, Magdeburg

Umschlaggestaltung: University of Bamberg Press, Larissa Günther

© University of Bamberg Press Bamberg, 2018

<http://www.uni-bamberg.de/ubp/>

ISSN: 1867-7401

ISBN: 978-3-86309-557-4 (Druckausgabe)

eISBN: 978-3-86309-558-1 (Online-Ausgabe)

URN: urn:nbn:de:bvb:473-opus4-512300

DOI: <http://dx.doi.org/10.20378/irbo-51230>

Meiner Familie

Geleitwort

Gemäß dem klassischen Verständnis betrieblicher Anwendungssysteme stehen die Verarbeitungsprinzipien OLTP (online transaction processing) und OLAP (online analytical processing) „quer“ zueinander. Während OLTP an einzelnen Geschäftsprozessen orientiert ist, zielt OLAP auf die geschäftsprozessübergreifende Auswertung von Kennzahlen, die in der Regel historisiert sind und nach unterschiedlichen weiteren Dimensionen analysiert werden. Anders ausgedrückt, unterstützt OLTP die operativen Bereiche der Leistungserstellung eines Unternehmens, während OLAP auf die dispositiven Managementbereiche Bezug nimmt.

Die Nachteile dieses klassischen Ansatzes treten insbesondere mit zunehmendem Umfang des Gesamtsystems zutage. Die Zweiteilung in ein operatives und ein dispositives Teilsystem ist zu unflexibel, um auf Änderungsanforderungen im System zeitnah zu reagieren.

Thomas Benker greift in seiner Dissertation „Konstruktion geschäftsprozessorientierter operativ-dispositiver Komponentenarchitekturen“ dieses Problem auf. Er schlägt für betriebliche Anwendungssysteme eine Komponentenarchitektur vor, die operative und dispositive Elemente integriert. Hierfür entwickelt er eine zugehörige Konstruktionsmethodik und evaluiert Architektur und Konstruktionsmethodik anhand einer ausführlichen und praxisrelevanten Fallstudie.

Den Untersuchungsgegenstand der Arbeit bilden somit Architektur und Entwicklung prozessorientierter Data-Warehouse-Systeme. Zugehörige Untersuchungsziele sind: (1) die Architektur so zu gestalten, dass eine flexible Anpassbarkeit an Veränderungen der Prozessstruktur sowie die bedarfs- und zeitgerechte Unterstützung heterogener Nutzergruppen ermöglicht und (2) eine umfassende, abgestimmte Methodik zur Entwicklung dieser Architektur bereitgestellt wird.

Mit der vorliegenden Arbeit werden die Themenbereiche (a) der konzeptuellen Grundlagen betrieblicher Informationssysteme, (b) der Systementwicklung und (c) des Data-Warehousing zusammengeführt. Thomas Benkers Ansatz ist theoretisch wie methodisch fundiert und gleichzeitig praxistauglich. Die Lektüre der

Arbeit kann allen empfohlen werden, die sich in Wissenschaft und Praxis mit Fragen der ganzheitlichen, komponentenbasierten und prozessorientierten Architektur von Anwendungssystemen beschäftigen. Möge die Dissertation einen nachhaltigen Beitrag leisten, die entwicklungshistorisch bedingte Trennung zwischen OLTP und OLAP mit dem Ziel der Gewinnung flexibler Anwendungssystemarchitekturen zu überwinden.

Bamberg, im Januar 2018

Prof. Elmar J. Sinz

Vorwort

Geschäftsprozesse bilden die Lösungsverfahren betrieblicher Systeme und gleichzeitig die zentralen Artefakte zur Realisierung der unternehmerischen Ziele. In Zeiten dynamischer Märkte mit intensivem Wettbewerb ist es umso wichtiger, zeitnah auf Veränderungen am Markt sowie auf Ereignisse in den Prozessen selbst zu reagieren. In der Konsequenz gilt es, die eigenen Geschäftsprozesse stetig zu hinterfragen und Maßnahmen zu deren effizienterer und innovativerer Durchführung abzuleiten. Mit dem *geschäftsprozessorientierten Data-Warehouse-Konzept* als zentralem Baustein operativ-dispositiver Anwendungssysteme leistet die Arbeit einen Beitrag zur Realisierung dieser Zielsetzung.

Die Intention der effizienten und innovativen Durchführung lässt sich auch auf den Promotionsprozess übertragen. Hier ist jedoch das Umfeld mindestens genauso wichtig wie der Prozess selbst. Zunächst möchte ich mich bei meinem Doktorvater Herrn Prof. Dr. Elmar J. Sinz für seine Unterstützung von der ersten Idee bis hin zur Fertigstellung bedanken. Ein immer offenes Ohr und kontroverse, aber stets sachliche und v.a. konstruktive Diskussionen haben maßgeblich zum Gelingen der Arbeit beigetragen. Ebenso möchte ich den weiteren Mitgliedern der Promotionskommission, Herrn Prof. Overhage und Herrn Prof. Wirtz, für ihre Unterstützung herzlich danken.

Das kollegiale und konstruktive Umfeld des Lehrstuhls stellte ideale Rahmenbedingungen zur Promotion dar. Herzlich danken möchte ich allen Kollegen für die zahlreichen Beiträge in Doktorandenseminaren und persönlichen Gesprächen. Einige Weggefährten möchte ich aufgrund ihrer intensiven, freundschaftlichen Unterstützung hervorheben: Für Matthias Wolf waren das Teilen eigener Erfahrungen, die stetige Motivation und die fachliche Diskussion eine Selbstverständlichkeit. Mit stets kritischem Blick und vielen wertvollen Ratschlägen trug Carsten Jürck maßgebend zur Qualität der Arbeit bei. Mit Andree Teusch wusste ich den idealen Sparringspartner im Nachbarbüro. Armin Duske, Markus Plaha und Achim Ulbrich-vom Ende hatten immer ein offenes Ohr und sorgten nicht zuletzt für die technischen und organisatorischen Rahmenbedingungen zur Fertigstellung der Dissertation.

Zu guter Letzt möchte ich meiner Familie danken. Liebe Anja, lieber Lasse, ohne eure Liebe, eure Unterstützung, eure Kraft und eure Rückendeckung wäre dies alles nicht möglich gewesen. Gleiches möchte ich auch meinen Eltern Winfried und Brigitta sagen. Danke für die bedingungslose Unterstützung und die stete Motivation, *endlich* fertig zu werden.

Bamberg, im Januar 2018

Thomas Benker

Inhaltsverzeichnis

Fachliches Abkürzungsverzeichnis	XII
Abbildungsverzeichnis	XV
Tabellenverzeichnis	XVII
Verzeichnis der Code-Ausschnitte	XIX
1 Einführung	1
1.1 Motivation und Problemstellung der Arbeit	1
1.1.1 Motivation des Themengebietes	1
1.1.2 Einführung prozessorientierter operativ-dispositiver Anwendungssysteme	3
1.1.3 Explikation der Forschungslücke	4
1.2 Untersuchungsproblem der Arbeit	8
1.2.1 Untersuchungsgegenstand	8
1.2.2 Untersuchungsziel	8
1.3 Hypothesen der Arbeit	10
1.4 Lösungsstrategie und methodisches Vorgehen der Arbeit	12
1.4.1 Lösungsstrategie	12
1.4.2 Methodisches Vorgehen	13
1.5 Aufbau der Arbeit	16
2 Methodische Grundlagen zur Gestaltung betrieblicher Informationssysteme	19
2.1 Grundlagen betrieblicher Informationssysteme	19
2.1.1 Abgrenzung des betrieblichen Informationssystems	19
2.1.2 Einführung des betrieblichen Aufgabenmodells	22
2.1.3 Modellierung betrieblicher Informationssysteme	23
2.2 Gestaltung betrieblicher Informationssysteme auf Basis des Semantischen Objektmodells	26
2.2.1 Einordnung des Prozessbegriffs im Kontext des betrieblichen Informationssystems	26
2.2.2 Die Unternehmensarchitektur der SOM-Methodik	28
2.2.3 Das Vorgehensmodell der SOM-Methodik	30
2.3 Komponentenarchitekturen zur Automatisierung betrieblicher Informationssysteme	31
2.3.1 Begriffliche Grundlagen zu Komponentenarchitekturen	31
2.3.2 Grundlagen zur Spezifikation von Komponentenarchitekturen	34
3 Methodische Grundlagen zu prozessorientierten Data-Warehouse-Systemen	41
3.1 Grundlagen dispositiver Anwendungssysteme	41
3.1.1 Bestimmung des Begriffs des dispositiven Anwendungssystems	41

3.1.2	Einordnung von Data-Warehouse-Systemen als dispositive Anwendungssysteme	44
3.2	Architektur von Data-Warehouse-Systemen	48
3.2.1	Einführung der Architektur von Data-Warehouse-Systemen	48
3.2.2	Ein generisches Schichtenmodell zur Beschreibung der Architektur von Data-Warehouse-Systemen	50
3.3	Einführung des multidimensionalen Datenmodells	51
3.3.1	Struktur des multidimensionalen Datenmodells	52
3.3.2	Operatoren des multidimensionalen Datenmodells	57
3.3.3	Realisierungsformen des multidimensionalen Datenmodells	58
3.4	Modellierung multidimensionaler Datenschemata mit dem Semantischen Data-Warehouse-Modell	60
3.5	Grundlagen prozessorientierter Data-Warehouse-Systeme	62
3.5.1	Abgrenzung prozessorientierter Data-Warehouse-Systeme	63
3.5.2	Anwendungsszenarien prozessorientierter Data-Warehouse-Systeme	65
3.5.3	Latenzen im Kontext prozessorientierter Data-Warehouse-Systeme	67
4	Einführung der durchgehenden Fallstudie e-Car Net	69
4.1	Grundlagen der Geschäftsprozessmodellierung mit SOM	69
4.2	Einführung in die Fallstudie e-Car Net	72
4.3	Einführung in den Geschäftsprozess Mobility Provision	73
4.3.1	Struktursicht auf den Geschäftsprozess Mobility Provision	73
4.3.2	Verhaltenssicht auf den Geschäftsprozess Mobility Provision	75
4.4	Automatisierung des Mobility Provision-Vertriebsprozesses	77
5	Referenzarchitektur prozessorientierter operativ-dispositiver Systeme	79
5.1	Anforderungen an die Architektur operativ-dispositiver Anwendungssysteme	79
5.2	Stand der Literatur: Architekturen prozessorientierter Data-Warehouse-Systeme	82
5.2.1	Diskussion der Eignung klassischer Data-Warehouse-Architekturen	82
5.2.2	Diskussion verwandter Arbeiten zu pDWH-Architekturen	84
5.3	Konzeption einer komponentenorientierten Referenzarchitektur für operativ-dispositive Anwendungssysteme	90
5.3.1	Klassifikation als Referenzarchitektur	90
5.3.2	Konzeption der flexiblen Data-Mart-Architektur	91
5.3.3	Spezifikation von Komponenten für operativ-dispositive Systeme	96
5.3.4	Anwendung der Referenzarchitektur auf die Fallstudie Mobility Provision	99
5.4	Konzeption der Komponenten des prozessorientierten DWH-Systems aus Innensicht	103

5.4.1	Angebotsorientierte Datenbereitstellung durch operative Komponenten	103
5.4.2	Integration multidimensionaler Datenstrukturen	105
5.4.3	Nachfrageorientierte Datenbereitstellung in Data Marts	107
5.5	Integration prozess- und themenorientierter Data-Warehouse-Systeme .	108
5.6	Zusammenfassung und Einordnung der Ergebnisse	112
6	Datenstrukturen prozessorientierter Data-Warehouse-Systeme	115
6.1	Grundlagen der Business Process Model and Notation	115
6.1.1	Diagrammarten der Business Process Model and Notation . . .	116
6.1.2	Konformitätsklassen der Business Process Model and Notation	117
6.1.3	Das Metamodell der Business Process Model and Notation . . .	118
6.2	Anforderung an die Datenstrukturen prozessorientierter Data-Warehouse-Systeme	122
6.2.1	Anforderungen an das Datenmodell	122
6.2.2	Anforderungen an das Datenschema	123
6.3	Stand der Literatur: Datenstrukturen prozessorientierter Data-Warehouse-Systeme	127
6.4	Gestaltung eines multidimensionalen Prozessdatenschemas	133
6.4.1	Ableitung prozessorientierter Dimensionen auf Basis der BPMN	135
6.4.2	Spezifikation von prozessorientierten Kennzahlen	138
6.4.3	Anwendung des multidimensionalen Prozessmodells	139
6.4.4	Unterstützung der vollen BPMN-Konformität zur Prozessmodellierung	142
6.5	Bereitstellung prozessorientierter Daten durch die Komponentenarchitektur	144
6.6	Zusammenfassende Einordnung	148
7	Entwicklung geschäftsprozessorientierter operativ-dispositiver Systeme	151
7.1	Anforderungen an die Entwicklung geschäftsprozessorientierter operativ-dispositiver Systeme	151
7.2	Stand der Literatur: Entwicklung geschäftsprozessorientierter operativ-dispositiver Systeme	154
7.2.1	Geschäftsprozessorientierte Entwicklung komponentenbasierter Anwendungssysteme	155
7.2.2	Entwicklung geschäftsprozessorientierter Data-Warehouse-Systeme	157
7.3	Entwicklungsmethodik für geschäftsprozessorientierte operativ-dispositive Systeme	159
7.3.1	Grundlegende Konzeption der Entwicklungsmethodik	160
7.3.2	Methodik zur Entwicklung des operativen Teilsystems	161
7.3.3	Erweiterung der Methodik zur Entwicklung des prozessorientierten Data-Warehouse-Systems	163
7.4	Entwicklung des operativen Anwendungssystems	166
7.4.1	Spezifikation des fachlichen Anwendungsmodells	166

7.4.2	Spezifikation des komponentenorientierten Anwendungsmodells	173
7.5	Entwicklung des geschäftsprozessorientierten Data-Warehouse-Systems	181
7.5.1	Spezifikation des aufgabenträgerspezifischen Data-Warehouse-Schemas	181
7.5.2	Spezifikation des geschäftsprozessorientierten Data-Warehouse-Schemas	182
7.6	Entwicklung nachfrageorientierter Data Marts	191
7.6.1	Spezifikation von Data Marts für das strukturorientierte Szenario	192
7.6.2	Spezifikation von Data Marts für das verhaltensorientierte Szenario	194
7.7	Zusammenfassung und Diskussion der Ergebnisse	202
7.7.1	Zusammenfassung der Entwicklungsmethodik	202
7.7.2	Diskussion der Entwicklungsmethodik	203
8	Extension der Entwicklungsmethodik	207
8.1	Themenorientierte Positionierung der Ergebnisse	207
8.1.1	Positionierung im Kontext der Business Intelligence	207
8.1.2	Positionierung im Geschäftsprozess-Management	212
8.2	Abstraktion verwendeter Modellierungsansätze	213
8.2.1	Verallgemeinerung des prozessorientierten Data-Warehouse-Schemas	214
8.2.2	Verallgemeinerung der Entwicklungsmethodik	215
8.2.3	Data-Vault als Alternative zur Modellierung des prozessorientierten Data-Warehouse-Schemas	220
8.3	Abstraktion von der Anwendungsdomäne	223
8.3.1	Allgemeine Charakterisierung des strukturorientierten Szenarios	224
8.3.2	Allgemeine Charakterisierung des verhaltensorientierten Szenarios	228
8.3.3	Bewertung beispielhafter Geschäftsprozesse	230
9	Reflexion und Ausblick	235
9.1	Komponentenorientierung als Architekturparadigma für operativ-dispositive Anwendungssysteme	235
9.2	Die betriebliche Aufgabe als Konzept zur geschäftsprozessorientierten Einordnung von Workflow-Abläufen	237
9.3	Das multidimensionale Datenmodell als Grundlage zur nachfrageorientierten Datenbereitstellung	239
9.4	Aussagenkonsistenz durch hierarchische Integration prozess- und themenorientierter Data-Warehouse-Systeme	240
9.5	Das Geschäftsprozessmodell als Grundlage zur Erweiterung operativer Anwendungssysteme um dispositive Funktionen	242
9.6	Zusammenfassung	243

Anhang	245
A.1 Relationale Realisierung des prozessorientierten Data-Warehouse-Schemas	245
A.2 Implementierung aufgabenorientierter Sichten	247
A.3 Zerlegungsprotokoll zur Erweiterung der Fallstudie Mobility Provision .	249
Literaturverzeichnis	250

Fachliches Abkürzungsverzeichnis

ARIS	Architektur integrierter Informationssysteme
AT	Aufgabenträger
AwS	Anwendungssystem
BI	Business Intelligence
BPI	Business Process Intelligence
BPMN	Business Process Modell and Notation
DH	Dimensionshierarchiestufe
DSS	Decision Support System
DWH	Data-Warehouse
eEPK	erweiterte Ereignisgesteuerte Prozesskette
EIS	Executive Information System
EPK	Ereignisgesteuerte Prozesskette
ESS	Executive Support System
ETL	Extraktion-Transformation-Laden
EU	Entscheidungsunterstützung
EU-System	Entscheidungsunterstützungssystem
GDH	Gesamt-Dimensionshierarchiestufe
GP	Geschäftsprozess
gpDWH	geschäftsprozessorientiertes Data-Warehouse
GPM	Geschäftsprozess-Management
HOLAP	Hybrides OLAP
IAS	Interaktionsschema
KEP	Kurier-, Express- und Paketdienste
KOS	konzeptuelles Objektschema
KOT	konzeptueller Objekttyp
KPI	Key Performance Indikator
LOT	leistungsspezifischer Objekttyp
MDA	Model Driven Architecture
MIS	Management Information System
MOLAP	Multidimensionales OLAP
ODS	Operational Data Store
OMG	Object Management Group
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OOT	objektspezifischer Objekttyp

pDWH	prozessorientiertes Data-Warehouse
ROLAP	Relationales OLAP
SDH	Sub-Dimensionshierarchiestufe
SDWM	Semantisches Data-Warehouse-Modell
SERM	Strukturiertes Entity-Relationship-Modell
SOA	Serviceorientierte Architektur
SOM	Semantisches Objektmodell
SQL	Structured Query Language
TOT	transaktionsspezifischer Objekttyp
UML	Unified Modeling Language
VES	Vorgangs-Ereignis-Schema
VOS	Vorgangsobjektschema
VOT	Vorgangsobjekttyp
WfMC	Workflow Management Coalition
WS-BPEL	Web Services Business Process Execution Language

Abbildungsverzeichnis

Abb. 1.1:	Teilsysteme prozessorientierter operativ-dispositiver Anwendungssysteme	4
Abb. 1.2:	Schematische Darstellung des Untersuchungsgegenstandes	9
Abb. 1.3:	Methodisches Vorgehen zur Lösung des Untersuchungsproblems	14
Abb. 1.4:	Aufbau der Arbeit	16
Abb. 2.1:	Abgrenzung des Informationssystems im betrieblichen Objektsystem	21
Abb. 2.2:	Struktur der betrieblichen Aufgabe	22
Abb. 2.3:	Generischer Architekturrahmen	24
Abb. 2.4:	Zusammenhang von Aufgaben- und Aufgabenträgerebene	25
Abb. 2.5:	Unternehmensarchitektur der SOM-Methodik	28
Abb. 2.6:	Vorgehensmodell der SOM-Methodik	30
Abb. 2.7:	Idealtypische Entwicklung komponentenorientierter Anwendungssysteme	36
Abb. 2.8:	Auswirkungen der Faktoren Kohäsion und Kopplung	38
Abb. 2.9:	Spezifikation von Komponenten mit der UML	39
Abb. 3.1:	Integrierte Informationsverarbeitung	44
Abb. 3.2:	Regelkreisorientierte Einordnung des Data-Warehouse-Systems	46
Abb. 3.3:	Idealtypische Architektur eines Data-Warehouse-Systems	49
Abb. 3.4:	Generische Schichtenarchitektur für Data-Warehouse-Systeme	50
Abb. 3.5:	Beispiel einer Dimension mit parallelen Hierarchien	54
Abb. 3.6:	Beispiel einer Dimension mit unvollständiger Hierarchie	55
Abb. 3.7:	Beispiel einer Dimension mit rekursiver Strukturen	56
Abb. 3.8:	Metamodell der Dimensionssicht des Semantischen Data-Warehouse-Modells	60
Abb. 3.9:	Metamodell der Sicht auf Kennzahlensysteme des Semantischen Data-Warehouse-Modells	62
Abb. 3.10:	Beispielhafte Anwendung des Semantischen Data-Warehouse-Modells im universitären Kontext	63
Abb. 3.11:	Schematische Abgrenzung themen- und prozessorientierter Data-Warehouse-Systeme	64
Abb. 3.12:	Regelkreisorientierte Einordnung eines prozessorientierten Data-Warehouse-Systems	66
Abb. 3.13:	Zusammenhang von Wertbeitrag und Latenz der Reaktion auf Prozessereignisse	68
Abb. 4.1:	Metamodell und Sichten für SOM-Geschäftsprozessmodelle	70
Abb. 4.2:	Ersetzungsregeln für SOM-Geschäftsprozessmodelle	71
Abb. 4.3:	Geschäftsprozessmodell der Fallstudie <i>e-Car Net</i>	72
Abb. 4.4:	Protokoll der Transaktions- und Objektzerlegung zum Szenario <i>Mobility Provision</i>	74
Abb. 4.5:	Interaktionsschema zum Szenario <i>Mobility Provision</i>	75

Abb. 4.6:	Vorgangs-Ereignis-Schema zur Buchung im Szenario <i>Mobility Provision</i>	75
Abb. 4.7:	Vorgangs-Ereignis-Schema zur Buchungsbestätigung im Szenario <i>Mobility Provision</i>	76
Abb. 4.8:	Vorgangs-Ereignis-Schema zur Leistungserbringung im Szenario <i>Mobility Provision</i>	76
Abb. 5.1:	Referenzarchitektur zur flexiblen Spezifikation von Data Marts	92
Abb. 5.2:	Referenzarchitektur operativ-dispositiver Anwendungssysteme	97
Abb. 5.3:	Komponentenarchitektur zur Fallstudie <i>Mobility Provision</i>	100
Abb. 5.4:	Komponenteninterne, angebotsorientierte Datenbereitstellung	104
Abb. 5.5:	Komponentenübergreifende Datenintegration	106
Abb. 5.6:	Angebotsorientierte Datenbereitstellung zur Integration themen- und prozessorientierter Data-Warehouse-Systeme	110
Abb. 5.7:	Nachfrageorientierte Datenbereitstellung zur Integration themen- und prozessorientierter Data-Warehouse-Systeme	111
Abb. 5.8:	Kartierung der Integration themen- und prozessorientierter Data-Warehouse-Systeme in der Architektur des Analyseschemas	111
Abb. 6.1:	Kollaborationsdiagramm (public process) zur Auftragsverwaltung	117
Abb. 6.2:	Metamodell der Business Process Model and Notation	119
Abb. 6.3:	Grafische Syntax der Business Process Model and Notation	119
Abb. 6.4:	Ausschnitt des BPMN-Prozesses zum Vertriebskanal der Fallstudie <i>Mobility Provision</i>	134
Abb. 6.5:	Dimensionen für BPMN-Workflows	135
Abb. 6.6:	Exemplarische Instanzen des pDWH-Schemas zum Vertriebsprozess	140
Abb. 6.7:	Operative und multidimensionale Datenstrukturen der Komponente <i>Vertriebsdaten</i>	145
Abb. 6.8:	Operative und multidimensionale Datenstrukturen der Komponente <i>Kundendaten</i>	146
Abb. 7.1:	Modellarchitektur zur Entwicklung des operativen Systems	162
Abb. 7.2:	Modellarchitektur zur Entwicklung der angebotsorientierten Datenbereitstellung	164
Abb. 7.3:	Modellarchitektur zur Entwicklung der nachfrageorientierten Datenbereitstellung	166
Abb. 7.4:	Metamodell für SOM-Anwendungsmodelle	167
Abb. 7.5:	Konsolidiertes konzeptuelles Objektschema zur Fallstudie <i>Mobility Provision</i>	169
Abb. 7.6:	Initiales BPMN-Kollaborationsdiagramm zum Vertriebsprozess	172
Abb. 7.7:	Komponentenabgrenzung im KOS der Fallstudie <i>Mobility Provision</i>	176
Abb. 7.8:	Komponentenabgrenzung im Zerlegungsbaum der Fallstudie <i>Mobility Provision</i>	177
Abb. 7.9:	Konsolidiertes BPMN-Kollaborationsdiagramm des Vertriebskanals	179
Abb. 7.10:	Innensicht des Sub-Prozesses <i>Empfang Verfügbarkeitsliste</i>	181
Abb. 7.11:	Aufgabenorientierte Sicht – Durchschnittliche Laufzeit	186
Abb. 7.12:	Aufgabenorientierte Sicht – Anteil der Nachereignisse	187

Abb. 7.13:	Verhaltensorientierte Einordnung aufgabenorientierter Sichten	189
Abb. 7.14:	Strukturorientierte Einordnung aufgabenorientierter Sichten	191
Abb. 7.15:	Überlagerung von OpDW-Kreislauf und Referenzarchitektur	195
Abb. 7.16:	Ausschnitt des VES zur Angebotserweiterung	198
Abb. 7.17:	Beispiel zur Entscheidungsunterstützung als Prozessbaustein	200
Abb. 8.1:	Ausbaustufen operativer Business-Intelligence-Lösungen	210
Abb. 8.2:	Einordnung der pDWH-Szenarien im Prozesslebenszyklus	213
Abb. 8.3:	Sprachunabhängige Dimensionen des pDWH-Schemas	214
Abb. 8.4:	Das ARIS-Haus	218
Abb. 8.5:	Einordnung von Data-Vault in die flexible Data-Mart-Architektur . .	222
Abb. 8.6:	Allgemeine Charakterisierung des strukturorientierten pDWH- Szenarios	224
Abb. 8.7:	Allgemeine Charakterisierung des verhaltensorientierten pDWH- Szenarios	229
Abb. 8.8:	Charakterisierung des Vertriebsprozesses der Fallstudie <i>Mobility Pro- vision</i>	230
Abb. 8.9:	Charakterisierung des Immatrikulationsprozesses an Hochschulen .	231
Abb. 8.10:	Charakterisierung des Logistikprozesses von KEP-Dienstleistern . .	232
Abb. 8.11:	Charakterisierung des Dienstleistungsprozesses von Call Centern .	233
Abb. A.1:	Zerlegungsprotokoll der erweiterten Fallstudie <i>Mobility Provision</i> . .	249

Tabellenverzeichnis

Tab. 3.1:	Charakterisierung operativer und dispositiver Anwendungssysteme .	42
Tab. 3.2:	Anwendungsszenarien prozessorientierter Data-Warehouse-Systeme	65
Tab. 4.1:	Automatisierungsgrade der Aufgaben von Vertriebskanal und Auftrags- verwaltung	77
Tab. 5.1:	Berücksichtigte Anforderungen und pDWH-Szenarien in verwandten Arbeiten	84
Tab. 6.1:	Berücksichtigte Dimensionen und Prozessperspektiven in verwand- ten Arbeiten	130
Tab. 7.1:	Transformationsregeln zur Ableitung des konzeptuellen Objektsche- mas aus SOM-Geschäftsprozessmodellen	168
Tab. 7.2:	Transformations- und Konsolidierungsschritte der Fallstudie <i>Mobility Provision</i>	170
Tab. 7.3:	Transformationsregeln zur Ableitung von BPMN-Workflowschemata aus SOM-Geschäftsprozessmodellen	171
Tab. 7.4:	Kriterien zur Komponentenabgrenzung in objektorientierten Strukturen	175
Tab. 7.5:	Merkmale der betrieblichen Aufgabe >D: <i>Verfügbarkeitsliste</i>	184
Tab. 7.6:	Abgrenzung der Data-Mart-Nutzung im verhaltensorientierten Szenario	196

Verzeichnis der Code-Ausschnitte

Code 5.1:	Export-Schnittstelle der Komponente <i>Vertriebsdaten</i>	101
Code 5.2:	Export-Schnittstelle der Komponente <i>Buchungsdaten</i>	101
Code 5.3:	Schnittstelle der Datenintegration zum Szenario <i>Vertriebsgestaltung</i> .	102
Code 5.4:	Schnittstelle der Entscheidungskomponente zur individuellen Angebotsenerweiterung	103
Code A.1:	SQL-Implementierung prozessspezifischer Dimensionstabellen . .	245
Code A.2:	SQL-Implementierung domänenspezifischer Dimensionstabellen .	246
Code A.3:	SQL-Implementierung der Faktabelle	246
Code A.4:	SQL-Implementierung der aufgabenorient. Sicht <i>Zeiteffizienz</i>	247
Code A.5:	SQL-Implementierung der aufgabenorient. Sicht <i>Sachzielerreichung</i>	248

1 Einführung

Geschäftsprozesse (GP) bilden die Lösungsverfahren für Unternehmen zur Realisierung ihrer Ziele. Operative und dispositive Aufgaben zur Durchführung und Gestaltung dieser Lösungsverfahren bilden den fachlichen Kontext der Arbeit. Den zentralen Untersuchungsgegenstand stellen Anwendungssysteme (AwS) dar, die die Realisierung und die Integration dieser Aufgaben unterstützen und im Rahmen der Arbeit als *Geschäftsprozessorientierte operativ-dispositive Anwendungssysteme* bezeichnet werden. Nachfolgend wird der zugehörige thematische Rahmen zunächst motiviert (Abschn. 1.1) und, anhand der identifizierten Forschungslücke, das Untersuchungsproblem (Abschn. 1.2) abgeleitet. Dieses bildet die Grundlage zur Spezifikation von Hypothesen (Kap. 1.3) sowie von Lösungsstrategie und -methode (Abschn. 1.4). In Abschn. 1.5 wird der Aufbau der Arbeit mit Bezug zur Problemstellung eingeführt.

1.1 Motivation und Problemstellung der Arbeit

Den Einstieg bildet eine allgemeine Hinführung zum Thema der Arbeit. In Abschn. 1.1.1 werden die Relevanz von Geschäftsprozessen, deren stetige Analyse und Gestaltung sowie AwS zur Unterstützung operativer und dispositiver Aufgaben motiviert. In Abschn. 1.1.2 wird darauf aufbauend das Konzept des operativ-dispositiven Anwendungssystems eingeführt. Ausgehend von diesem Systemverständnis wird in Abschn. 1.1.3 die bestehende Forschungslücke des Themenkomplexes aufgezeigt.

1.1.1 Motivation des Themengebietes

In Zeiten dynamischer Märkte mit intensivem Wettbewerb stehen Unternehmen vor der Herausforderung, rechtzeitig und angemessen auf Veränderungen im Markt- und Kundenverhalten zu reagieren (Dorn et al. 2010). Neben der ständigen Überprüfung der Positionierung eines Unternehmens aus Außensicht, also seiner direkten Beziehungen zu Geschäftspartnern sowie der globalen Positionierung im Markt, rückt auch zusehends die Bewertung der Innensicht der Unternehmensaufgabe mit deren *effizienter und innovativer Durchführung* in den Fokus (Becker &

Kahn 2012, S. 1f.). GPs bilden das Lösungsverfahren zur Realisierung des Unternehmensplans und stellen die Innensicht der Unternehmensaufgabe dar (Ferstl & Sinz 2013, S. 195f.). Als Haupt- oder Serviceprozesse beeinflussen sie direkt bzw. indirekt die betriebliche Leistungserstellung (Ferstl & Sinz 2013, S. 196f.). Die effiziente und innovative Durchführung der Geschäftsprozesse wird aus organisatorischer Sicht in erster Linie durch den Wechsel von einer funktions- hin zu einer prozessorientierten Unternehmensorganisation (z.B. Gaitanides 1983, Porter 1989, Davenport 1993) sowie der damit einhergehenden Reduktion der Schnittstellen zwischen Funktionsbereichen gestärkt (Becker & Kahn 2012, S. 3). Zur Steigerung von Effizienz und Innovationsgrad der Prozesse ist aber auch deren fortwährende Analyse notwendig. Hierzu ist mit dem Übergang zur prozessorientierten Unternehmensorganisation auch die Prozessorientierung in Konzepten der Business Intelligence (BI) zu berücksichtigen (Bucher & Dinter 2008b). Zur fortwährenden Neuausrichtung von Geschäftsprozessen an sich verändernde Bedingungen des Unternehmensumfeldes (Kunden-, Marktverhalten) sowie zur Bewertung von Prozessqualität und Performance schlagen Becker & Chamoni (2008) die Unterstützung durch *Analytische Informationssysteme* vor. Etablierte Konzepte aus dem Data-Warehouse (DWH) bzw. BI-Umfeld können bei der Analyse der Ist-Situation sowie bei der Ableitung prozessbezogener Reaktionen unterstützen. Durch die Berücksichtigung einer Prozessperspektive in DWH-Systemen wird ein tieferes Verständnis der Prozesse, ihres Verhaltens und ihrer Schwankungen sowie das frühzeitige Gegensteuern ermöglicht (Becker & Chamoni 2008). Aus *Compliance*-Sicht wird zudem die Einhaltung regulatorischer Rahmenbedingungen gefördert (zur Muehlen & Shapiro 2010). Prozessbezogene Reaktionen können vom steuernden Eingreifen in laufende Prozessinstanzen bis hin zur mittelfristigen Prozessgestaltung reichen. Sowohl Modelle als auch laufende Instanzen der Geschäftsprozesse liefern den Kontext für eine qualitativ hochwertige Datenanalyse und sind in einem prozessorientierten BI-Ansatz zu berücksichtigen (Marjanovic 2007). Ohne diese Prozess-Assoziation würden zahlreiche Informationen nur außerhalb ihres Interpretationskontextes oder ggf. gar nicht Gegenstand von Analysen werden (Bucher et al. 2009).

1.1.2 Einführung prozessorientierter operativ-dispositiver Anwendungssysteme

Zur Bereitstellung entscheidungsrelevanter Daten zu Prozessstrukturen und Verhalten wird das Konzept des DWH-Systems vorgeschlagen, für das sich der Begriff *prozessorientiertes Data-Warehouse-System* (pDWH-System) etabliert hat (z.B. Becker & Chamoni 2008, Bange & Fuchs 2011, Casati et al. 2007). Ein prozessorientiertes Data-Warehouse-System übernimmt mit der Datenbereitstellung eine Basisfunktionalität bei der Analyse von Prozessen sowie deren effizienter und innovativer Gestaltung. Diese zentrale Rolle nimmt das pDWH-Konzept auch in der vorliegenden Arbeit ein. Die Betrachtung eines pDWH-Systems erfolgt jedoch nicht isoliert, sondern integriert mit Quellsystemen sowie nutzenden Personen und AwS. Die enge Integration mit operativen Systemen (als Quellsystem sowie als automatisierter pDWH-Nutzer) wird u.a. bei Russom (2011) gefordert, um ohne großen Zeitverzug Daten für prozessbezogene Analysen bereitzustellen und um mögliche Reaktionen in laufenden Prozessinstanzen berücksichtigen zu können. Diese ganzheitliche Betrachtung manifestiert sich im Begriff des *operativ-dispositiven Anwendungssystems*, das schematisch in Abb. 1.1 dargestellt ist. Auf Basis der Begriffsabgrenzung von Kemper et al. (2010, S. 15) erfolgt die Bildung von Teilsystemen. Das *operative Teilsystem* ist auf die Unterstützung der Abwicklung von Geschäftsprozessen ausgerichtet. Das *dispositive Teilsystem* übernimmt die Vorbereitung von Entscheidungen durch das pDWH-System sowie die Entscheidungsunterstützung (EU) selbst. Entscheidungen können sich dabei direkt oder indirekt auf die operative Prozessausführung auswirken. Geschäftsprozessen kommen in diesem Kontext zwei Funktionen zu: Zum einen dienen beendete und laufende Instanzen als Informationsquelle über Ereignisse im Prozess-, Markt- oder Kundenverhalten. Als Beispiel können Verzögerungen in Vertriebsprozessen genannt werden. Zum anderen stellen die Prozesse auch das Ziel entsprechender Reaktionen auf Instanz- oder Schemaebene dar.

Zur effizienteren und innovativeren Durchführung von Geschäftsprozessen können zwei grundsätzliche Szenarien zur Nutzung von pDWH-Systemen abgegrenzt werden: Bei Bucher & Dinter (2008a) und Bucher et al. (2009) wird die Nutzung analytischer Funktionen, realisiert auf Basis von pDWH-Systemen, zur aktiven Steuerung des *Verhaltens* von Prozessinstanzen vorgeschlagen. Das Entscheidungs-

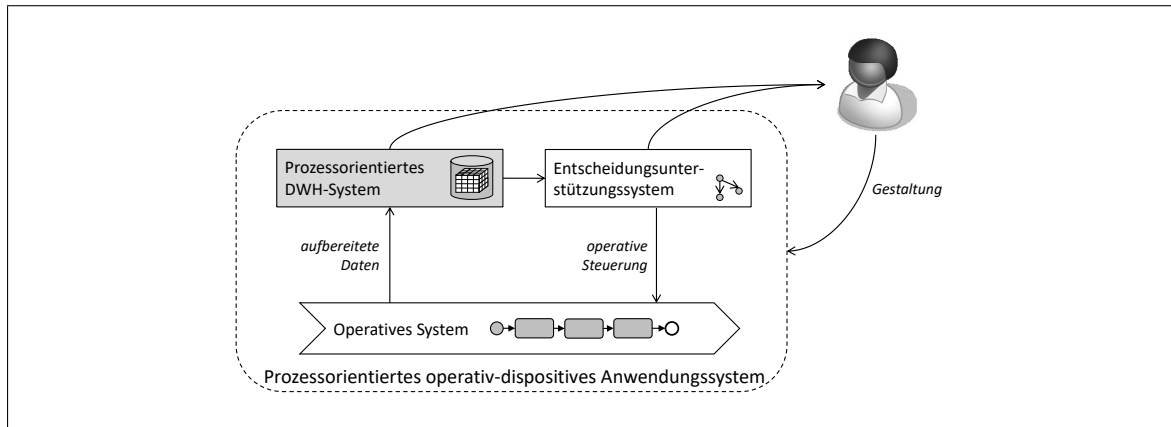


Abb. 1.1: Teilsysteme prozessorient. operativ-dispositiver Anwendungssysteme

unterstützungssystem (EU-System) kann dabei sowohl voll- als auch teilautomatisiert realisiert sein. Als übergeordnete Zielsetzung wird die Förderung der Prozessorientierung eines Unternehmens mit der Steigerung des Prozessverständnisses und der Identifikation von Kernkompetenzen genannt. Positive Auswirkungen werden zudem hinsichtlich der Prozesslaufzeiten, der Prozesskosten, der Qualität der Prozessleistung, der Prozessrisiken sowie der Ressourcennutzung erwartet (Bucher & Dinter 2008b). Ein zweites Szenario ist auf die mittelfristige Anpassung (Gestaltung) der *Prozessstruktur* sowie der Rahmenbedingungen (Ressourcen und Aufgabenträger) zur effizienteren und innovativeren Durchführung ausgerichtet (Felden et al. 2010). Für pDWH-Systeme leitet sich, auch auf Basis der beiden Szenarien, eine Vielzahl potenzieller Nutzer auf operativer, taktischer und strategischer Ebene ab (z.B. Eckerson 2007, White 2006, Hrach & Alt 2012). Für jede Nutzergruppe ist von individuellen Anforderungen an das pDWH-System (z.B. bzgl. Informationsbedarf oder Latenz) auszugehen.

1.1.3 Explikation der Forschungslücke

Auf Basis des kommunizierten Verständnisses eines geschäftsprozessorientierten operativ-dispositiven AwS kann sowohl hinsichtlich der AwS-Architektur als auch bzgl. der methodisch geleiteten Entwicklung ein weiterer Forschungsbedarf identifiziert werden. Dieser wird nachfolgend aufgezeigt. Die Ausführungen stellen jeweils eine Zusammenfassung der detaillierten Literaturanalyse zu Architektur (Kap. 5.2), Prozessperspektive bzw. Datenmodell (Kap. 6.3) sowie Entwicklung (Kap. 7.2) dar.

Lücke bzgl. der Architektur prozessorientierter operativ-dispositiver Anwendungssysteme

Das *Serviceorientierte Paradigma*, wie es auch zur Realisierung operativer AwS Anwendung findet, wird bei Bucher & Dinter (2008a), Bucher et al. (2009) oder auch Panian (2008) motiviert, um DWH-Anwendungen mit Analysefokus auf dem zugrundeliegenden Prozess zu realisieren. Als Vorteile gegenüber etablierten Architekturformen unternehmensweiter DWH-Systeme werden dabei v.a. die flexible Anpassbarkeit an sich dynamisch verändernde Anforderungen sowie die flexible Reaktionsfähigkeit genannt. Über das Konzept der *Services* kann zudem die Integration analytischer Funktionen in den operativen Betrieb erfolgen. Zahlreiche Arbeiten (z.B. Bucher et al. 2009, Markarian et al. 2007, Nguyen et al. 2005, Oberhofer & Nijkamp 2009, Pospiech & Felden 2013, Vogt et al. 2008) greifen dieses Funktionsparadigma auf, betrachten aber im Wesentlichen die Steuerung des Verhaltens operativer Prozesse durch die Identifikation und die Nutzung analytischer *Services* (Funktionen) zur *Laufzeit*. Die Realisierung der Dienste findet i.d.R. keine Berücksichtigung. Die mittelfristige effizientere und innovativere Gestaltung der Prozesse wird meist als Anwendung etablierter DWH-Konzepte verstanden, die um eine Prozessperspektive zu erweitern sind (z.B. zur Muehlen 2001). Dies erfolgt hauptsächlich durch die Definition entsprechender Datenstrukturen. Bei Schiefer et al. (2004) und Casati et al. (2007) werden zudem Erweiterungen der DWH-Komponenten zum Laden, Verwalten und Verarbeiten von Prozessdaten eingeführt. Die Betrachtung der Flexibilität von DWH-Architekturen zur Realisierung der genannten Szenarien kommt in verwandten Ansätzen generell zu kurz.

Bestehende serviceorientierte Ansätze bieten eine solide und geeignete Ausgangsbasis bzgl. der Identifikation und der Integration von dispositiven Diensten in operative AwS. Die Realisierung der Dienste sowie die Strukturierung eines DWH-Systems anhand des zugehörigen Funktionsparadigmas werden nicht hinreichend betrachtet. Dies gilt sowohl für die Interaktion innerhalb des DWH-Systems als auch mit dem operativen und dem *EU-System*. Ein ganzheitlicher Architekturansatz wirkt sich zudem positiv auf die Anwendbarkeit von pDWH-Systemen im betrieblichem Kontext aus und trägt zur Beherrschung der damit verbundenen Komplexität bei. Ebenfalls erscheint die Unterstützung beider Nutzungsszenarien (aktive Verhaltenssteuerung und Gestaltung der Prozessstruktur)

durch ein gemeinsames pDWH-System sinnvoll. Verwandte Architekturkonzepte sind i.d.R. jedoch nur auf eines der Szenarien ausgerichtet.

Lücke bzgl. der Prozessperspektive in prozessorientierten DWH-Systemen

Die Bereitstellung einer Prozessperspektive durch ein pDWH-System ist im Wesentlichen durch die Berücksichtigung von Prozessmerkmalen in den Datenstrukturen beeinflusst. Ansätze hierzu verwenden den Begriff *Prozess* softwarenahe im Sinne von *Workflow*. Prozessausführende *Workflow Management Systeme* dienen dabei als Quellsysteme für das pDWH-System. Der Begriff *Workflow* wird jedoch heterogen abgegrenzt. Einige Arbeiten verwenden ein informales, andere ein proprietäres Begriffsverständnis, z.T. ohne grafische Notation (z.B. Eder et al. 2002, Grigoria et al. 2004, Pau et al. 2007). Durch den informalen Begriff wird die Nachvollziehbarkeit des multidimensionalen Datenmodells, insbes. die Begründung von Dimensionen und Kennzahlen, eingeschränkt. Die Verwendung proprietärer *Workflow-Modelle* erschwert die Anwendbarkeit der abgeleiteten Prozessperspektive. Andere Arbeiten (z.B. Mansmann et al. 2007b, Mansmann et al. 2007a, Kueng et al. 2001) beschreiben Datenstrukturen zur Prozessanalyse ausgerichtet an den spezifischen Anforderungen einer fachlichen Domäne. Generische Datenstrukturen (z.B. Casati et al. 2007, Sun 2014) abstrahieren zwar von dieser, jedoch i.d.R. auch von der *Workflow-Sprache*. Alle Ansätze sind insbesondere gegenüber aktuellen *Workflow-Sprachen*, wie z.B. der *Business Process Model and Notation (BPMN)* (OMG 2011), eingeschränkt. Die Lücke lässt sich als Fehlen eines multidimensionalen Datenmodells beschreiben, das zum einen generisch bzgl. der Anwendungsdomäne und zum anderen mächtig genug bzgl. aktueller *Workflow-Konzepte* ist.

Abzugrenzen von der softwarenahen Prozessinterpretation ist der eingangs verwendete und dem *Workflow* übergeordnete Begriff des *Geschäftsprozesses*. Dieser beschreibt, ausgerichtet an Unternehmenszielen, die fachliche Ausgestaltung der betrieblichen Leistungserstellung (Ferstl & Sinz 2013, S. 196f.) und stellt die Grundlage zur Ableitung softwarenaher Prozessbeschreibungen (*Workflows*) dar. Fachliche Maßnahmen zur effizienteren und innovativeren Gestaltung werden primär anhand der Geschäftsprozesse durchgeführt (Becker & Kahn 2012, S. 1ff.) und darauf folgend auf die zugehörigen *Workflows* zur Realisierung abgebildet. Die dazu notwendige fachliche Interpretation von Prozessdaten sowie deren Ein-

ordnung in den Kontext der Unternehmensaufgabe wird jedoch nur eingeschränkt thematisiert. Lediglich bei Becker (2007) wird die Geschäftsprozess-Perspektive über ein multidimensionales Datenmodell für die *erweiterte Ereignisgesteuerte Prozesskette* (eEPK) berücksichtigt, jedoch ohne eine Einordnung in den übergeordneten Kontext der betrieblichen Leistungserstellung vorzunehmen. Ebenso wenig werden Workflows auf technischer Ebene betrachtet, sodass bei Analysen kein Wechsel von fachlicher zu technischer Perspektive möglich ist.

Lücke bzgl. der Entwicklung geschäftsprozessorientierter operativ-dispositiver Anwendungssysteme

Die methodisch geleitete Entwicklung geschäftsprozessorientierter operativ-dispositiver AwS mit einem pDWH-System als zentralem Teilsystem bildet einen weiteren Aspekt der bestehenden Lücke. Die konsistente Entwicklung der einzelnen Teilsysteme sowie die Integration dieser stellen komplexe Aufgaben dar. Der Entwickler ist in seiner Tätigkeit zur Beherrschung dieser Komplexität zu unterstützen. Lediglich der Ansatz nach Becker (2007) thematisiert neben Architektur und Datenmodell auch das Vorgehen zu Gestaltung und Nutzung von pDWH-Systemen. Das ganzheitliche Konzept wird als *modellbasierter closed-loop-Ansatz* für das GP-Controlling charakterisiert. Er rückt Strategie sowie Unternehmens- und Prozessmodelle der ARIS-Methodik (*Architektur integrierter Informationssysteme* nach Scheer (2001, 2002)) in den Fokus. Diese Modelle dienen sowohl als Ausgangspunkt zur Spezifikation von operativen AwS als auch des pDWH-Systems (ETL-Prozess und DWH-Schema), sind rückgekoppelt aber selbst Gegenstand von Anpassungen aufgrund DWH-basierter Analysen. Zur Unterstützung des *Prozess-Controllings* berücksichtigt der Ansatz neben Anpassung der Prozessstrukturen auch verhaltenssteuernde Maßnahmen in Prozessinstanzen, z.B. ausgelöst durch Mail-Benachrichtigungen. Die Architektur wird, unabhängig von einem Architekturparadigma, als prozessorientierte Erweiterung etablierter DWH-Architekturen beschrieben. Das Entwicklungsvorgehen orientiert sich aufgrund dessen am idealtypischen Prozess zur DWH-Entwicklung, erweitert um Beziehungen zur Konsistenzprüfung mit dem parallel verlaufenden Prozess des *Geschäftsprozess-Managements*. Die Entwicklung von DWH-Systemen mit Berücksichtigung des komponenten- bzw. serviceorientierten Paradigmas wird in der Literatur nicht thematisiert.

1.2 Untersuchungsproblem der Arbeit

Anhand der Forschungslücke wird das Untersuchungsproblem abgeleitet. Dieses setzt sich aus Untersuchungsgegenstand (Abschn. 1.2.1) und Untersuchungsziel (Abschn. 1.2.2) zusammen. Betrachtet wird die spezifische Form des Konstruktionsproblems. Zielsetzung dabei ist die Spezifikation einer Struktur zur Realisierung des geforderten Systemverhaltens (Sinz 2010).

1.2.1 Untersuchungsgegenstand

Das in Abschn. 1.1.2 eingeführte *prozessorientierte operativ-dispositive AwS* bildet den Untersuchungsgegenstand der Arbeit mit dem zentralen Artefakt des *prozessorientierten DWH-Systems*. Der Aufgabenbereich des pDWH-Systems umfasst Extraktion, Aufbereitung und Bereitstellung entscheidungsrelevanter Prozessdaten. Wie in Abschn. 1.1.3 deutlich wird, besteht v.a. bzgl. der *Architektur*, der *Prozessperspektive* sowie der *Entwicklung* von pDWH-Systemen und operativ-dispositiven AwS weiterer Forschungsbedarf. Diese bilden Teilaspekte des Untersuchungsgegenstandes und werden nachfolgend detaillierter vorgestellt:

1. **Architektur:** Die Architektur von prozessorientierten operativ-dispositiven Anwendungssystemen stellt einen Teilaspekt dar. Das prozessorientierte DWH-System bildet hier den zentralen Architekturbaustein. Die Betrachtung erfolgt sowohl aus Außen- als auch aus Innensicht anhand seiner Funktionen und Datenstrukturen zur Bereitstellung der Prozessperspektive. Das operative AwS sowie das EU-System werden hingegen nur aus Außensicht über die Integration mit dem pDWH-System betrachtet.
2. **Entwicklung:** Aufbauend auf einer Referenzarchitektur für prozessorientierte operativ-dispositive AwS (siehe Punkt 1) bildet die Entwicklung dieser Systeme auf Basis von Geschäftsprozessmodellen einen weiteren Bestandteil des Untersuchungsobjektes. Auch hier stellt das pDWH-System das zentrale Entwicklungsartefakt dar.

1.2.2 Untersuchungsziel

Als übergeordnete Zielsetzung wird die Unterstützung der *innovativeren und effizienteren Durchführung von Geschäftsprozessen* (siehe Abschn. 1.1) durch Bereitstellung von Referenzarchitektur und Entwicklungsmethodik für operativ-dispositive

AwS aufgestellt. Als Anwendungsszenarien sind dabei sowohl die Nutzung analytischer Informationen zur Steuerung laufender Prozessinstanzen als auch die mittelfristige Gestaltung der Prozessstruktur zu unterstützen. Abb. 1.2 zeigt schematisch das Untersuchungsproblem mit Fokus auf dem zentralen Artefakt des pDWH-Systems. Nachfolgend wird das Untersuchungsziel anhand von Referenzarchitektur und Entwicklungsmethodik weiter präzisiert:

1. Zielsetzung der Referenzarchitektur für das pDWH-System sowie für operativ-dispositive AwS ist die Integration der operativen Ausführung mit der Analyse von Geschäftsprozessen über das zentrale Artefakt des pDWH-Systems. Zu berücksichtigende Anforderungen stellen die *flexible Anpassbarkeit* an Veränderungen der Prozessstruktur sowie die *bedarfs- und zeitgerechte Unterstützung* heterogener Nutzergruppen mit einer *prozessorientierten Perspektive* dar.
2. Durch die Bereitstellung einer Entwicklungsmethodik wird die Anwendbarkeit der Referenzarchitektur in betrieblichen Anwendungsszenarien angestrebt. Zudem soll im Rahmen eines mehrstufigen Entwicklungsvorgehens die Einordnung von Informationen zur operativen Prozessausführung in den Kontext der betrieblichen Leistungserstellung ermöglicht werden. Neben der prozessorientierten soll auch eine fachliche, geschäftsprozessorientierte Perspektive bereitgestellt werden. Die Herausforderung der Methodik stellt die Komplexität der Entwicklung sowie der konsistenten Abstimmung operativer und dispositiver Teilsysteme dar.

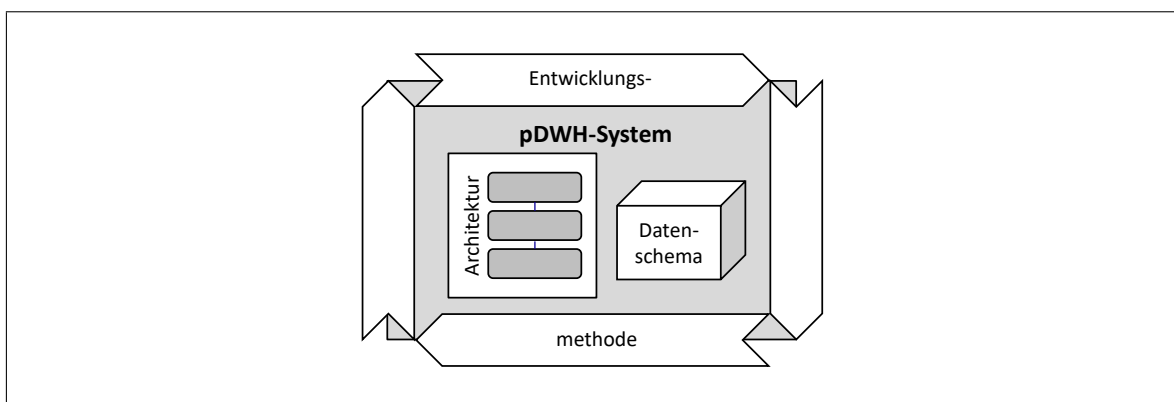


Abb. 1.2: Schematische Darstellung des Untersuchungsgegenstandes

Es wird davon ausgegangen, dass sich durch eine flexible Referenzarchitektur, ausgerichtet auf die eingeführten Nutzungsszenarien einschließlich einer darauf

abgestimmten Entwicklungsmethodik, positive Auswirkungen hinsichtlich der Effizienz und Innovation in den Geschäftsprozessen einer Unternehmung ergeben.

1.3 Hypothesen der Arbeit

Zur Lösung des Untersuchungsproblems wird die Markt-Metapher von *Angebot und Nachfrage* aufgegriffen. In der Betriebswirtschaftslehre ist ein Markt wenigstens über Angebot, Nachfrage, Tauschobjekt (knappes Gut) und Tauschmittel (i.d.R. Geld) definiert (Gabler 2014, S. 2106f.). Als Tauschobjekt werden Daten bzw. Informationen zu Geschäftsprozessen festgelegt. Angebot und Nachfrage dienen der Bildung von Teilsystemen. Als angebotsorientiert wird ein Teilsystem charakterisiert, das ausgehend von Vorsystemen (Quellsystemen) ein aufbereitetes und vollständiges Informationsangebot bereitstellt. Das davon entkoppelte nachfrageorientierte Teilsystem richtet sich nach dem Informationsbedarf der Systemnutzer und erfüllt diesen auf Basis des angebotsorientierten Teilsystems. Das Tauschmittel wird, auch aufgrund mangelnder Knappheit von Prozessdaten (vollständiges Angebot), im Rahmen der Metapher vernachlässigt. Ausgerichtet auf diese Metapher werden fünf Hypothesen formuliert:

Hypothese: *Die Komponentenorientierung stellt ein geeignetes Paradigma zur Strukturierung operativ-dispositiver Anwendungssysteme dar.*

Das pDWH-System bildet den zentralen Architekturbaustein zur Integration operativer AwS auf Angebots- und dispositiver Systeme auf Nachfrageseite. Zur Spezifikation einer flexiblen Systemstruktur für das pDWH-System sowie für das operativ-dispositive AwS stellt die Komponentenorientierung geeignete Gestaltungsprinzipien bereit. Minimale Kopplung zwischen Komponenten bei maximaler Kohäsion innerhalb einer Komponente (z.B. Szyperski et al. 2002, S. 40) oder die lose Kopplung über wohldefinierte Schnittstellen (z.B. Ackermann et al. 2002) sind stellvertretend als wesentliche Eigenschaften des Komponentenentwurfs zu nennen.

Hypothese: *über das Konzept der betrieblichen Aufgabe können aus pDWH-Systemen abgeleitete Informationen auch im Kontext der betrieblichen Leistungserstellung bewertet werden.*

Ein Geschäftsprozess, als Menge vernetzter betrieblicher Aufgaben beschrieben,

spezifiziert die betriebliche Leistungserstellung (Ferstl & Sinz 2013, S. 196f.) und kann als Brücke zwischen operativer und strategischer Unternehmensebene verstanden werden (Marjanovic 2007). Das Modell der betrieblichen Aufgabe kann, verbunden mit der Differenzierung von Außen- und Innensicht, als Hilfsmittel zur Einordnung softwarenah angebotener Ablaufdaten in den übergeordneten betrieblichen Kontext dienen. Neben dem Informationsbedarf technisch-orientierter Nutzer kann auch der Bedarf von Nutzern mit betriebswirtschaftlichem Hintergrund gedeckt werden.

Hypothese: *Das für themenorientierte DWH-Systeme etablierte multidimensionale Datenmodell ist auch zur Realisierung prozessorientierter DWH-Systeme anwendbar.*

Die Datenbereitstellung in DWH-Systemen erfolgt, ausgerichtet an relevanten Entscheidungsfeldern (*subject-oriented*), in multidimensionaler Form (Sinz 2002b). Das multidimensionale Datenmodell ist auch zur Datenbereitstellung in prozessorientierten DWH-Systemen, sowohl auf der Angebots- als auch auf der Nachfrageseite, anwendbar. Aus der übergeordneten Zielsetzung, der effizienteren und innovativeren Durchführung von Geschäftsprozessen nach Becker & Kahn (2012, S. 1f.), ergeben sich jedoch auch Herausforderungen: Zum einen wird neben dem Prozessergebnis auch der Ablauf zum Gegenstand von DWH-Systemen und damit auch des multidimensionalen Datenschemas. Zum anderen ist von der stetigen Anpassung der operativen Prozessstrukturen (z.B. durch deren innovativere Gestaltung) auszugehen und im pDWH-System entsprechend zu berücksichtigen.

Hypothese: *Prozessorientierte DWH-Systeme stellen eine konsistente Erweiterung prozessübergreifender themenorientierter DWH-Systeme dar.*

Themenorientierte DWH-Systeme sind, basierend auf einem aggregierten Datenbestand, auf bestimmte Analysethemen ausgerichtet und abstrahieren von einzelnen Geschäftstransaktionen (Bauer & Günzel 2013, S. 10f.). pDWH-Systeme hingegen führen explizit Daten auf feinsten Granularität der Geschäftstransaktionen und assoziieren diese mit relevanten Prozessdaten. pDWH-Systeme können als Vorsysteme themenorientierter Systeme verstanden und als solche integriert werden. Die Bildung von Teilsystemen anhand von *Angebot und Nachfrage* im pDWH-System ermöglicht heterogene Schnittstellen zur Integration. Mehrwert aus der Integration ergibt sich v.a. mit Blick auf die Konsistenz der Aussagen, die aus beiden Systemen abgeleitet werden können.

Hypothese: *Bestehende Ansätze der geschäftsprozessorientierten Systementwicklung können zur Entwicklung operativ-dispositiver Anwendungssysteme erweitert werden.*

Methodiken zur Entwicklung betrieblicher operativer AwS, die das Konzept des Geschäftsprozesses als zentrales Entwicklungsartefakt verstehen, können zur Entwicklung geschäftsprozessorientierter operativ-dispositiver Anwendungssysteme erweitert werden. Anhand der Metapher von *Angebot und Nachfrage* kann dabei die zeitlichen Entkopplung der Entwicklungsabschnitte einzelner Teilsysteme beschrieben und begründet werden. Die Entwicklungsmethodik kann zudem für bestehende Entwicklungsprojekte übernommen und in diese integriert werden und erfordert keine vollständige Neugestaltung der operativen Systeme.

1.4 Lösungsstrategie und methodisches Vorgehen der Arbeit

Im Weiteren wird in Abschn. 1.4.1 die grundsätzliche Strategie zur Lösung des Untersuchungsproblems vorgestellt sowie in Abschn. 1.4.2 auf die methodische Behandlung eingegangen.

1.4.1 Lösungsstrategie

Die Architektur sowie die Entwicklung von operativ-dispositiven AwS bilden das Untersuchungsobjekt. Ziel ist es, eine geeignete Struktur für beide zur Realisierung der postulierten Eigenschaften *Flexibilität, Geschäftsprozessorientierung* sowie *bedarfs- und zeitgerechte Informationsversorgung* zu finden. Das Untersuchungsproblem wird als Konstruktionsproblem nach Sinz (2010) klassifiziert. Als grundsätzliche Lösungsstrategie leitet sich daraus ein gestaltungsorientiertes Vorgehen ab, womit sich die Arbeit in den entsprechenden Gegenstandsbereich der Wirtschaftsinformatik einordnet. Als Erkenntnisziele der gestaltungsorientierten Wirtschaftsinformatik werden „(...) *Handlungsanleitungen (normative, praktisch verwendbare Ziel-Mittel-Aussagen) zur Konstruktion und zum Betrieb von betrieblichen Informationssystemen sowie Innovationen in den Informationssystemen (Instanzen) selbst*“ (Österle et al. 2010) beschrieben. An diesen Erkenntniszielen werden Strategie und methodisches Vorgehen der Arbeit unter Berücksichtigung nachfolgender Aspekte ausgerichtet:

- Bei der Konzeption der Referenzarchitektur sowie der Entwicklungsmethodik bildet die *Komponentenorientierung*, abgeleitet aus den Hypothesen der Arbeit in Abschn. 1.3, das durchgehende Systemparadigma.
- Zur Beherrschung der Komplexität bei der Bearbeitung des Untersuchungsproblems erfolgt die Konzeption entlang von *Aufgaben- und Aufgabenträgerebene*. Die Aufgabenebene ermöglicht durch Beschreibung von Aufgaben und Informationsbeziehungen die fachliche Betrachtung des Untersuchungsobjekts. Auf Aufgabenträgerebene wird hingegen die Aufgabendurchführung durch einen Aufgabenträger (AT) betrachtet (Ferstl & Sinz 2013, S. 4f.). Anhand der Ebenen wird zudem die Differenzierung der Prozessperspektive (*Workflow* vs. *Geschäftsprozess*) ermöglicht (vgl. Hypothese Abschn. 1.3).
- Die Entwicklung *neuer* Modellierungssprachen und -methoden für betriebliche AwS ist nicht Ziel der Arbeit. Etablierte Lösungen werden aufgegriffen und hinsichtlich der Problemlösung erweitert. Das *Semantische Objektmodell* (SOM) nach Ferstl & Sinz (vgl. Kap. 2.2) wird als Methode zur geschäftsprozessorientierten Systementwicklung, die *Business Process Modell and Notation* (BPMN) (vgl. Kap. 6.1) als Modellierungssprache für Workflows und das *Semantische Data-Warehouse-Modell* (SDWM) nach Böhnlein (vgl. Kap. 3.4) zur Spezifikation multidimensionaler Datenschemata genutzt. Ergebnisse der Arbeit werden zunächst auf Grundlage dieser Konzepte erarbeitet, bevor abschließend deren Allgemeingültigkeit aufgezeigt wird.

Die Arbeit startet mit der Konzeption von Referenzarchitektur und Datenschema, auf deren Grundlage im nächsten Schritt eine Entwicklungsmethodik spezifiziert wird. Damit wird auch der Zielsetzung, die Anwendbarkeit der Referenzarchitektur in betrieblichen Szenarien zu ermöglichen, Rechnung getragen.

1.4.2 Methodisches Vorgehen

Die der Arbeit zugrunde gelegte Forschungsmethodik ist an den idealtypischen Erkenntnisprozess der gestaltungsorientierten Wirtschaftsinformatik nach Becker (2010) angelehnt. Entsprechend der Forderung von Frank (2006) wird die Methode an die spezifischen Anforderungen und Eigenschaften der Problemstellung angepasst. Die resultierende Methodik ist in Abb. 1.3 mit Ablauf- sowie Validie-

rungsbeziehungen angegeben. Neben projektspezifischen Anforderungen sind die allgemeinen Anforderungen an gestaltungsorientierte Ergebnisse nach Frank (2006) und Österle et al. (2010) berücksichtigt: *Abstraktion, Originalität, Begründung, Nutzen*. Grundsätzlich wird ein methodisches Vorgehen zur Argumentation (*Begründung*) gewählt, das sich anhand des Methodenspektrums der Wirtschaftsinformatik nach Wilde & Hess (2007) als qualitativ-konstruktiv, im Speziellen als konzeptionell- sowie argumentativ-deduktiv klassifizieren lässt.

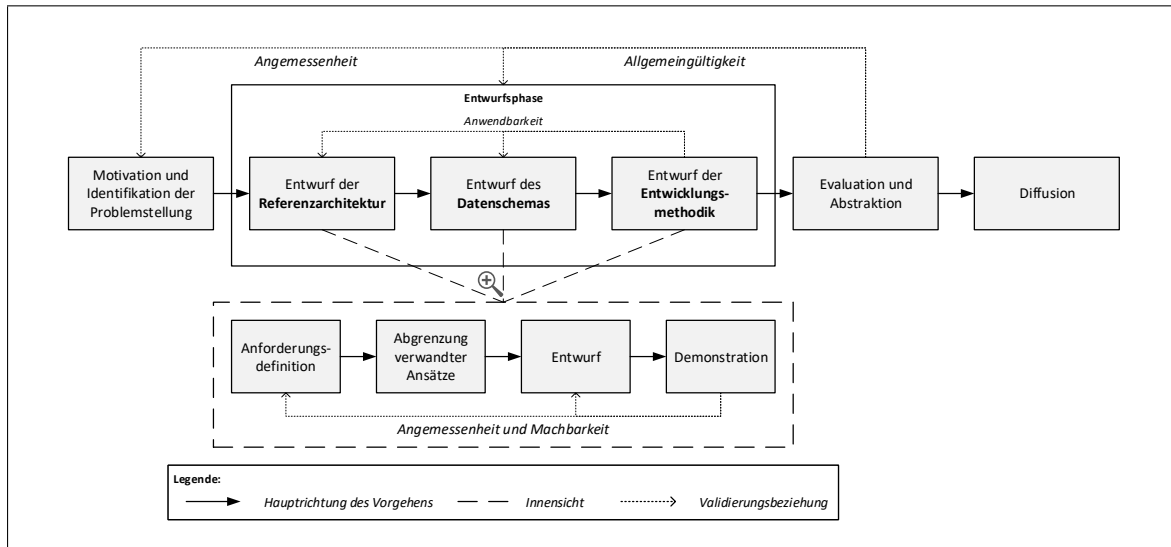


Abb. 1.3: Methodisches Vorgehen zur Lösung des Untersuchungsproblems

Motivation und Identifikation der Problemstellung bilden die *Analysephase* nach Becker (2010). Als Forschungsziele werden ebenfalls in dieser Phase die Zielartefakte der Referenzarchitektur mit Datenschema sowie der Entwicklungsmethodik identifiziert. Während der folgenden Entwurfsphase werden der Entwurf der Referenzarchitektur, der zugehörigen Datenstrukturen sowie der Entwicklungsmethodik aus Gründen der Komplexitätsbeherrschung in jeweils eigenen Schritten durchgeführt. Die sequentielle Abfolge der Schritte ergibt sich aus den Abhängigkeiten der Artefakte. Die Sequenz beschreibt dabei, wie im gesamten Ablaufschema, die Haupttrichtung des Vorgehens. Rücksprünge oder Iterationen zwischen den Schritten sind hinsichtlich Konsistenz und Qualität der Artefakte notwendig. Der Entwurf der einzelnen Artefakte erfolgt einheitlich. Neben der eigentlichen Konzeption der Artefakte erfolgt die Abgrenzung gegenüber bekannten Lösungen aus Wissenschaft und Praxis (*Originalität*) (Becker 2010). Konzeption und Abgrenzung erfolgen auf Basis der Literatur sowie daraus extrahierter Anforderungen (*Begründung*).

Die Validierung der Ergebnisse erfolgt mehrstufig. Begleitend zur Konzeption der einzelnen Artefakte erfolgt die Demonstration der Ergebnisse anhand einer Fallstudie und dient gleichzeitig als *Machbarkeitsstudie*. Angelehnt an Frank (2006) wird zudem die *Angemessenheit* der Artefakte zur Realisierung der erhobenen Anforderungen bewertet. Einen Aspekt der Angemessenheit stellt die Anwendbarkeit der Ergebnisse dar. Die Entwicklungsmethodik mit der Zielsetzung der *Anwendbarkeit* der Referenzarchitektur (vgl. Abschn. 1.2.2) wird als Beitrag zur Validierung verstanden. *Evaluation und Abstraktion* bilden den artefaktübergreifenden Teil der Validierung und den *Proof-of-Concept* im eigentlichen Sinn. Basierend auf Frank (2006) erfolgt die Evaluation gegenüber den Hypothesen der Arbeit. Abstraktion beschreibt die Allgemeingültigkeit der Ergebnisse. In der Arbeit wird von Modellierungssprachen und Anwendungsfällen, die während der Entwurfsphase Anwendung finden, abstrahiert. Zur Generierung eines *Nutzens* gegenüber Interessenten aus Wissenschaft und Praxis werden auf Basis der Ergebnisse normative Aussagen abgeleitet und veröffentlicht. Einen wichtigen Teil der Evaluation übernehmen auch wissenschaftliche Begutachtungen und Diskussionen der Artefakte (Becker 2010), die in der Phase der *Diffusion* zielgerichtet veröffentlicht wurden bzw. noch werden. Neben der Veröffentlichung dieser Arbeit wurden zur Qualitätssicherung Ergebnisse auch bei nachfolgenden Konferenzen und Workshops diskutiert:

- Begutachtung, Diskussion und Publikation (Benker 2013) des initialen Architekturentwurfs im Rahmen des *Fifth International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2013)*.
- Begutachtung und Diskussion des Exposé zur Arbeit im Rahmen des Doktorandensymposiums der Tagung *Modellierung 2014*.
- Begutachtung und Diskussion des fortgeschrittenen Architekturkonzeptes im Rahmen des *7. Workshop für Business Intelligence (WSBI 2015)* sowie Veröffentlichung (Benker 2015) als *Bamberger Beitrag zur Wirtschaftsinformatik und Angewandten Informatik (Nr. 97)*.
- Begutachtung, Diskussion und Publikation (Benker 2016) des pDWH-Schemas im Rahmen der *19th International Conference on Business Information Systems (BIS 2016)*.

1.5 Aufbau der Arbeit

Der Aufbau der Arbeit, schematisch dargestellt in Abb. 1.4, leitet sich aus der Spezifikation des Untersuchungsproblems (Abschn. 1.2) sowie der Forschungsmethode (Abschn. 1.4) ab. Die Kapitel 2 bis 4 bilden das Fundament der Arbeit. In **Kapitel 2** werden methodische Grundlagen zur Gestaltung betrieblicher Informationssysteme über mehrere Beschreibungsebenen vom Geschäftsprozessmodell hin zur Automatisierung basierend auf dem komponentenorientierten Paradigma eingeführt. **Kapitel 3** führt darauf folgend Grundlagen zu DWH-Systemen im Allgemeinen sowie zu *prozessorientierten* DWH-Systemen im Speziellen ein. In **Kapitel 4** wird eine umfassende Fallstudie vorgestellt. Diese dient der Demonstration sowie als Machbarkeitsstudie für die Ergebnisse der Arbeit in nachfolgenden Kernkapiteln.

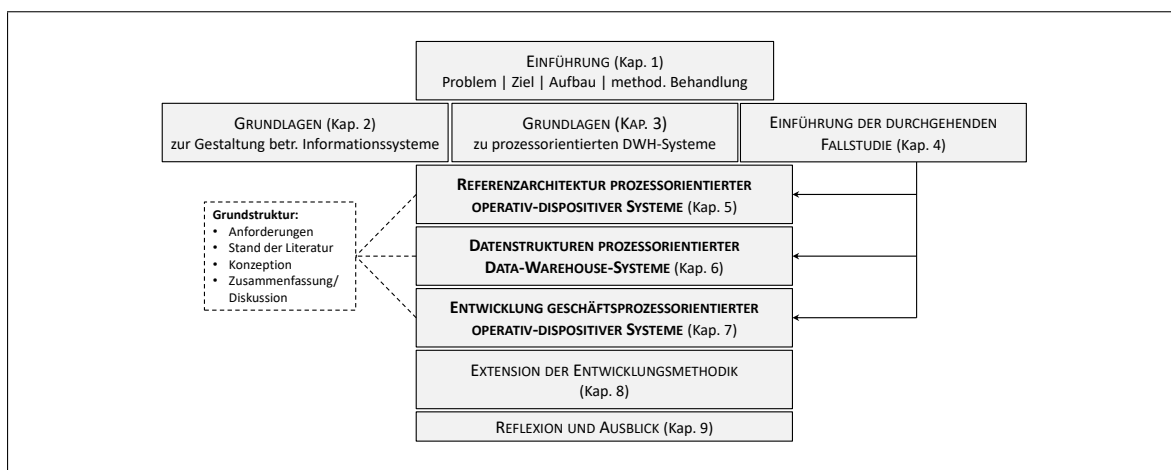


Abb. 1.4: Aufbau der Arbeit

Kapitel 5, 6 und 7 bilden den Kern der Arbeit. Sie dienen der Vorstellung sowie der Begründung der zentralen Ergebnisse. Sie folgen einer einheitlichen Grundstruktur. Zunächst werden *Anforderungen* an den jeweiligen Teil des Untersuchungsobjektes auf Basis der Literatur erarbeitet. Anhand dieser Anforderungen werden unter *Stand der Literatur* verwandte Arbeiten gespiegelt und diskutiert. Im Kern des Kapitels erfolgt die *Konzeption* des jeweiligen Artefakts. Den Kapitelabschluss bilden Zusammenfassung und Diskussion der Ergebnisse mit Blick auf die eingangs formulierten Anforderungen. **Kapitel 5** stellt eine Referenzarchitektur für prozessorientierte operativ-dispositive AwS vor. Im Fokus der *funktionsorientierten* Betrachtung steht im Speziellen das pDWH-System aus Außen- und Innensicht sowie dessen Integration in die Komponentenarchitektur des ganzheitli-

chen Anwendungssysteme. In **Kapitel 6** werden die *Datenstrukturen* als zentraler Aspekt des prozessorientierten DWH-Systems bzw. der Referenzarchitektur vertieft betrachtet. Zum einen wird ein generisches multidimensionales Schema für pDWH-Systeme betrachtet, das als Grundlage zur Entscheidungsfindung in aktiven Prozessinstanzen als auch zur Gestaltung des Prozessschemas dient. Zum anderen wird die Bereitstellung des Datenschemas anhand der in Kapitel 5 vorgestellten Komponenten thematisiert. Die Entwicklung geschäftsprozessorientierter operativ-dispositiver Komponentenarchitekturen ist Gegenstand von **Kapitel 7**. Die Entwicklungsmethodik ist auf die anhand ihrer Funktionen und Datenstrukturen (Kapitel 5 u. 6) vorgestellte Referenzarchitektur ausgerichtet. In Kapitel 5 und 6 wird der Prozessbegriff im Sinne eines *Workflows* als Lösungsverfahren betrieblicher Aufgaben verwendet. Durch die Entwicklungsmethodik erfolgt die Einordnung der Datenstrukturen, Funktionen und Komponenten in den fachlichen Kontext von Geschäftsprozessen und begründet den Begriff des *geschäftsprozessorientierten AwS*.

Kapitel 8 u. 9 bilden den Abschluss der Arbeit. **Kapitel 8** thematisiert über die Extension der Entwicklungsmethodik die Gültigkeit der Ergebnisse und trägt damit maßgeblich zum *Proof-of-Concept* bei. Gegenstand sind die Einordnung der Ergebnisse sowie die Abstraktion von Modellierungssprachen und Methoden, die im Rahmen der Konzeption angewendet werden. Zudem werden allgemeine Merkmale von Geschäftsprozessen aufgezeigt, anhand derer sich Anwendungsfälle operativ-dispositiver AwS abgrenzen lassen. Das finale **Kapitel 9** reflektiert die aufgestellten Hypothesen anhand der Ergebnisse der Arbeit und gibt darauf aufbauend einen Ausblick auf künftige Aktivitäten.

2 Methodische Grundlagen zur Gestaltung betrieblicher Informationssysteme

Kapitelzusammenfassung: Das *betriebliche Informationssystem* stellt den zentralen Untersuchungsgegenstand der Wirtschaftsinformatik dar. Ausgehend von der Abgrenzung des betrieblichen Informationssystems anhand verschiedener Kriterien werden Grundlagen zu dessen Gestaltung eingeführt (Abschn. 2.1). Auf Basis dieser Abgrenzung werden der zentrale Begriff der Arbeit, der des *Geschäftsprozesses* näher bestimmt sowie Grundlagen der SOM-Methodik zur modellbasierten Entwicklung betrieblicher Anwendungssysteme auf Basis von Geschäftsprozessmodellen geschaffen (Abschn. 2.2). Die Konzeption dieser Anwendungssysteme basiert in der Arbeit auf dem komponentenorientierten Paradigma, dessen methodische Grundlagen abschließend eingeführt werden (Abschn. 2.3).

Abschn. 2.1 basiert in Teilen auf der Veröffentlichung Benker & Jürck (2016, S. 25f.).

2.1 Grundlagen betrieblicher Informationssysteme

Zu Beginn werden für die Arbeit notwendige Grundlagen betrieblicher Informationssysteme vorgestellt. Zunächst erfolgt die Abgrenzung und Bestimmung des Konzeptes *Informationssystem* (Abschn. 2.1.1), bevor das betriebliche Aufgabenmodell als ein zentrales Beschreibungskonzept eingeführt wird (Abschn. 2.1.2). Grundlagen zur Modellierung des betrieblichen Informationssystems werden schließlich mit Abschn. 2.1.3 geschaffen.







2.1.1 Abgrenzung des betrieblichen Informationssystems

Kerngegenstand der Wirtschaftsinformatik bilden Informationssysteme als Teil betrieblicher Systeme (beispielsweise Unternehmen oder Unternehmensbereiche) aus Wirtschaft und Verwaltung sowie mit zunehmender Vernetzung auch

privater Haushalte (Ferstl & Sinz 2013, S. 3). Unter einem Unternehmen wird in der Wirtschaftsinformatik ein *offenes, sozio-technisches* und *zielorientiertes* System verstanden. Unternehmen treffen eigenständige Entscheidungen bei eigenem Risiko. *Offen* meint die Interaktion eines Unternehmens mit seiner Umwelt, *zielorientiert* die Ausrichtung eines Unternehmens anhand eines Zielsystems bestehend aus Sach- und Formalzielen. Als *sozio-technisch* werden Systeme bezeichnet, die sowohl technische Komponenten als auch Personen samt Mensch-Technik-Interaktion umfassen. Die Interaktion erfolgt zweckorientiert zur Erreichung der genannten Ziele (Ferstl & Sinz 2013, S. 65f.). Ein betriebliches System, präziser ein *betriebliches Objektsystem*, kann anhand unterschiedlicher Kriterien in Teilsysteme strukturiert werden. Das betriebliche Objektsystem bezeichnet einen zweckorientiert abgegrenzten Ausschnitt der Realität (*betriebliche Diskurswelt*) sowie seine Beziehungen zu Objekten der Umwelt (Ferstl & Sinz 2013, S. 7). Ferstl & Sinz (2013, S. 7f.) schlagen als Kriterien zur Strukturierung *Objektprinzip*, *Phasenprinzip* sowie die *Art des Aufgabenträgers* vor. Die resultierenden Teilsysteme werden nachfolgend vorgestellt und sind in Abb. 2.1 dargestellt (Ferstl & Sinz 2013, S. 7f.):

- Entlang des *Objektprinzips* wird das betriebliche Objektsystem anhand behandelte bzw. ausgetauschte Objekte strukturiert. Das resultierende *Informationssystem* bezeichnet dabei das Teilsystem, das die Objektart Information verarbeitet bzw. austauscht. Das *Basissystem* beschreibt das Teilsystem, das nicht Informationen sondern die physischen Güter (beispielsweise materielle Güterflüsse) zum Gegenstand hat.
- Das *Phasenprinzip* strukturiert das betriebliche Objektsystem anhand der Aufgabenphasen Planung, Steuerung, Kontrolle der Leistungserstellung sowie der Leistungserstellung selbst. Letztere wird dem *Leistungssystem* zugeordnet. Planung, Steuerung und Kontrolle bilden das *Lenkungssystem*.
- Abgegrenzt durch die *Art des Aufgabenträgers* werden das *automatisierte* sowie das *nicht automatisierte* Teilsystem unterschieden. Personen übernehmen dabei nicht automatisierte Aufgaben, Maschinen, sowohl Anwendungssysteme als auch Produktions- oder Transportmaschinen die automatisierten Aufgaben.

Die in Abb. 2.1 dargestellten Grenzen der Teilsysteme bestimmen sich jeweils in Abhängigkeit vom konkreten Objektsystem. Informations- und Lenkungssys-

Objektprinzip	Aufgabenträger		Phasenprinzip
	automatisiert	nicht automatisiert	
Informationssystem (Information)	Anwendungssysteme 		Lenkungssystem (Planung, Steuerung, Kontrolle)
			
Basissystem (Nicht-Information)			Leistungssystem (Durchführung)

Informationssystem i.w.S.
Informationssystem i.e.S.

Abb. 2.1: Abgrenzung des Informationssystems im betrieblichen Objektsystem (Benker & Jürck 2016 basierend auf Ferstl & Sinz 1995)

tem können zusammenfallen, wenn ausschließlich physische Güter durch das Leistungssystem erstellt werden. Beratungsunternehmen als reine Dienstleister verfügen ausschließlich über ein Informationssystem. Das Basissystem entfällt in diesem Fall. Das Informationssystem umspannt dann sowohl Lenkungs- als auch Leistungssystem (Ferstl & Sinz 2013, S. 7). Der Begriff des Informationssystems nach Ferstl & Sinz (2013) umfasst sowohl die automatisierten als auch die nicht automatisierten Aufgabenträger. Diese weite Begriffsabgrenzung wird nachfolgend als *Informationssystem im weiteren Sinne (i.w.S.)* bezeichnet. Häufig findet sich der Begriff jedoch auch enger abgegrenzt und kann als *Informationssystem im engeren Sinn (i.e.S.)* bezeichnet werden. Bei Gabriel (2014) wird so das Informationssystem mit dem des betrieblichen Anwendungssystems im Leistungssystem (z.B. operative Transaktionssysteme) oder im Lenkungssystem (z.B. Management Support Systeme) gleichgesetzt. Beide Begriffsinterpretationen sind in Abb. 2.1 kartiert. Im weiteren Verlauf der Arbeit wird unter einem Informationssystem das Informationssystem i.w.S. verstanden. Zentraler Bestandteil des Begriffs des Informationssystems ist die *Information*. Nach Ferstl & Sinz (2013, S. 144f.) stellt eine Information die semantische Interpretation von Daten (codierte Zeichenfolge auf syntaktischer Ebene) dar. Eine Information leitet sich durch Anwendung einer Interpretationsvorschrift auf diesen Daten ab (Bauer et al. 1991, S. 4). Daten selbst stellen das Ergebnis der Modellierung eines Ausschnittes der betrieblichen Realität dar. Wissen bzgl. dieser Modellbildung ist in der Interpretationsvorschrift zu berücksichtigen (Ferstl & Sinz 2013, S. 144f.).

2.1.2 Einführung des betrieblichen Aufgabenmodells

Ein zentrales Konzept zur Beschreibung des betrieblichen Informationssystems stellt die *betriebliche Aufgabe* dar. Das Informationssystem wird dabei als Menge von Aufgaben zur Informationsverarbeitung verstanden, die über entsprechende Informationsbeziehungen miteinander verknüpft sind (Ferstl & Sinz 2013, S. 4). Der Begriff der Aufgabe basiert auf den Ausführungen von Kosiol (1976), der eine Aufgabe als Zielsetzung zweckbezogenen menschlichen Handelns über die fünf Merkmale Verrichtungsvorgang, Aufgabenobjekt, benötigte Arbeits- und Hilfsmittel sowie Raum und Zeit der Verrichtung charakterisiert. Ferstl & Sinz (2013, S. 98) definieren die Struktur einer betrieblichen Aufgabe aus Außen- und Innensicht, die in Abb. 2.2 dargestellt ist.

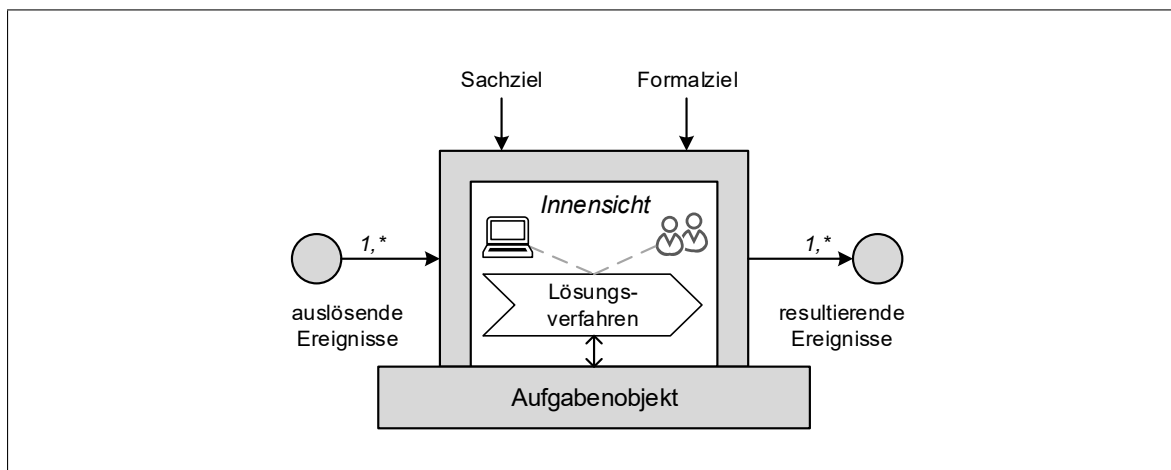


Abb. 2.2: Struktur der betrieblichen Aufgabe (Benker & Jürck 2016 basierend auf Ferstl 1992)

Die *Außensicht* einer Aufgabe ist über folgende Bestandteile beschreibbar (Ferstl & Sinz 2013, S. 97ff.): Das *Aufgabenobjekt* beschreibt die von der Aufgabendurchführung betroffenen Datenstrukturen. Die Auslöser der Aufgabendurchführung stellen *Vorereignisse*, die Resultate *Nachereignisse* dar. Die bei Aufgabendurchführung zu erreichenden Zielsetzungen werden über *Sach-* und *Formalziele* spezifiziert. Das Lösungsverfahren zur Durchführung der Aufgabe ist Teil der *Innensicht* (Ferstl & Sinz 2013, S. 98ff.): Die Beschreibung des Lösungsverfahrens erfolgt ausgerichtet auf einen *Aufgabenträgertyp*. Hier können personelle Aufgabenträger, maschinelle Aufgabenträger (Anwendungssysteme) oder eine Kombination aus beiden als Mensch-Rechner-Systeme unterschieden werden. Aufgabenträger sind für die eigenständige Aufgabendurchführung verantwortlich. Die Aufgabendurchführung erfolgt in Form von *Vorgängen*. Die Durchführung eines Lösungs-

verfahrens auf einem Aufgabenobjekt wird über einen Vorgangstyp spezifiziert und zur Laufzeit instanziiert. Das Lösungsverfahren ist als Menge von Aktionen auf dem Aufgabenobjekt sowie der übergeordneten Aktionensteuerung zur Berücksichtigung der Sach- und Formalziele beschreibbar (Ferstl & Sinz 2013, S. 102ff.).

2.1.3 Modellierung betrieblicher Informationssysteme

Betriebliche Informationssysteme setzen sich aus einer Vielzahl von Aufgaben, deren Aufgabenträgern (Personen und AwS) sowie der zwischen ihnen bestehenden Interaktions-, Koordinations- und Kommunikationsbeziehungen zusammen. Zur Beherrschung der damit einhergehenden Komplexität bei Analyse und Gestaltung betrieblicher Informationssysteme haben sich Modelle mit damit in Beziehung stehenden Mechanismen der Komplexitätsbeherrschung etabliert (Benker & Jürck 2016, S. 28). Der generische Architekturrahmen nach Sinz (1997) zur Spezifikation der Modellarchitekturen betrieblicher Informationssysteme beschreibt diese Konzepte im Zusammenhang und wird nachfolgend eingeführt.

Der generische Architekturrahmen

Der generische Architekturrahmen, dargestellt in Abb. 2.3 nach Sinz (1997), berücksichtigt als Konzepte zur Beherrschung der Komplexität bei Analyse und Gestaltung betrieblicher Informationssysteme die Modellbildung auf heterogenen Beschreibungsebenen sowie die Verwendung von Sichten und Strukturmustern (*Patterns*) innerhalb einer Ebene. Die nachfolgenden grundlegenden Ausführungen zum generischen Architekturrahmen entstammen der Spezifikation nach Sinz (1997) und werden, sofern notwendig, um weitere Quellen ergänzt: Eine *Modellebene* umfasst die vollständige Beschreibung eines Systems unter einer bestimmten Perspektive. Ein Beispiel hierfür ist die konzeptuelle Modellierung mit Betrachtung des Systems aus fachlicher Perspektive. Die Realisierung eines AwS mit Plattformbezug stellt als weiteres Beispiel die softwaretechnische Betrachtung in den Mittelpunkt (Benker & Jürck 2016, S. 30). Ausgerichtet auf den Modellierungszweck, also die zu unterstützende Perspektive einer Beschreibungsebene, ist als zentrales Konzept ein *Metamodell* vorgesehen. Ein Metamodell beschreibt das zur Modellierung zur Verfügung stehende Begriffssystem. Dies umfasst die zur Verfügung stehenden Modellbausteine und Beziehungen (einschl. der grafi-

schen Darstellung), die Semantik von Bausteinen und Beziehungen sowie Regeln zu deren Verknüpfung (Ferstl & Sinz 2013, S. 137f.). Eine umfassende Einführung von Konzepten der Modellierung und Metamodellierung findet sich beispielsweise bei Strahringer (2016). *Sichten* gestalten die Komplexität einer Modellebene beherrschbar und beschreiben einen Ausschnitt dieser. Bei Sinz (1999) und Ferstl & Sinz (2013, S. 142f.) werden vier wesentliche Sichten zur Beschreibung des betrieblichen Informationssystems unterschieden: Zu den statischen, strukturorientierten Sichten zählen die *Funktions-*, die *Daten-* sowie die *Interaktionssicht*. Die *Funktionssicht* hebt die Funktionen des betrieblichen Informationssystems bzw. seiner Bausteine hervor, während die *Datensicht* auf die zu unterstützenden Datenstrukturen fokussiert. Die *Interaktionssicht* beschreibt Kommunikationskanäle zwischen Bestandteilen bzw. Teil-Funktionen des Informationssystems. Das Verhalten des Informationssystems wird über die dynamische *Vorgangssicht* beschrieben. Sichten können als Projektionen auf ein Metamodell interpretiert und realisiert werden. Die Wiederverwendung von Modellierungswissen oder auch strukturelle Integritätsbedingungen können in Form von Strukturmustern einer Ebene zugeordnet werden.

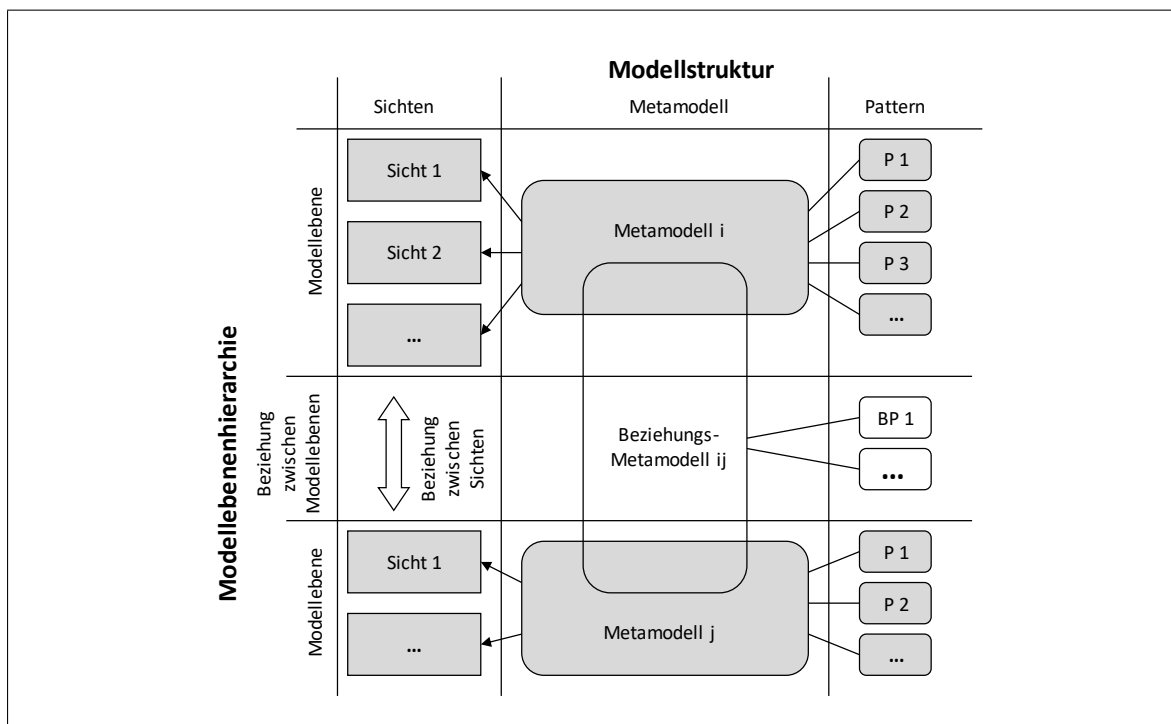


Abb. 2.3: Generischer Architekturrahmen (Sinz 1997)

Auf den verschiedenen aufeinander aufbauenden Modellebenen können jeweils zur Zielsetzung der Ebene passende Modellierungssprachen eingesetzt werden.

Dies führt zu der Notwendigkeit, Beziehungen zwischen Ebenen bzw. der dort verorteten Modellierungssprachen näher zu beschreiben (Benker & Jürck 2016, S. 31). Der generische Architekturrahmen sieht hierzu die Beschreibung über die Beziehung der Objekte der Metamodelle vor. Das Konzept wird als *Beziehungs-Metamodell* bezeichnet. Über Beziehungsmuster kann heuristisches Wissen bzgl. der Modellableitung berücksichtigt werden.

Modellebenen des betrieblichen Informationssystems

Als generelle Abstraktionsebenen zur orthogonalen Strukturierung der Teilsysteme aus Abschn. 2.1.1, im Speziellen des betrieblichen Informationssystems, können *Aufgaben-* und *Aufgabenträgerebene* (AT-Ebene) unterschieden werden. Die Abgrenzung der Ebenen folgt dem Modell der *betrieblichen Aufgabe* sowie der Differenzierung von *Innen-* und *Außensicht* (vgl. Abschn. 2.1.2). Auf Aufgabenebene wird das Informationssystem in Form von Aufgaben zur Informationsverarbeitung (aus Außensicht) beschrieben, die über Informationsbeziehungen interagieren. Die AT-Ebene beschreibt Aufgabenträger zur Durchführung der Aufgaben anhand spezifischer Lösungsverfahren (Innensicht der betrieblichen Aufgabe) (Ferstl & Sinz 2013, S. 4f.). Der Zusammenhang der Ebenen ist in Abb. 2.4 schematisch dargestellt.

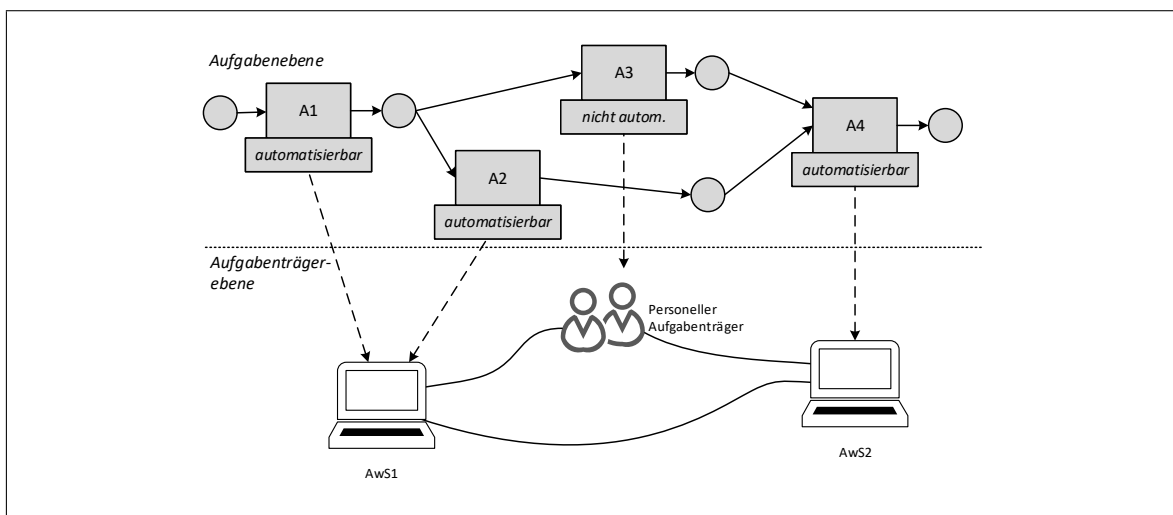


Abb. 2.4: Zusammenhang von Aufgaben- und Aufgabenträgerebene (Benker & Jürck 2016)

Bei der Ableitung der Spezifikation der AT-Ebene aus der Aufgabenebene ist die Automatisierbarkeit der Aufgaben festzulegen. Wie in Abschn. 2.1.2 beschrieben, werden automatisierte, teilautomatisierte und nicht-automatisierte Aufgaben

unterschieden. Das Beispiel aus Abb. 2.4 nach Benker & Jürck (2016, S. 29) klassifiziert die Aufgaben A1, A2 und A4 als automatisierbar. Diese werden auf AT-Ebene AwS1 (A1 u. A2) und AwS2 (A4) zugeordnet. Die nicht automatisierbare Aufgabe (A3) wird personell durchgeführt. Zudem sind die Informationsbeziehungen zwischen Aufgaben auf Kommunikationsbeziehungen zwischen Aufgabenträgern abzubilden (Ferstl & Sinz 2013, S. 5). Die Kommunikation von AwS1 mit dem personellen Aufgabenträger des Beispiels (Abb. 2.4) leitet sich aus der Aufgabenbeziehung A1-A3, die Beziehung zu AwS2 aus der Informationsbeziehung A2-A4 ab.

2.2 Gestaltung betrieblicher Informationssysteme auf Basis des Semantischen Objektmodells

Der Begriff *Geschäftsprozess* ist für das Untersuchungsproblem der Arbeit sowie generell für das betriebliche Informationssystem von zentraler Bedeutung. Zunächst wird dieser Begriff abgegrenzt (Abschn. 2.2.1) und in den Kontext der SOM-Methodik eingeordnet (Abschn. 2.2.2). Das Semantische Objektmodell (SOM) stellt zum einen die methodische Grundlage zur Beschreibung der Fallstudie der Arbeit dar, zum anderen bildet sie die Basis der Konzeption der Entwicklungsmethodik für GP-orientierte operativ-dispositive AwS in Kap. 7. In diesem Abschnitt werden zunächst allgemeine methodische Grundlagen zur SOM-Methodik eingeführt. Eine detaillierte Betrachtung erfolgt mit der Anwendung der einzelnen SOM-Konzepte in nachfolgenden Kapiteln.

2.2.1 Einordnung des Prozessbegriffs im Kontext des betrieblichen Informationssystems

Der Begriff *Prozess* wird im Kontext der Modellierung betrieblicher Informationssysteme im Wesentlichen in den Spezialisierungen *Geschäftsprozess* und *Workflow* verwendet. Bei Pütz & Sinz (2010) werden beide Begriffe, basierend auf dem Geschäftsprozessverständnis nach Ferstl & Sinz (2013, S. S. 196f.), abgegrenzt:

- **Geschäftsprozess:** Die Spezifikation von Geschäftsprozessen erfolgt anhand betrieblicher Aufgaben (vgl. Abschn. 2.1.2) und der Ereignisbeziehungen zwischen diesen. Ausgerichtet an den Unternehmenszielen erfolgt in einem Geschäftsprozessmodell (GP-Modell) die Spezifikation der betriebli-

chen Leistungserstellung, der zugehörigen Lenkung sowie der benötigten Ressourcen. Ein GP-Modell ist der Aufgabenebene zugeordnet.

- **Workflow:** Workflows werden dagegen der AT-Ebene zugeordnet. Durch einen Workflow wird das Lösungsverfahren zur Durchführung einer oder mehrerer betrieblicher Aufgaben, ausgerichtet auf den jeweiligen Aufgabenträger, beschrieben. Als Aufgabenträger können sowohl maschinelle Aufgabenträger (von Interesse sind hier insbesondere AWS) sowie Personen sein. Die Spezifikation von Workflows erfolgt anhand von Aktivitäten (elementar oder nicht-elementar) und ihrer ablauforientierten Beziehungen.

In der Literatur finden sich weitere heterogene Interpretationen des Begriffs *Geschäftsprozess*. Nachfolgend werden exemplarisch zwei betrachtet. Bei Becker & Kahn (2012) wird eine v.a. bzgl. der Bedeutung der Unternehmensziele vergleichbare Definition des Begriffs *Geschäftsprozess* angegeben und die Interaktion mit der Unternehmensumwelt hervorgehoben: „*Ein Geschäftsprozess ist ein spezieller Prozess, der der Erfüllung der obersten Ziele der Unternehmung (Geschäftsziele) dient und das zentrale Geschäftsfeld beschreibt. Wesentliche Merkmale eines Geschäftsprozesses sind die Schnittstellen des Prozesses zu den Marktpartnern des Unternehmens (z.B. Kunden, Lieferanten)*“ (Becker & Kahn 2012, S. 6f.). Häufig erfolgt jedoch keine oder nur eine unscharfe Trennung zwischen *Geschäftsprozess* und *Workflow* (Pütz & Sinz 2010). Bei Weske (2012, S. 5) werden die Begriffe eher als Synonym verwendet. Dieser Definition folgend stellt ein Geschäftsprozess eine Menge von Aktivitäten einer Organisation zur Realisierung bestimmter Geschäftsziele dar. Die Aktivitäten sind in bestimmten organisationalen und technischen Umgebungen ausführbar (Weske 2012, S. 5). In dieser Interpretation bildet die Ausführbarkeit des Prozesses ein zentrales Kriterium und deutet auf ein Prozessverständnis auf AT-Ebene hin.

Für die vorliegende Problemstellung erscheint die Abgrenzung nach Pütz & Sinz (2010) sinnvoll und findet nachfolgend Anwendung. Durch die Abgrenzung der Begriffe wird die Verortung von Ergebnisartefakten auf Aufgaben- bzw. AT-Ebene deutlich sowie ein Beitrag zur Komplexitätsbeherrschung bei der Lösung des Untersuchungsproblems geleistet.

2.2.2 Die Unternehmensarchitektur der SOM-Methodik

Das *Semantische Objektmodell* stellt einen umfassenden Modellierungsansatz für betriebliche Systeme dar. Er bildet zusammen mit Unternehmensarchitektur und Vorgehensmodell (Abschn. 2.2.3) die *SOM-Methodik* (Ferstl & Sinz 2013, S. 194f.). Die SOM-Methodik betrachtet ein betriebliches System aus Außensicht als offen, sozio-technisch und ziegerichtet. Die Innensicht wird als verteiltes System verstanden, dessen Komponenten kooperieren, um gemeinsame übergeordnete Ziele zu erreichen (Ferstl & Sinz 2013, S. 197). Bei Ferstl & Sinz (2006) sowie Ferstl & Sinz (2013, S. 194ff.) wird die SOM-Methodik ganzheitlich beschrieben. Der Teilaspekt der objektorientierten Modellierung betrieblicher Informationssysteme auf Aufgabenträgerebene wird initial in Ferstl & Sinz (1990) vorgestellt und in Ferstl & Sinz (1991) in das Vorgehensmodell eingeordnet. Die Konzeption von Unternehmensarchitektur und Geschäftsprozessebene (Aufgabenebene) ist in Ferstl & Sinz (1995) erstmalig publiziert. Für nachfolgende Belege zur SOM-Methodik wird als aktuellste ganzheitliche Quelle Ferstl & Sinz (2013, S. 194ff.) verwendet.

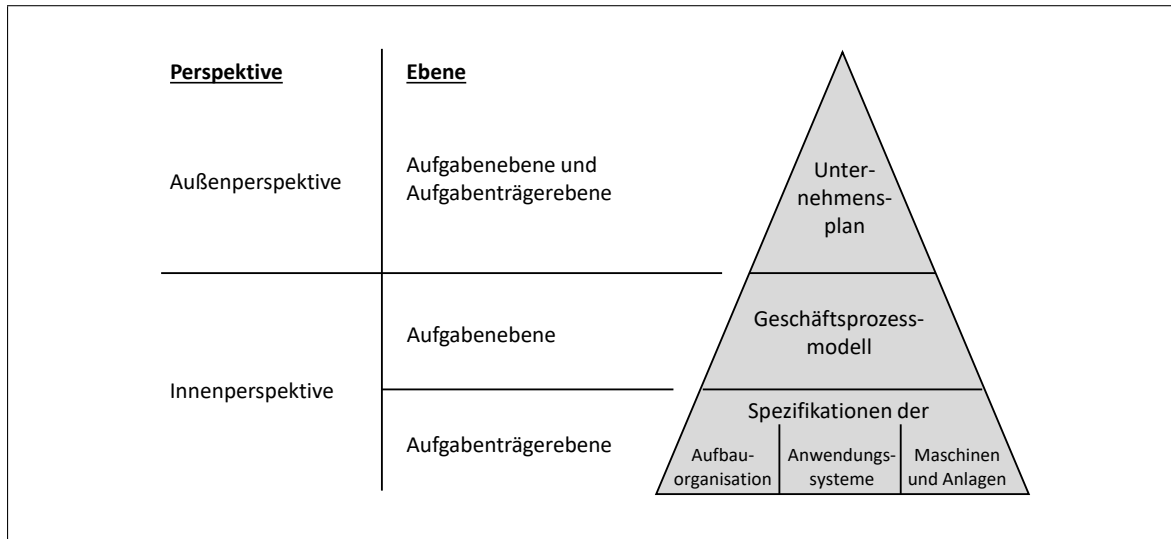


Abb. 2.5: Unternehmensarchitektur der SOM-Methodik (Ferstl & Sinz 2013, S. 195)

In der Unternehmensarchitektur (Abb. 2.5) ist die Beschreibung betrieblicher Systeme durch Teilmodelle vorgesehen. Mit der Zuordnung dieser Teilmodelle zu Modellebenen wird ein Beitrag zur Beherrschung der Komplexität der Modellierungsaufgabe geleistet. Die in Abschn. 2.1.3 eingeführte Differenzierung zwischen Aufgaben- und Aufgabenträgerebene wird, zusammen mit der Differenzierung zwischen Innen- und Außenperspektive, zur Abgrenzung der Modellebenen

herangezogen (Ferstl & Sinz 2013, S. 195f.) und damit auch die Differenzierung zwischen Geschäftsprozess- und Workflow-Verständnis (vgl. Abschn. 2.2.1) unterstützt. Die SOM-Methodik berücksichtigt auch die Beziehungen zwischen den nachfolgend eingeführten Teilmodellen, um deren konsistente Abstimmung zu ermöglichen.

- **Unternehmensplan** (Ferstl & Sinz 2013, S. 196): Der Unternehmensplan betrachtet ein betriebliches System als globale Unternehmensaufgabe (Metapher). Die Spezifikation berücksichtigt sowohl die Abgrenzung von Diskurswelt und Umwelt als Aufgabenobjekt, die Aufgabenziele, die Leistungsbeziehungen zwischen Diskurs- und Umwelt sowie die notwendigen Ressourcen. Das betriebliche System wird dabei aus Außenperspektive sowohl auf Aufgaben- (Leistungserbringung) als auch auf Aufgabenträgerebene (Ressourcen zur Leistungserbringung) beschrieben.
- **Geschäftsprozessmodell** (Ferstl & Sinz 2013, S. 196f.): Mit Betrachtung der Geschäftsprozessebene (GP-Ebene) erfolgt der Wechsel zur Innenperspektive auf das betriebliche System. Das Geschäftsprozessmodell beschreibt dabei auf Aufgabenebene das Lösungsverfahren der Unternehmensaufgabe ausgerichtet an der Metapher des verteilten Systems. Haupt- und Serviceprozesse, die sich hinsichtlich der Erfüllung gemeinsamer Ziele koordinieren, bilden die Systemkomponenten. Hauptprozesse dienen unmittelbar der Erfüllung der Sachziele des betrieblichen Systems und stehen in Leistungsbeziehungen mit der Umwelt. Serviceprozesse hingegen erstellen Leistungen für Haupt- oder andere Serviceprozesse.
- **Ressourcenmodell** (Ferstl & Sinz 2013, S. 197): Auf der dritten Ebene erfolgt die Spezifikation der Ressourcen des betrieblichen Systems (Innenperspektive, Aufgabenträgerebene). Von zentraler Bedeutung für das betriebliche Informationssystem sowie diese Arbeit ist die Ressource AwS zur automatisierten Aufgabendurchführung. Workflows (vgl. Abschn. 2.2.1) werden dieser Ebene zugeordnet. Weitere Ressourcen sind Maschinen und Anlagen sowie Personal. Auf AT-Ebene wird die teilautomatisierte Durchführung der Unternehmensaufgabe als sozio-technisches System (Metapher) beschrieben.

2.2.3 Das Vorgehensmodell der SOM-Methodik

Das Vorgehensmodell der SOM-Methodik (*V-Modell*, Abb. 2.6, initial vorgestellt in Ferstl & Sinz 1991) beschreibt auf drei mit der Unternehmensarchitektur korrespondierenden Ebenen die Modellierung betrieblicher Systeme. Je Modellebene erfolgt die Modellierung sowohl aus struktur- (linker V-Schenkel) als auch aus verhaltensorientierter Sicht (rechter V-Schenkel). Die Hauptrichtung der Modellierung ist, mit abnehmenden Freiheitsgraden in der Modellgestaltung, entlang der Ebenen von oben nach unten (Ferstl & Sinz 2013, S. 197ff.).

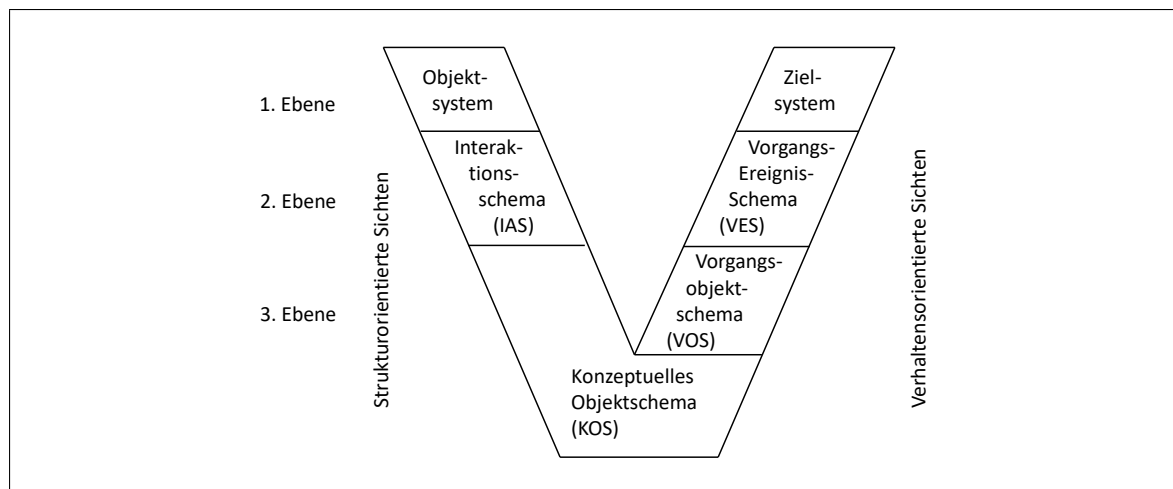


Abb. 2.6: Vorgehensmodell der SOM-Methodik (Ferstl & Sinz 2013, S. 198)

- **1. Ebene** (Ferstl & Sinz 2013, S. 198): Die Spezifikation des Unternehmensplans erfolgt durch Objekt- und Zielsystem. Durch das Objektsystem sind die Abgrenzung von Diskurs- und Umwelt sowie deren Leistungsbeziehungen, durch das Zielsystem Sach- und Formalziele, Strategien und Rahmenbedingungen zur Realisierung beschrieben.
- **2. Ebene** (Ferstl & Sinz 2013, S. 198): Auf der zweiten Ebene erfolgt die Spezifikation des Geschäftsprozessmodells anhand von Interaktionsschema (IAS) und Vorgang-Ereignis-Schema (VES). Das IAS beschreibt dabei die Struktur, das VES das Verhalten eines Geschäftsprozesses. Die Struktur wird durch betriebliche Objekte sowie deren Transaktionen zur Interaktion beschrieben. Das Verhalten wird dazu konsistent in Form von Vorgangstypen und deren Ereignisbeziehungen erfasst.
- **3. Ebene** (Ferstl & Sinz 2013, S. 198f.): Aus dem Geschäftsprozessmodell leitet sich für zu automatisierende Aufgaben die fachliche AWS-Spezifikation

ab. Als konzeptuelles Objektschema (KOS) wird die objektorientierte Spezifikation der Struktursicht in Form von Objekttypen und Nachrichtenbeziehungen bezeichnet. Das Zusammenwirken zur Realisierung einer betrieblichen Aufgabe wird durch das Vorgangsobjektschema (VOS) spezifiziert.

2.3 Komponentenarchitekturen zur Automatisierung betrieblicher Informationssysteme

Nachfolgend werden Grundlagen von Komponentenarchitekturen zur Automatisierung betrieblicher Informationssysteme eingeführt. Zunächst werden hierzu relevante Begriffe geklärt (Abschn. 2.3.1), bevor die Spezifikation entsprechender AwS betrachtet wird (Abschn. 2.3.2).

2.3.1 Begriffliche Grundlagen zu Komponentenarchitekturen

Der Begriff *Komponentenarchitektur* meint die Architektur eines AwS (Software-Architektur), die entlang der Grundsätze des komponentenorientierten Paradigmas spezifiziert ist. Nachfolgend werden zunächst die einzelnen Begriffsbestandteile *Software-Architektur* und *Komponente* eingeführt.

Der Begriff der Software-Architektur

Im Unternehmenskontext unterstützen AwS bei der Erreichung betrieblicher Zielsetzungen. Der Architektur des AwS kommt dabei eine zentrale Rolle zu. Sie kann als Brücke zur Realisierung z.T. abstrakter Ziele durch konkrete Systeme verstanden werden und dient als Hilfsmittel zur Beherrschung der Komplexität der Entwicklung (Bass et al. 2013, S. 3). Unter einer *Software-Architektur* wird die grundlegende Organisation, also der Bauplan eines AwS, verstanden. Dies umfasst die einzelnen Komponenten (i.S.v. Bestandteil) des Systems mit deren Beziehungen untereinander sowie zur Umwelt. Eine *Software-Architektur* folgt dabei bestimmten Konstruktionsregeln und Prinzipien, über die der grundlegende Entwurf bestimmt ist (Reussner & Hasselbring 2009, S. 1, Sinz 2002a). Als Beispiel können verteilte Systeme genannt werden, deren autonome Komponenten basierend auf dem Prinzip der losen Kopplung hinsichtlich der Erreichung gemeinsamer Ziele kommunizieren (Sinz 2002a). Gerade die Realisierung nicht-fachlicher Anforderungen (z.B. Performanz, Wartbarkeit, Verlässlichkeit, Verteilbarkeit) hängt

maßgeblich von Architekturentscheidungen bzw. -formen ab (Bosch 2000, S. 11). Software-Architekturen können hinsichtlich des unterstützten Abstraktionsgrades unterschieden werden (Sommerville 2016, S. 169): Die *Feinarchitektur* bezieht sich im Wesentlichen auf die Strukturierung der Realisierung von Teilsystemen. Die *Grobarchitektur* beschreibt hingegen die Strukturierung, Verteilung und Interaktion eines komplexen AwS.

Basierend auf Sommerville (2016, S. 169) und Bass et al. (2013, S. 328f.) können der Nutzen und die Funktion einer dokumentierten Software-Architektur unter nachfolgenden Punkten zusammengefasst werden:

- Die Architektur kann als *Kommunikationsgrundlage* zwischen verschiedenen Projektbeteiligten (*Stakeholdern*) eingesetzt werden.
- Die Architektur stellt die *Grundlage zur Systemanalyse* und den *Ausgangspunkt der Entwicklung* dar und macht Vorgaben zur Implementierung.
- Die Dokumentation der Architektur kann als *Schulungsgrundlage* für neue Mitarbeiter auf Entwickler- als auch auf Nutzerseite dienen.
- Die Architektur eines AwS kann *Grundlage zur Wiederverwendung* sein. Die Architektur bzw. Teile davon können auf zu entwickelnde Systeme mit vergleichbaren Anforderungen übertragen werden.

Anzumerken ist, dass sich der Detailgrad eines Architekturmodells bzw. der Dokumentation je nach Anwendung unterscheidet (Reussner & Hasselbring 2009, S. 2).

Der Begriff der Komponente

In der vorliegenden Arbeit wird eine Architektur auf Basis des komponentenorientierten Paradigmas vorgestellt. Die Begründung des gewählten Paradigmas erfolgt im Rahmen der Konzeption in Kap. 5. Die Grundbegriffe werden im nachfolgenden Abschn. vorgestellt. Der Begriff der *Komponente* ist dabei von zentraler Bedeutung. Zunächst wird dieser allgemein eingeführt, bevor die Spezialisierung der *Fachkomponente* erläutert wird.

Software-Komponente Bei Ackermann et al. (2002) wird der Begriff der (Software-) Komponente wie folgt definiert: „*Eine Komponente besteht aus verschiedenar-*

tigen (Software-) Artefakten. Sie ist wiederverwendbar, abgeschlossen und vermarktbar, stellt Dienste über wohldefinierte Schnittstellen zur Verfügung, verbirgt ihre Realisierung und kann in Kombination mit anderen Komponenten eingesetzt werden, die zur Zeit der Entwicklung nicht unbedingt vorhersehbar ist.“ Eine Komponente wird zudem ausschließlich als Einheit verwendet und verfügt über explizit spezifizierte Kontextabhängigkeiten (Szyperski et al. 2002, S. 41). Unter Kontextabhängigkeiten werden sowohl Abhängigkeiten zu anderen Komponenten als auch zum zu Grunde liegenden Komponentenmodell und zur Laufzeitumgebung verstanden (Szyperski et al. 2002, S. 44). Der Bezug zu Komponentenmodell und Laufzeitumgebung (als *Socket* bezeichnet) wird auch bei Herzum & Sims (2000, S. 7) genannt. Der eigentliche Nutzen einer Komponente entsteht durch Komposition und Kollaboration mit anderen Komponenten zur Realisierung einer höheren Funktionalität (Herzum & Sims 2000, S. 6). Als zentrale Eigenschaft hierzu wird von Hopkins (2000) die Trennung von Schnittstelle und Implementierung hervorgehoben. Über das Konzept der Schnittstelle wird die Funktionalität einer Komponente beschrieben und nach außen angeboten. Eine Komponente kann mehrere Schnittstellen besitzen, die mit den verschiedenen Zugriffspunkten einer Komponente korrespondieren (Szyperski et al. 2002, S. 43). Schnittstellenspezifikationen werden auch als Nutzungsvertrag zwischen Client und Komponente eingeordnet. Hierzu sind neben Festlegung der Funktionalität (einschl. Parameter) weitere Spezifikationen vorzunehmen (Szyperski et al. 2002, S. 53f.): Dazu zählen vom Client zu erfüllenden Vorbedingungen sowie Nachbedingungen seitens der Komponente. Hinzu kommen Versionsinformationen sowie zugesicherte qualitative Eigenschaften der Komponente wie Durchsatz, Datensicherheit, Performance, Verfügbarkeit und durchschnittliche Wartezeiten.

Fachkomponente Im Kontext betrieblicher Informationssysteme ist die Funktionalität einer Komponente häufig aufgrund fachlicher anstatt rein technischer Aspekte festgelegt (Birkmeier & Overhage 2012b). Als *Fachkomponente* wird eine Komponente verstanden, „(...) die eine bestimmte Menge von Diensten einer betrieblichen Anwendungsdomäne anbietet“ (Ackermann et al. 2002). Durch die Komponente wird ein abgeschlossenes Funktionsbündel zur Durchführung einer spezifischen fachlichen Aufgabe, basierend auf einem technologischen Framework, angeboten. Sie stellt die Implementierung eines autonomen fachlichen Konzeptes oder Prozesses dar (Birkmeier & Overhage 2012b, basierend auf Baster et al. 2001

u. Herzum & Sims 2000, S. 40). Nach Herzum & Sims (2000, S. S. 118ff.) besitzt die innere Struktur einer Fachkomponente eine oder mehrere der nachfolgenden vier logisch abgrenzbaren Ebenen: Innerhalb der *User*-Ebene werden Elemente und deren Darstellung zur Nutzerinteraktion beschrieben. Auf *Workspace*-Ebene wird die lokale Komponentenlogik zur Unterstützung der Nutzerinteraktion beschrieben. Die globale fachliche Logik mit Zugriff auf fachliche Ressourcen ist der *Enterprise*-Ebene zugeordnet. Der physische Zugriff auf diese Ressource erfolgt über die sog. *Resource*-Ebene. Nach Herzum & Sims (2000, S. S. 173ff.) lassen sich drei wesentliche Kategorien von Fachkomponenten unterscheiden: *Process Business Components* sind für die Ausführung von Prozessen oder von Teilen dieser verantwortlich. Die Aufgabe von *Entity Business Components* ist die Verwaltung (Zugriffsoperationen und Wartung) von fachlichen Konzepten und Entitäten. Sonstige übergreifend benötigte unterstützende Konzepte werden durch *Utility Business Components* zur Verfügung gestellt.

Sowohl bei der Bestimmung des Begriffs der *Software-Komponente* als auch der *Fachkomponente* anhand der Literatur wird der Bezug zur Laufzeitumgebung thematisiert. Bzgl. des Untersuchungsproblems der Arbeit wird davon abstrahiert. Der Fokus liegt auf konzeptueller Ebene und untersucht die fachliche Abgrenzung und Interaktion von Komponenten.

2.3.2 Grundlagen zur Spezifikation von Komponentenarchitekturen

Nachfolgend werden Grundlagen zur Spezifikation von komponentenorientierten AwS eingeführt. Zunächst werden generelle Eigenschaften dieser Systeme aufgezeigt, bevor ein idealtypisches Vorgehen zur Komponentenentwicklung, grundlegende Überlegungen zur Granularität sowie das Komponentendiagramm der Unified Modeling Language (UML) vorgestellt werden.

Eigenschaften der komponentenorientierten Software-Entwicklung

Zentrale Treiber der komponentenorientierten Software-Entwicklung stellen *Evolution* und *Wiederverwendung* dar (Hopkins 2000). Bei Sharp & Ryan (2005) wird zudem die Eigenschaft der *Portierbarkeit* ergänzt. Cheesman & Daniels (2001, S. 2) heben die *Flexibilität* (Evolutionsfähigkeit) als eigentlichen Treiber der komponentenorientierten Softwareentwicklung nochmals von den genannten Eigenschaf-

ten ab. Auch bei Sharp & Ryan (2005) wird explizit die Entwicklung hoch flexibler Systeme genannt, um Herausforderungen sich dynamisch verändernder Geschäftsprozesse gewachsen zu sein. Komponentenorientiert entwickelte AwS können als hochgradig evolutionsfähig (*highly evolvable systems*) charakterisiert werden (Herzum & Sims 2000, S. 55). Durch die Integration vordefinierter und wiederverwendbarer Softwarekomponenten können Kosten-, Zeit- und Wartungsaufwand im Vergleich zu objektorientierten oder auch strukturierten Entwicklungsansätzen reduziert werden (Sharp & Ryan 2005). Nach Szyperski et al. (2002, S. S. 4ff.) werden, ein funktionierender Marktplatz für Komponenten vorausgesetzt, durch die kombinierte Nutzung von Standardkomponenten und Eigenentwicklungen konfigurierbare AwS mit überschaubaren Risiken (z.B. finanzielle und zeitliche Risiken) und flexibler Anpassbarkeit ermöglicht. Durch eingekaufte Standardkomponenten werden Risiken sowie Aufwände für Anpassung und Wartung an den Komponentenhersteller übertragen. Die Nutzung eigenentwickelter Komponenten im Zielsystem ermöglicht die flexible Ausrichtung auf individuelle Anforderungen und Rahmenbedingungen (Szyperski et al. 2002, S. 4ff.).

Vorgehen zur komponentenorientierten Softwareentwicklung

Cheesman & Daniels (2001) stellen ein idealtypisches Vorgehen zur Entwicklung komponentenorientierter AwS vor. Der in der Praxis durchaus verbreitete Entwicklungsprozess (Birkmeier & Overhage 2012b) ist in Abb. 2.7 dargestellt. Als idealtypischer Prozess wird von mehrfachen Iterationen während der Entwicklung abstrahiert. In der ersten Phase erfolgt die Spezifikation fachlicher Anforderungen (*Requirements Definition*). Als Resultate der Anforderungsanalyse werden konzeptuelle Modelle zur fachlichen Domäne sowie Anwendungsfälle zur Spezifikation funktionaler Anforderungen vorgeschlagen. Die Phase zur Spezifikation (*Specification*) der Komponentenarchitektur erfolgt in den drei Schritten *Component Identification*, *Component Interaction* und *Component Specification* (Cheesman & Daniels 2001, S. 27). Ausgehend von einer fachlichen Spezifikation der Komponentenarchitektur wird mit jedem Schritt des Entwicklungsvorgehens der technische Detaillierungsgrad erhöht. Im Rahmen der *Component Identification* erfolgt ein initiales Architekturmodell, das Komponenten und deren Schnittstellen überwiegend auf Basis der fachlichen Anforderungen der *Requirements Definition* beschreibt. Mit der *Component Interaction* wird eine detaillierte Sicht auf die initiale Architektur eingenommen und diese ggf. unter Berücksichtigung der techni-

schen Realisierung restrukturiert. Zudem werden Abhängigkeiten zwischen Komponenten bzw. deren Schnittstellen spezifiziert. Sofern eine stabile Architektur vorliegt, werden Schnittstellenspezifikationen im Schritt *Component Specification* weiter präzisiert. Als Beispiele sind die Definition der vollständigen Signaturen von Operatoren, Vor- und Nachbedingungen zu deren Ausführung sowie die möglichen Zustände der verwalteten Objekte zu nennen (Cheesman & Daniels 2001, S. 33ff.). Gegenstand der Phasen *Provisioning* und *Assembly* sind die Bereitstellung der implementierten Komponenten sowie deren Integration zu einem ausführbaren AwS (Cheesman & Daniels 2001, S. 27).

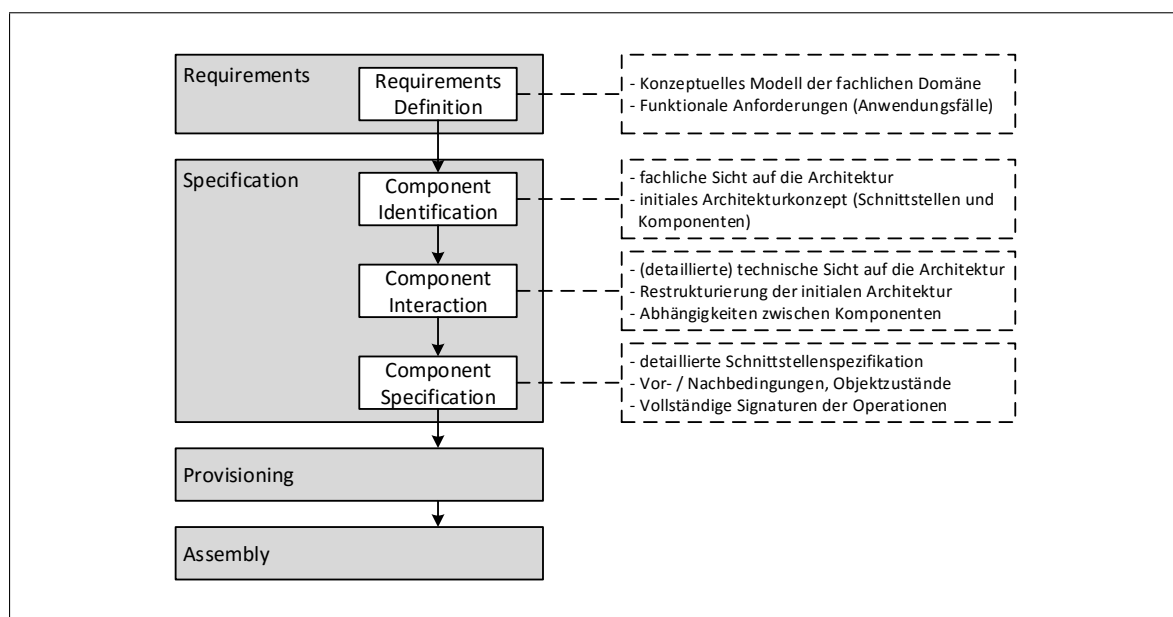


Abb. 2.7: Idealtypische Entwicklung komponentenorientierter Anwendungssysteme (Cheesman & Daniels 2001, S. 27)

Granularität von Komponenten

Die Bestimmung der Granularität von Komponenten stellt eine zentrale Fragestellung bei der Entwicklung komponentenorientierter AwS dar. *Kopplung* (*Coupling*) und *Kohäsion* (*Cohesion*) sind dabei die entscheidenden Faktoren (Hopkins 2000). Unter *Kopplung* wird der Grad der Beziehung von Klassen innerhalb einer Komponente zu Klassen außerhalb dieser Komponente verstanden. *Kohäsion* ist hingegen über den Bezug der Klassen innerhalb einer Komponente zueinander definiert (Vitharana et al. 2004).

Je weiter Komponenten gefasst werden, desto komplexer gestalten sich deren Schnittstellen und desto häufiger werden diese zum Gegenstand von Anpassungen. Komponenten mit zu weit gefasstem Funktionsumfang schränken folglich die Flexibilität eines Gesamtsystems ein (Hopkins 2000). Die Spezifikation von Komponenten mit eher eng gefasstem Funktionskontext und Auslagerung von Sekundärfunktionen führt zwar zu einer Steigerung der Flexibilität des Systems, gleichzeitig aber auch zu zunehmenden Abhängigkeiten zwischen Komponenten (Hopkins 2000, Szyperski et al. 2002, S. 45f.). Dies führt zu einer zeit- und ressourcenintensiven Kommunikation der Komponenten (Hopkins 2000). Mit Blick auf die *Vermarktbarkeit* stellen komplexe Abhängigkeiten auch eine Einschränkung der Anwendbarkeit einer Komponente in heterogenen Szenarien dar (Szyperski et al. 2002, S. S. 45f.).

Grundsätzlich gilt es die Balance zwischen möglichst niedriger Kopplung zwischen Komponenten und möglichst hoher Kohäsion innerhalb der Komponenten zu finden. Die *richtige* Balance ist im Einzelfall festzulegen und hängt u.a. von den konkreten Anforderungen an Abstraktionsgrad, Flexibilität oder Komplexität der Komponenten ab (Hopkins 2000). Als zentrales Prinzip der komponentenorientierten Softwareentwicklung gilt also *Maximierung der Kohäsion bei gleichzeitiger Minimierung der Kopplung von Komponenten (maximize cohesion and minimize coupling)*. Das Prinzip geht, wie auch andere Prinzipien der komponentenorientierten Entwicklung, auf Kriterien zur Modularisierung von Systemen nach Parnas (1972) zurück (Szyperski et al. 2002, S. 40). Ein Vorschlag von Birkmeier & Overhage (2012b) lautet, zur Minimierung der Abhängigkeiten Funktionen und Datenobjekte, die zur Laufzeit gemeinsam verwendet werden, in einer Komponente zusammenzufassen. Zur Maximierung der Kohäsion sollten auch nur diese gemeinsam verwendeten Funktionen und Datenobjekte eine Komponente bilden.

Bei Vitharana et al. (2004) sowie darauf basierend bei Birkmeier & Overhage (2012b) werden Auswirkungen von Kopplung und Kohäsion auf übergeordnete Ziele beleuchtet. Diese sind in Abb. 2.8 dargestellt und werden nachfolgend zusammenfassend erläutert:

- **Kosteneffektivität:** Die Steigerung der Kopplung bzw. der Abhängigkeiten zwischen Komponenten führt notwendigerweise zu weiteren und komplexeren Schnittstellen. Ein höherer Entwicklungsaufwand ist nach Birkmei-

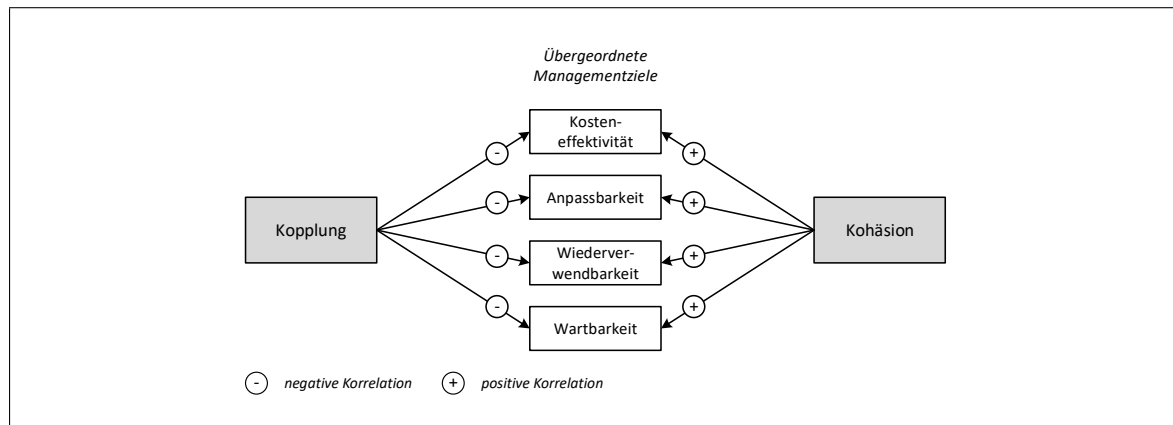


Abb. 2.8: Auswirkungen der Faktoren Kohäsion und Kopplung (Birkmeier & Overhage 2012b) (eigene Darstellung)

er & Overhage (2012b) die Folge. Die Zusammenfassung zusammengehöriger Elemente zu einer Komponente reduziert gleichzeitig die Kopplung zwischen diesen und führt zu einem positiven Effekt bzgl. der Kosteneffektivität (Birkmeier & Overhage 2012b), wenngleich auch bei Vitharana et al. (2004) darauf hingewiesen wird, dass mit zunehmender Größe und Komplexität der inneren Strukturen einer Komponente die Entwicklungskosten steigen.

- **Anpassbarkeit des Anwendungssystems:** Unter *Anpassbarkeit* wird die Eigenschaft eines AwS (nicht einzelner Komponenten) verstanden, durch die Kombination unabhängig voneinander entwickelter Komponenten Anforderungen des Gesamtsystems zu realisieren. Nach Vitharana et al. (2004) steigt mit zunehmender Kohäsion der Funktionen einer Komponente auch die Wahrscheinlichkeit, dass durch die Komponente zusammengehörige Anforderungen des Ziel-AwS realisiert werden. Dadurch wird ein positiver Effekt der Kohäsion auf die *Anpassbarkeit* des Gesamtsystems erwartet. Durch starke Kopplung zwischen Systembestandteilen wird die Anpassung des Gesamtsystems hingegen erschwert (Vitharana et al. 2004).
- **Wiederverwendbarkeit:** Mit zunehmender Kopplung von Komponenten wird die Wiederverwendbarkeit in anderen Anwendungsszenarien eingeschränkt (Vitharana et al. 2004). Szyperski et al. (2002, S. S. 45f.) stützen dieses Argument und prognostizieren eine bessere Vermarktbarkeit für größere Komponenten. Stark gekoppelte Komponenten sind i.d.R. auch in heterogenen Anwendungsszenarien gemeinsam zu verwenden (Birkmeier & Overhage

2012b). In sich geschlossene Komponenten mit hoher Kohäsion der Funktionen und der realisierten Anforderungen sind aufgrund weniger Abhängigkeiten besser zur Wiederverwendung geeignet (Szyperski et al. 2002, S. 45f., Vitharana et al. 2004).

- **Wartbarkeit:** Starke Kopplung beeinflusst die Wartbarkeit eines komponentenorientierten AwS negativ. Änderungen an einer Komponente (z.B. deren Schnittstelle) betreffen mit erhöhter Wahrscheinlichkeit auch davon abhängige Komponenten (Vitharana et al. 2004). Zwar wird von Vitharana et al. (2004) auch ein hoher Wartungsaufwand für Komponenten mit hoher Kohäsion angenommen, nach Birkmeier & Overhage (2012b) ist aus Sicht des Gesamtsystems der Wartungsaufwand einzelner im Vergleich zu stark gekoppelten Komponenten niedriger.

Das Komponentendiagramm der Unified Modeling Language

Zur Spezifikation der Struktur einer Komponentenarchitektur wird in der vorliegenden Arbeit das Komponentendiagramm der UML (OMG 2015, S. 206ff.) eingesetzt. Die Ausführungen dazu beschränken sich auf für die Arbeit notwendigen Aspekte. Das im vorherigen Abschnitt vorgestellte Komponentenverständnis wird auch durch die UML-Spezifikation unterstützt. Eine Komponente wird als autonome Einheit verstanden, die ihre Funktionalität über eine bzw. mehrere wohl-definierte Schnittstellen anbietet. Zur Realisierung der Funktionalität kann eine Komponente auch andere Komponenten über deren Schnittstellen nutzen. Durch das Konzept der Schnittstelle ist eine Komponente innerhalb einer definierten Umgebung flexibel austauschbar und wiederverwendbar (OMG 2015, S. 206).

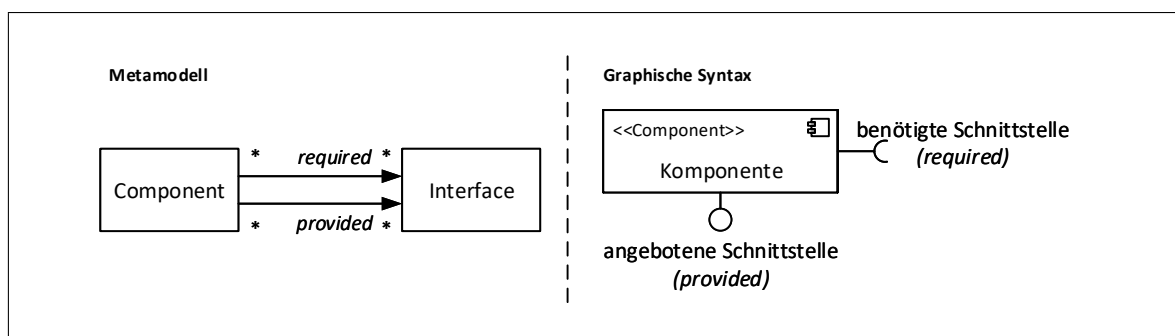


Abb. 2.9: Spezifikation von Komponenten mit der UML basierend auf OMG (2015, S. 206ff.)

In Abb. 2.9 ist das Konzept zur Modellierung von Komponenten (Stereotyp <<Component>>) aus Außensicht mit Metamodell (Verkürzung bzgl. relevanter Bestandteile) und graphischer Syntax dargestellt. Angebotene Schnittstellen (*provided*) einer Komponente werden über die *Lollipop*-Notation, benötigte Schnittstellen über *Sockets* modelliert (OMG 2015, S. 208f.). Schnittstellen können über Bezeichner und öffentliche Eigenschaften bzw. Operationen einer Komponente näher beschrieben sowie über *Ports* logisch strukturiert werden (OMG 2015, S. 208). Die Spezifikation von Abhängigkeiten zwischen Komponenten erfolgt durch die Verknüpfung kompatibler benötigter und angebotener Schnittstellen. Neben der Funktionalität können zur Bestimmung der Kompatibilität auch nicht-fachliche Anforderungen, wie beispielsweise Performance-Anforderungen, relevant sein (OMG 2015, S. 208). Für alternative bzw. weiterführende Notationen sowie für Konzepte zur Modellierung der Innensicht von Komponenten wird auf die Literatur, primär auf die zugehörige Spezifikation OMG (2015, S. 206ff.), verwiesen.

3 Methodische Grundlagen zu prozessorientierten Data-Warehouse-Systemen

Kapitelzusammenfassung: Zunächst wird der Begriff des *dispositiven Anwendungssystems* allgemein abgegrenzt und im Anschluss das Konzept *Data-Warehouse-System* als solches eingeordnet (Abschn. 3.1). DWH-Systeme stellen eine etablierte Klasse von Anwendungssystemen dar. Die zugehörigen Grundlagen zu Architektur und Datenmodell werden in Abschn. 3.2 sowie Abschn. 3.3 geschaffen. Zur Modellierung multidimensionaler Datenstrukturen wird in der Arbeit das *Semantische Data-Warehouse-Modell* (SDWM) verwendet. Dieses wird anhand seines Metamodells in Abschn. 3.4 eingeführt. Im Mittelpunkt des Untersuchungsproblems der Arbeit steht das prozessorientierte DWH-System. Dieses wird in Bezug auf das klassische DWH-Verständnis abgegrenzt und anhand möglicher Anwendungsszenarien als Spezialisierung vorgestellt (Abschn. 3.5).

Abschn. 3.1.2, 3.2.1, 3.3 u. 3.4 basieren in Teilen auf der Diplomarbeit Benker (2010).

3.1 Grundlagen dispositiver Anwendungssysteme

Abschn. 3.1 bildet die Basis nachfolgender Abschnitte, die spezifische Grundlagen zu DWH-Systemen schaffen. Hierzu wird zunächst der Begriff des *dispositiven Anwendungssystems* eingeführt (Abschn. 3.1.1) und das Konzept *DWH-System* als solches eingeordnet (Abschn. 3.1.2).

3.1.1 Bestimmung des Begriffs des dispositiven Anwendungssystems

Nach Kemper et al. (2010, S. 15f.) basiert das Konzept des *dispositiven Anwendungssystems* auf dem Begriff der *dispositiven Arbeitsleistung* nach Gutenberg (1983, S. 3). Demnach handelt es sich um dispositive Leistungen, wenn die Arbeiten „(...)

mit der Leitung und Lenkung der betrieblichen Vorgänge in Zusammenhang stehen“ (Gutenberg 1983, S. 3). AwS zur Unterstützung der dispositiven Leistung werden auch als *dispositive Systeme* (Kemper et al. 2010, S. 15) bezeichnet. Auf Grundlage dieser ersten Begriffsnäherung kann, bei phasenorientierter Systemabgrenzung nach Ferstl & Sinz (1995) (vgl. Kap. 2.1.1), ein dispositives System als Aufgabenträger des *Lenkungssystem* (mit den Aufgaben Planung, Steuerung und Kontrolle) verstanden werden. Als *operative Anwendungssysteme* werden hingegen, z.B. bei Kemper et al. (2010, S. 15), AwS verstanden, die unmittelbar die Leistungserstellung bzw. -verwertung unterstützen. Als solches werden diese dem *Leistungssystem* zugeordnet. Eine Abgrenzung beider Systemarten anhand der Ausrichtung sowie der Eigenschaften ihrer Datenstrukturen wird, basierend auf Kemper et al. (2010, S. 16) und Christmann (1996, S. C822.07), in Tab. 3.1 gezeigt. Während die operativen Systeme auf die Abwicklung von Geschäftsprozessen ausgerichtet sind und auf einer detaillierten, transaktionsorientierten und zeitpunktbezogenen Datenbasis aufsetzen, zeichnet sich die dispositive Datenbasis durch transformierte, themenbezogene und verdichtete Datenstrukturen zur Managementinformation und Entscheidungsunterstützung aus.

	Charakteristika operativer AwS	Charakteristika dispositiver AwS
Ausrichtung d. AwS	Abwicklung von Geschäftsprozessen	Managementinformation; Entscheidungsunterstützung
Granularität	detaillierte Daten zu Geschäftsvorfällen	i.d.R. verdichtete u. transformierte Daten
Zeitbezug	aktueller, zeitpunktbezogener Datenbestand	historisierter Datenbestand; aufgabenabh. Aktualität
Datenstrukturierung	funktionsorientiert; abgeleitet aus dem Geschäftsprozess	themenorientiert; Berücksichtigung des Endbenutzers
Datenaktualisierung	laufend u. konkurrierend	ständige Erweiterung des Datenbestandes
Anfragen	strukturiert	ad-hoc Fragestellungen u. Standardauswertungen

Tab. 3.1: Charakterisierung operativer und dispositiver Anwendungssysteme basierend auf Kemper et al. (2010, S. 16) und Christmann (1996, S. C822.07)

Im Zeitverlauf entwickelten sich zahlreiche Systeme zur Unterstützung des Managements eines Unternehmens bei der Durchführung dispositiver Aufgaben. Die typischerweise genannten Systemformen werden basierend auf Gluchowski et al. (2008, S. 55ff.) nachfolgend kurz vorgestellt:

- **Management Information Systeme (MIS):** Ein Management Information System (MIS) dient der Erstellung standardisierter, periodischer Berichte für Führungskräfte verschiedener Hierarchieebenen. Damit ist das vergangenheitsbezogene Monitoring von Geschäftsaktivitäten möglich. Einblicke in den operativen Datenbestand können sowohl detailliert als auch verdichtet vorgenommen und extrahiert werden. Aus der Historie betrachtet stellen MIS einen ersten Schritt zur Unterstützung dispositiver Aufgaben auf Führungsebene dar (Gluchowski et al. 2008, S. 55f.).
- **Decision Support Systeme (DSS):** In einem DSS bzw. Entscheidungsunterstützungssystem wird das Verhalten von Fach- und Führungskräften bei der Lösung fachlicher Probleme abgebildet. DSS stellen interaktive Systeme dar, die basierend auf Modellen, Methoden und problembezogenen Daten Nutzer im Entscheidungsprozess unterstützen (Gluchowski et al. 2008, S. 63). DSS sind auf ein spezifisches Problem oder eine Klasse von Aufgaben ausgerichtet (Mertens & Meier 2009, S. 12). Die Anwendung ist sowohl bei strukturierten aber auch semi- und unstrukturierten Entscheidungssituationen denkbar (Gluchowski et al. 2008, S. 63).
- **Executive Information Systeme (EIS):** Ein EIS (auch *Führungsinformationssystem*) stellt ein interaktives AWS mit flexibel anpassbarer und intuitiver Nutzeroberfläche dar, das (nicht nur) Führungskräften Einblicke in aktuelle interne und externe Informationen ermöglicht. Explorative Datenanalysen basierend auf multidimensionalen Strukturen, Standardreporting sowie Ad-hoc-Abfragen stellen zentrale Charakteristika dar (Gluchowski et al. 2008, S. 74ff.).
- **Executive Support Systeme (ESS):** ESS nutzen Funktionen von EIS zur Aufdeckung grundlegender betrieblicher Zusammenhänge und erweitern diese um Funktionalitäten von DSS. Betriebswirtschaftliche Kausalmodelle sowie Methoden zur Analyse, Simulation, Prognose und Optimierung sind beispielsweise zu nennen. ESS sind auf die ganzheitliche, phasen- und problemübergreifende Unterstützung von Führungskräften ausgerichtet (Gluchowski et al. 2008, S. 82ff.).

3.1.2 Einordnung von Data-Warehouse-Systemen als dispositive Anwendungssysteme

Nachfolgend wird die Rolle von DWH-Systemen anhand zweier unterschiedlicher, dennoch komplementärer Perspektiven im Kontext dispositiver Systeme beleuchtet.

Einordnung von Data-Warehouse-Systemen anhand des Konzeptes der integrierten Informationsverarbeitung

Bei Mertens (2009, S. 6) werden im Konzept der *integrierten Informationsverarbeitung* zwei grundsätzliche Klassen von Anwendungssystemen, *operative Systeme* sowie *Planungs- und Kontrollsysteme*, unterschieden. Der Gesamtzusammenhang ist in Abb. 3.1 dargestellt.

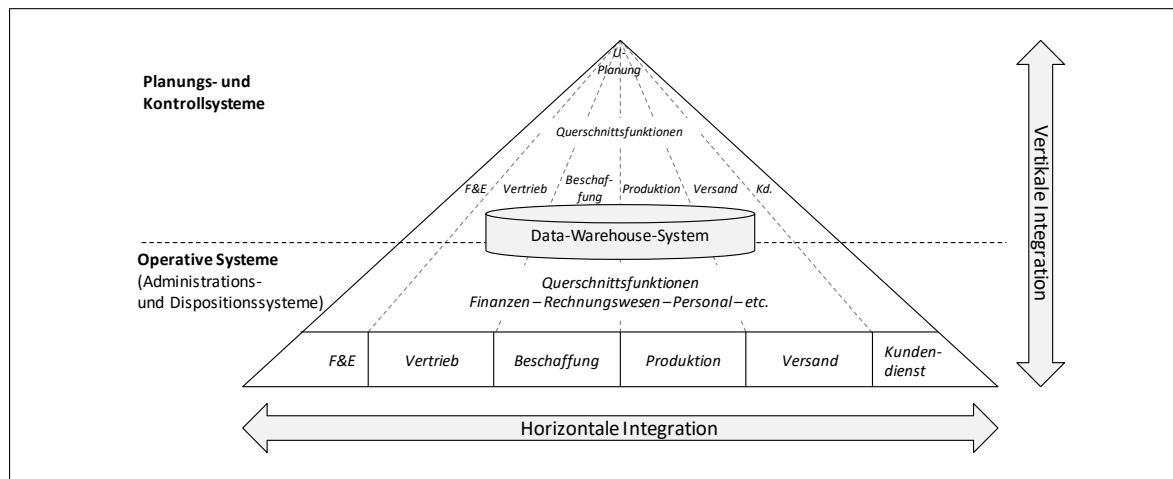


Abb. 3.1: Integrierte Informationsverarbeitung (Mertens 2009, S. 6) (eigene Darstellung)

Als *operative Systeme* werden *Administrations-* und *Dispositionssysteme* unterschieden. Administrationssysteme dienen insbesondere der Kostensenkung und Entlastung menschlicher Aufgabenträger durch die Übernahme von Routineaufgaben (Mertens 2009, S. 13). Sie bilden transaktionsorientiert den Einsatz von Elementarfaktoren einer Unternehmung im Prozess der Leistungserstellung ab (Chamoni & Gluchowski 2016, S. 6). Dispositionssysteme übernehmen darüber hinaus die Entscheidungsunterstützung auf operativer Ebene. Entscheidungen sind entweder teilautomatisiert, werden also automatisiert vorbereitet und personell getroffen, oder vollautomatisiert (Mertens 2009, S. 13). Als Entscheidungsaufgaben des Dispositionssystems werden Steuerungs- und Lenkungsaufgaben mit strukturierter Entscheidbarkeit und Delegationsfähigkeit verstanden (Chamoni &

Gluchowski 2016, S. 6). Der Begriff des *operativen Systems* ist bei Mertens (2009) somit weiter gefasst und umfasst mit Dispositionssystemen auch einen Teil des *Lenkungssystems*.

Planungs- und Kontrollsysteme sind den operativen Systemen übergeordnet. Planungssysteme können als Fortsetzung der Dispositionssysteme verstanden werden, unterscheiden sich jedoch in nachfolgenden Punkten (Mertens 2009, S. 14): Planungssysteme sind auf schlecht strukturierte Entscheidungsprobleme ausgerichtet, die in größeren Zeitabständen und unregelmäßig auftreten. Zielgruppe bildet das höhere Management, das in Mensch-Computer-Interaktion die Lösung der Probleme anstrebt. Als Grundlage dient ein auf die Entscheidung ausgerichteter Datenbestand, wie er beispielsweise durch DWH-Systeme zur Verfügung gestellt wird. Dispositionssysteme bearbeiten hingegen gut strukturierte, z.T. vollautomatisierbare Entscheidungsprobleme auf einem detaillierten, transaktionsorientierten Datenbestand. Ziel- oder Nutzergruppe von Dispositionssystemen bildet das operative/mittlere Management (Mertens 2009, S. 14). *Kontrollsysteme* bilden den komplementären Part von Planungssystemen zur Prüfung der Planeinhaltung (Mertens 2009, S. 14f.).

In Abb. 3.1 wird zudem die *horizontale* und die *vertikale Integration* dargestellt. Die *horizontale* Integration beschreibt dabei im Wesentlichen die Integration der betrieblichen Teilsysteme in der Wertschöpfungskette. Über die *vertikale Integration* wird die Datenversorgung von Planungs- und Kontrollsystemen aus den operativen Systemen beschrieben (Mertens 2009, S. 5). Basierend auf den Ausführungen von Mertens wird die Funktion des DWH-Systems deutlich und kann nach Gluchowski et al. (2008, S. 118) als Bereitstellung entscheidungsrelevanter Inhalte für Planungs- und Kontrollsysteme und damit als Unterstützung dispositiver Aufgaben spezifiziert werden.

Regelkreisorientierte Einordnung von Data-Warehouse-Systemen

Bei Sinz (2002b) wird die Einordnung von DWH-Systemen anhand einer regelkreisorientierten Betrachtung von betrieblichen Systemen vorgenommen. Grundlage dazu bildet die Systemabgrenzung nach dem Phasenprinzip in *Lenkungs- und Leistungssystem* (vgl. Kap. 2.1.1). Durch die regelkreisorientierte Betrachtung lassen sich DWH-Systeme und ihre Funktionen im betrieblichen Kontext beschrei-

ben (vgl. Abb. 3.2). Das DWH-System übernimmt auf Managementebene die Funktion einer Hilfsregelstrecke und dient der Bereitstellung aufbereiteter operativer Daten (in Form von Berichten und Analysen) zur Unterstützung des Managements als *GP-übergreifender* Regler. Als Managementaufgaben werden sowohl die Gestaltung als auch die Lenkung des betrieblichen Systems, insbesondere der Geschäftsprozesse anhand entsprechender Zielvorgaben, verstanden (Sinz 2002b, S. 310f.). Die GP-spezifische Lenkung (*Geschäftsprozessebene*) kann anhand eines weiteren Regelkreises beschrieben und als Regelstrecke dem Regler *Management* untergeordnet werden. Aufgabe der Lenkung von Geschäftsprozessen (*Lenkungssystem, Regler*) ist die Planung, Steuerung und Kontrolle der laufenden Geschäftsprozesse (*Leistungssystem, Regelstrecke*). Als Hilfsregelstrecke dient der operative Datenbestand. Während Daten in DWH-Systemen in hierarchischen Klassifikationsstrukturen aufbereitet zur interaktiven Analyse vorliegen, bildet der operative Datenbestand GP-Transaktionen sowie deren Zustand detailliert ab (Sinz 2002b, S. 310f.).

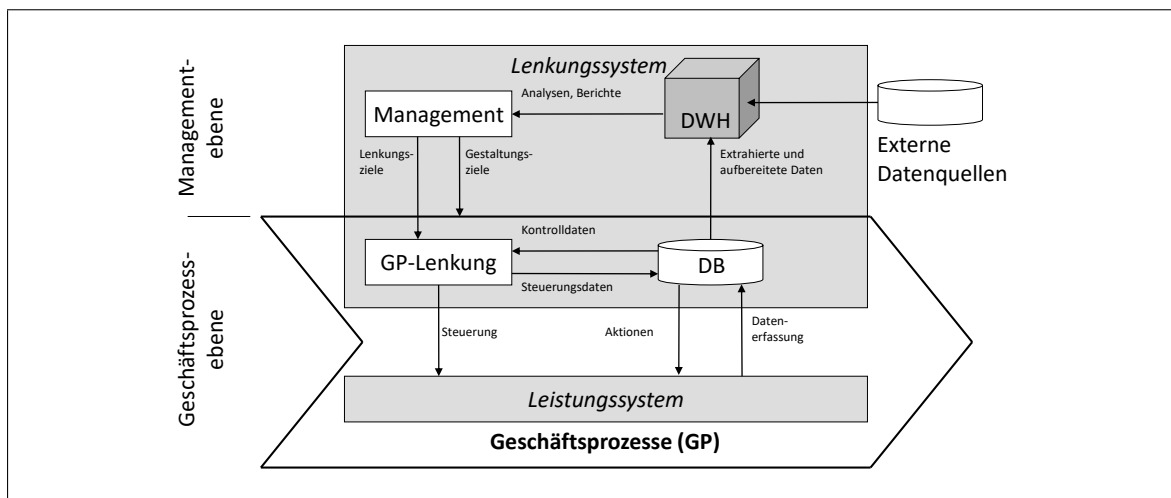


Abb. 3.2: Regelkreisorientierte Einordnung des Data-Warehouse-Systems (Sinz 2002b, S. 310)

Eigenschaften eines Data-Warehouse-Systems als dispositives Anwendungssystem

Nach Abgrenzung des DWH-Konzeptes werden zentrale Eigenschaften hinsichtlich der identifizierten Funktionen (Datenbereitstellung im Rahmen der vertikalen Integration sowie Hilfsregelstrecke im betrieblichen Regelkreis) identifiziert. Hierzu kann die häufig zitierte Definition eines DWH-Systems nach Inmon (2005, S. 29ff.) herangezogen werden: „A data warehouse is a subject oriented, integra-

ted, non-volatile, and timevariant collection of data in support of management's decisions.“ Ein DWH-System ist demzufolge durch die Eigenschaften *Themenorientierung (subject-orientied)*, *Datenintegration (integrated)*, *Beständigkeit (non-volatile)* und *Zeitorientierung (time-variant)* charakterisiert (z.B. Gluchowski et al. 2008, S. 119f., Hahne 2014, S. 1f.):

- **subject-oriented:** Während sich operative Datenbanken in Aufbau und Inhalt an GP-Transaktionen orientierten, sind DWH-Systeme auf entscheidungsrelevante betriebliche Themen ausgerichtet, d.h. der Datenbestand orientiert sich am Informationsbedarf der Entscheidungsträger. Daten operativer Systeme ohne Entscheidungsrelevanz werden nicht ins DWH-System übernommen.
- **integrated:** Als Datenquellen für ein DWH-System sind heterogene, sowohl interne (operative) als auch externe Quellen zu nennen (vgl. Abb. 3.2). Die gewonnenen Daten sind beim Laden ins *DWH* von semantischen und syntaktischen Inkonsistenzen zu bereinigen und hinsichtlich eines stimmigen und akzeptablen, also konsistenten Datenbestandes bzgl. Struktur, Formaten, Begriffen, Codierung und Maßeinheiten zu vereinheitlichen.
- **non-volatile:** Die Inhalte eines DWH-Systems werden zur Analyse im Zeitverlauf ständig erweitert. Datenzugriffe erfolgen in aller Regel lesend, nur in Ausnahmefällen werden Inhalte wieder gelöscht. Einmal auf Grundlage des DWH-Systems erstellte Berichte bleiben damit reproduzierbar.
- **time-variant:** Zur Unterstützung von Entscheidungen auf Managementebene werden Daten benötigt, die die Unternehmensentwicklung im Zeitverlauf darstellen. Für die periodisch erweiterten Inhalte eines DWH-Systems wird explizit ein Zeitbezug definiert und der Datenbestand historisiert.

Das Konzept des *Online Analytical Processing (OLAP)* beschreibt die Leitlinie bzw. das Nutzungsprinzip von Führungsinformationssystemen, das bewusst in Abgrenzung zum *Online Transaction Processing (OLTP)*, dem Nutzungsprinzip operativer AwS spezifiziert wurde (Chamoni & Gluchowski 2010, S. 199). Auch ein DWH-System folgt diesem Nutzungsprinzip, das sich durch die interaktive Analyse multidimensionaler Strukturen, also die Analyse quantitativer Kennzahlen entlang qualitativer und hierarchischer Klassifikationsstrukturen auszeichnet (Sinz

2002b, S. 312). Diese entscheidungsorientierte Sichtweise auf betriebswirtschaftliche Sachverhalte gilt als zentrales Charakteristikum von OLAP- und DWH-Systemen und soll die flexible Analyse komplexer Sachverhalte sowie ad-hoc-Abfragen mit geringem Aufwand ermöglichen. Als bildliche Beschreibung wird häufig die Metapher eines mehrdimensionalen Würfels aufgegriffen (Hahne 2014, S. 4). Eine detaillierte Einführung des multidimensionalen Datenmodells erfolgt in Abschn. 3.3. Bei Codd et al. (1993) wurden initial, einschließlich der Multidimensionalität, zwölf Regeln zur Evaluation von OLAP-Werkzeugen vorgeschlagen. Diese wurden im Laufe der Zeit kritisch diskutiert und erweitert. Als Beispiel kann der FASMI-Ansatz (*Fast, Analysis, Shared, Multidimensional, Information*) nach Pendse & Creeth (1995) genannt werden. Für weitere Informationen sowie eine detaillierte Vorstellung und Diskussion der Regeln wird beispielsweise auf Bauer & Günzel (2013) oder Hahne (2014, S. 4f.) verwiesen.

3.2 Architektur von Data-Warehouse-Systemen

Zunächst werden der grundlegende Aufbau sowie die Funktionsweise von DWH-Systemen erläutert (Abschn. 3.2.1). In Abschn. 3.2.2 wird zudem ein generisches Schichtenmodell zur funktionsorientierten Beschreibung von DWH-Architekturen eingeführt.

3.2.1 Einführung der Architektur von Data-Warehouse-Systemen

In Abb. 3.3 ist die *idealtypische* logische Architektur eines DWH-Systems nach Böhnlein & Ulbrich-vom Ende (2000, S. 17ff.) dargestellt: Anhand dreier vertikal angeordneter Ebenen werden die Funktionen eines DWH-Systems eingeordnet. Auf *Datenerfassungsebene* wird der *ETL-Prozess* mit den Phasen *Extraktion*, *Transformation* und *Laden* angesiedelt. Nach Extraktion (Filterung) von relevanten Daten bzw. von deren Änderungen aus heterogenen Quellsystemen werden die Daten syntaktisch und semantisch in der *Transformationsphase* bereinigt und aufbereitet (Böhnlein & Ulbrich-vom Ende 2000, S. 19f.). Als Quellsysteme können sowohl unternehmensinterne Datenbanken operativer AwS als auch unternehmensexterne Datenquellen dienen (Golfarelli & Rizzi 2009, S. 8). Die *Ladephase* beschreibt schließlich das Laden der Daten in das *Data-Warehouse* auf *Datenhaltungsebene* (Böhnlein & Ulbrich-vom Ende 2000, S. 18ff.). Mit dem Begriff *Data-Warehouse* wird die eigentliche Datenbasis bezeichnet. Ein DWH-System

umfasst zusätzlich das *Data-Warehouse-Management-System* mit den Funktionen der übrigen Ebenen sowie zu deren Administration. Auf *Datenbereitstellungsebene* werden durch die dort angesiedelten OLAP-Server Daten in multidimensionaler Form über entsprechende Schnittstellen nachgelagerten Anwendungssystemen und Präsentationswerkzeugen angeboten. Über alle Ebenen hinweg sind *Administrationsfunktionen* auf Basis von Metadaten vorgesehen (Böhnlein & Ulbrich-vom Ende 2000, S. 17f.).

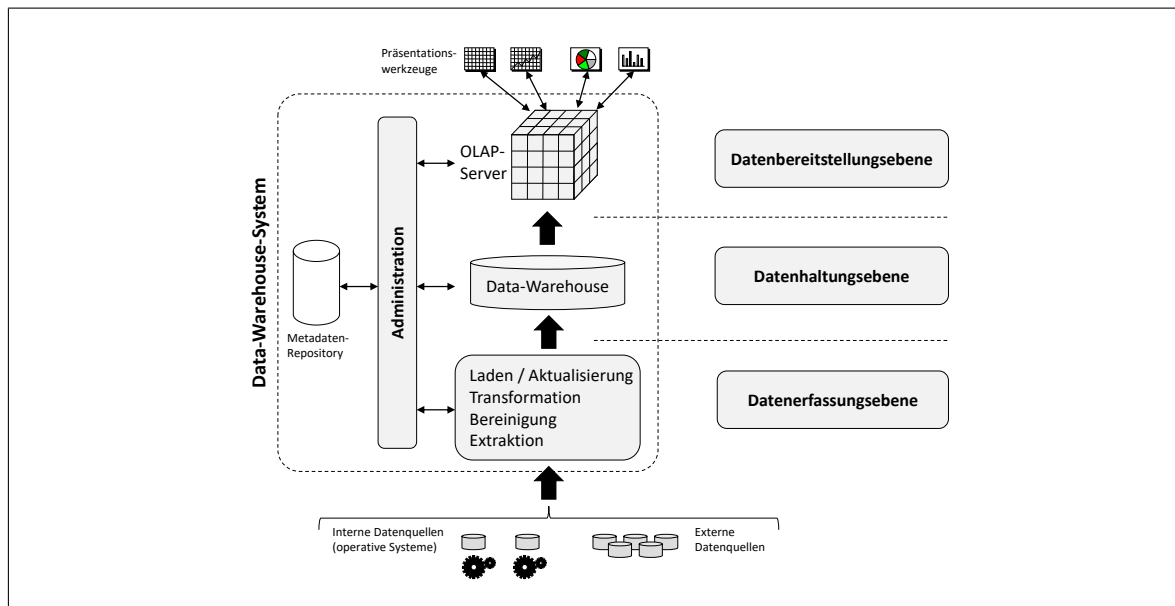


Abb. 3.3: Idealtypische Architektur eines Data-Warehouse-Systems (Böhnlein & Ulbrich-vom Ende 2000, S. 17)

Der Zugriff auf das zentrale DWH kann sowohl direkt als auch über sog. *Data Marts* (in Abb. 3.3 nicht dargestellt) erfolgen. *Data Marts* stellen somit ein weiteres mögliches Element der *Datenhaltungsebene* dar. Durch ein Data Mart wird eine Teilmenge des zentralen Data-Warehouse-Datenbestandes, ausgerichtet auf die Anforderungen einer spezifischen Zielgruppe, z.B. einen Unternehmensbereich, bereitgestellt (Golfarelli & Rizzi 2009, S. 8f.). Grundsätzlich wird mit dem Einsatz von Data Marts die Zielsetzung verfolgt, die Komplexität des Datenmodells sowie das Datenvolumen des Kern-Data-Warehouses zu reduzieren. Gründe für den Einsatz von Data Marts können dabei unterschiedlich sein. Als Beispiele sind die Steigerung der Eigenständigkeit der Zielgruppe bzw. des Unternehmensbereichs, Datenschutzaspekte, Performance-Gründe oder auch notwendige Lastverteilungen denkbar (Bauer & Günzel 2013, S. 68). Die Ausrichtung auf Anforderungen einer spezifischen Zielgruppe erfolgt bei *abhängigen Data Marts* durch die Bildung von abteilungs- oder auch nutzerspezifischen (materialisierten) Teil-

sichten auf das Kern-DHW. Als *unabhängig* werden Data Marts bezeichnet, wenn diese ohne den Zwischenschritt des Kern-DWH und somit direkt auf den Quellsystemen aufgebaut werden. Mit unabhängigen Data Marts erhöht sich, je nach Ausgestaltung, jedoch auch das Risiko von Inkonsistenzen in den abgeleiteten Aussagen (Bauer & Günzel 2013, S. 70ff., Golfarelli & Rizzi 2009, S. 10).

3.2.2 Ein generisches Schichtenmodell zur Beschreibung der Architektur von Data-Warehouse-Systemen

Im vorangehenden Abschn. 3.2.1 wird die Architektur eines DWH-Systems anhand seiner Bestandteile beschrieben. Bei Sinz & Ulbrich-vom Ende (2010) wird dazu ein generisches Schichtenmodell vorgeschlagen. Die Schichten sind anhand der durchzuführenden Aufgabe sachzielorientiert abgegrenzt. Eine Schicht kapselt bestimmte Funktionen und ggf. Datenbestände, die zur Durchführung der Aufgabe notwendig sind. Eine Schicht ist über die angebotene Schnittstelle durch benachbarte Schichten zugreifbar. Das generische Schichtenmodell ist in Abb. 3.4 dargestellt. Zur Realisierung der Schichten des Kern-DWH-Systems ist, je nach Realisierungsform, sowohl *persistente* als auch *virtualisierte* Datenverwaltung möglich.

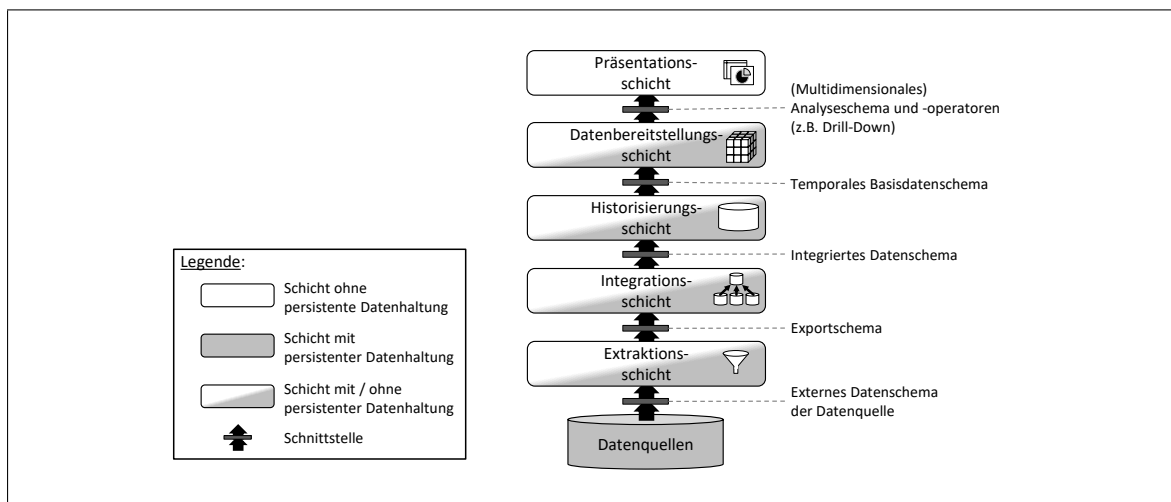


Abb. 3.4: Generische Schichtenarchitektur für Data-Warehouse-Systeme (Sinz & Ulbrich-vom Ende 2010)

Die Schichten lassen sich wie folgt beschreiben (Sinz & Ulbrich-vom Ende 2010):

- **Datenquellen:** Unternehmensinterne oder externe Datenquellen stellen eine Vorstufe des eigentlichen DWH-Systems dar. Über ein *externes Schema*

werden die für das DWH-System bereitgestellten Datenstrukturen beschrieben.

- **Extraktionsschicht:** Die Funktion der Extraktionsschicht ist das Laden der im DWH-System benötigten Datenstrukturen aus den Datenquellen. Über das *Exportschema* werden die extrahierten Datenstrukturen zur Weiterverarbeitung durch übergeordnete Schichten bereitgestellt.
- **Integrationsschicht:** Die Funktion der Integrationsschicht umfasst sowohl die *Transformation* (syntaktische und semantische Bereinigung) als auch die *Integration* der im Exportschema bereitgestellten Daten. Über ein *integriertes Datenschema* werden die resultierenden Datenstrukturen weitergegeben.
- **Historisierungsschicht:** In der Historisierungsschicht erfolgt die Erweiterung des integrierten Datenschemas um zeitliche Aspekte hin zu einem *temporalen Basisdatenschema* und ermöglicht so Analysen im Zeitverlauf.
- **Datenbereitstellungsschicht:** Gegenstand der Datenbereitstellungsschicht ist der Aufbau einer multidimensionalen Datenstruktur, die als *multidimensionales Analyseschema* mit den zugehörigen OLAP-Operatoren bereitgestellt werden. Das multidimensionale Datenmodell sowie dessen mögliche Realisierungen werden in Abschn. 3.3 eingeführt.
- **Präsentationsschicht:** Werkzeuge zur Nutzung des DWH-Systems werden in der Präsentationsschicht realisiert und i.d.R. nicht als Bestandteil von DWH-Systemen gesehen.

Anhand des generischen Architekturmodells können spezifische Architekturformen, ausgerichtet auf die zu unterstützenden Organisationsstrukturen und die Reichweite des DWH-Systems, als Extension beschreiben werden. Form und Anzahl der Exemplare je Schicht sowie deren Konfiguration stellen die zentralen Gestaltungskriterien dar. Die Konfiguration erfolgt im Wesentlichen hinsichtlich der Verteilung der Schichten sowie der Form der Datenverwaltung (Sinz & Ulbrich-vom Ende 2010).

3.3 Einführung des multidimensionalen Datenmodells

DWH-Systeme, im Speziellen deren Datenstrukturen, folgen der Metapher des mehrdimensionalen Kubus (*Hypercube*). Damit wird die Klassifikation quantifi-

zierbarer, entscheidungsrelevanter Kenngrößen (häufig synonym auch Maßzahlen oder Fakten) entlang qualifizierender Dimensionen eines mehrdimensionalen Würfels beschrieben. Ausprägungen einer Kenngröße werden dabei eindeutig den Dimensionen zugeordnet (Sinz 2002b, S. 312). Die Multidimensionalität von Daten hat ihren Ursprung in den OLAP-Regeln nach Codd et al. (1993). Nachfolgend wird das multidimensionale Datenmodell anhand seiner Struktur (Abschn. 3.3.1) sowie der darauf definierten Operatoren (Abschn. 3.3.2) eingeführt.

3.3.1 Struktur des multidimensionalen Datenmodells

Wesentliche Strukturbausteine des multidimensionalen Datenmodells sind *Kennzahlen* und *Dimensionen*. Diese werden nachfolgend eingeführt und in Zusammenhang gesetzt.

Kennzahlen als quantifizierender Bestandteil des multidimensionalen Datenmodells

Bei Kennzahlen handelt es sich um „*Größen zur quantitativen Darstellung betriebswirtschaftlicher Sachverhalte. Sie dienen dazu, komplexe Vorgänge schneller erfassbar zu machen*“ (Becker et al. 2011, S. 130). Als solche bilden sie auch den quantifizierenden Teil des multidimensionalen Datenmodells (Lehner 2003, S. 67). Kennzahlen resultieren aus der Verdichtung mehrerer einzelner Werte und können sich auf unterschiedliche Aspekte wie Zeitraum, Region oder Produkt beziehen (Becker et al. 2011, S. 130). Unterscheiden lassen sich dabei absolute und Verhältniskennzahlen. *Absolute Kennzahlen* sind Einzelwerte wie Bestandsgrößen, Summen oder Differenzen, während *Verhältniszahlen* relative Größen ausdrücken. Unterscheiden lassen sich dabei wiederum Beziehungs-, Index- oder Gliederungszahlen (Küpper et al. 2013, S. 471f.). In Beziehung gesetzte verschiedenartige Kenngrößen, wie beispielsweise Gewinn zu Kapital als Rentabilität, stellen *Beziehungszahlen* dar. *Indexzahlen* beschreiben die Relation gleichartiger, jedoch örtlich oder zeitlich verschiedener Werte, mit der Zielsetzung, Abweichungen in Bezug auf eine Basisgröße zu erkennen. Über *Gliederungszahlen* wird der Anteil einer Größe an einer Gesamtmenge bestimmt. Beide Größen werden dabei in derselben Dimension gemessen, wie beispielsweise der Anteil einzelner Kostenarten an den Gesamtkosten (Küpper et al. 2013, S. 471f.). Zur Beurteilung komplexer wirtschaftlicher Sachverhalte werden i.d.R. mehrere zueinander in Beziehung stehende Kennzahlen, die eine gewisse Ordnung aufweisen, herangezogen. Man spricht

von einem Kennzahlensystem, das die Beziehung zwischen den relevanten Kennzahlen widerspiegelt (Küpper et al. 2013, S. 472f.).

Dimensionen als qualifizierender Bestandteil des multidimensionalen Datenmodells

Dimensionen bzw. Dimensionshierarchien bilden den qualifizierenden Teil des multidimensionalen Datenmodells (Lehner 2003, S. 65). Kennzahlen multidimensionaler Datenstrukturen werden entlang von Dimensionen aggregiert. Die Innensicht einer Dimension ist über Dimensionselemente unterschiedlicher Stufe sowie entsprechende Verdichtungswege über diese Elemente strukturiert. Entlang dieser Verdichtungswege können Werte niedrigerer Stufe zu einem Wert höherer Stufe zusammengefasst werden. In Gegenrichtung kann ein aggregierter Wert in Einzelwerte aufgeteilt werden (Totok 2000, S. 87). Dimensionen werden grundsätzlich auf Schemaebene modelliert. Ihre Ausprägungen können in einer hierarchischen Baumstruktur abgebildet werden. Die Dimensionselemente bzw. die durch diese beschriebene Verdichtungsstufe wird in Form von Knoten und Blättern des Baumes abgebildet (Bauer & Günzel 2013, S. 119f.). Dimensionen werden hinsichtlich ihrer inneren Struktur unterschieden. Nachfolgend werden verschiedene für die Arbeit relevante Formen vorgestellt.

Einfache Hierarchien Die einfachste Form einer Dimensionsstruktur wird als *einfache Hierarchie* bzw. häufig auch als *Standarddimension* bezeichnet (Totok 2000, S. 92). Mit Ausnahme der Wurzel (höchste Hierarchiestufe) besitzen demnach alle Knoten (Dimensionselemente) genau ein übergeordnetes Element. Für alle Elemente feinsten Granularität ist die Pfadlänge (Anzahl zu durchlaufender Hierarchiestufen) hin zur Wurzel identisch. Auf Ausprägungsebene besitzt diese Hierarchieform somit einen streng hierarchischen Aufbau (Gluchowski et al. 2008, S. 153). Auf Schemaebene sind die einzelnen Hierarchiestufen sequentiell entsprechend der Verdichtungsstufen angeordnet. Durch die niedrigste Hierarchiestufe einer Dimension (Blätter auf Ausprägungsebene) werden atomare Werte einer Kennzahl beschrieben, die auf höheren Stufen in aggregierter Form vorliegen (Bauer & Günzel 2013, S. 120).

Parallele Hierarchien *Parallele Hierarchien* (oder auch *multiple Hierarchien*) entstehen durch Berücksichtigung unterschiedlicher unabhängiger Konsolidierungs-

pfade innerhalb einer Dimension (Hahne 2014, S. 37). Durch die Verdichtung entlang eines Pfades wird ein Aspekt der Dimension beleuchtet, während die übrigen Pfade/Aspekte ausgeblendet werden (Bauer & Günzel 2013, S. 121). Parallele Hierarchien müssen in mindestens zwei Hierarchiestufen übereinstimmen, jedoch nicht zwangsläufig in höchster und niedrigster (Böhnlein 2001, S. 134). Eine vollständige Verdichtung zu einem gemeinsamen übergeordneten Knoten führt, unabhängig vom gewählten Pfad, zu einem konsistenten Ergebnis (Hahne 2014, S. 38). Die Ableitung eines übergeordneten Wertes unter Berücksichtigung mehrerer paralleler Pfade führt hingegen zu einem verfälschten Ergebnis (Totok 2000, S. 93). Abb. 3.5 zeigt beispielhaft eine parallele Hierarchie auf Schema- und Ausprägungsebene aus dem universitären Kontext nach Böhnlein (2001, S. 135). Die Aggregation *Studienfach*-bezogener Kennzahlen hin zur Hierarchiestufe *Universität* kann alternativ über den *Fachbereich* oder über die *Fachgruppe* erfolgen. Auf Stufe der *Universität* liegt wieder ein konsistentes Ergebnis vor.

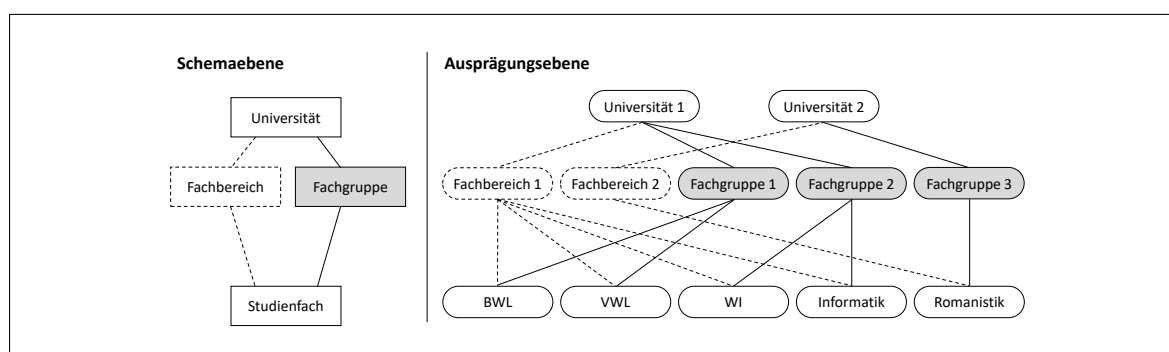


Abb. 3.5: Beispiel einer Dimension mit parallelen Hierarchien (Böhnlein & Ulbrich-vom Ende 2000, S. 135) (eigene Darstellung)

Unvollständige und rekursive (unbalancierte) Hierarchien Die Begriffe *unvollständige* oder *unbalancierte* Hierarchie werden in der Literatur z.T. heterogen abgegrenzt. Grundsätzlich bilden diese Hierarchieformen auf Ausprägungsebene einen nicht ausgeglichenen Baum. Das bedeutet, dass der Weg von Wurzelement zu Blattelement nicht für alle Blätter gleich ist (Oehler 2006, S. 139). Bei Golfarelli & Rizzi (2009, S. 116f.) wird zwischen *unbalancierter* und *unvollständiger* Hierarchie differenziert. In einer *unvollständigen Hierarchie* fehlen auf Instanzebene ein oder mehrere Dimensionsstufen für einzelne Datensätze. Hieraus ergeben sich nicht nur unterschiedliche Pfadlängen von Wurzel zu Blatt, sondern auch semantisch heterogene, inkonsistente Eltern-Kind-Beziehungen innerhalb eines Hierarchiebaumes. Abb. 3.6 zeigt dies beispielhaft: Während die Bezirke *Oberfran-*

ken und Oberbayern eine Hierarchiebeziehung zur Landesebene haben, ist Norfolk direkt einem Element der Dimensionsstufe *Staat* zugeordnet (Golfarelli & Rizzi 2009, S. 116f.). Wie das Beispiel zeigt, ist eine unvollständige Hierarchie auf Schemaebene nicht ersichtlich.

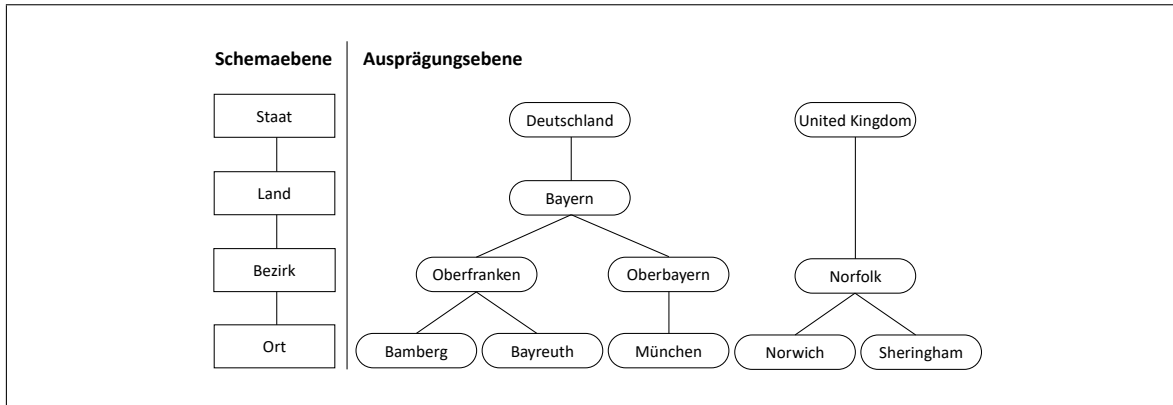


Abb. 3.6: Beispiel einer Dimension mit unvollständiger Hierarchie (Golfarelli & Rizzi 2009, S. 116) (eigene Anpassung)

Eine *rekursive (unbalancierte)* Hierarchie zeigt ebenfalls heterogene Pfadlängen von Wurzel zu Blatt, hat jedoch semantisch homogene, konsistente Eltern-Kind-Beziehungen (Golfarelli & Rizzi 2009, S. 117f.). Als Beispiel ist die unterschiedlich tiefe Strukturierung von Organisationseinheiten zu nennen, wie in Abb. 3.7 dargestellt. Eine Organisationseinheit *Realisierung* ist einer Organisationseinheit *Entwicklung* und diese wiederum der Einheit *IT* untergeordnet, während beispielsweise eine *Marketing*-Einheit nicht weiter untergliedert ist. Trotz unterschiedlicher Pfadlängen kann jede Organisationseinheit wiederum in einer konsistenten Eltern-Kind-Beziehung einer übergeordneten Organisationseinheit zugeordnet werden. In einer rekursiven Hierarchie ist keine feste Ebenenzahl vorgebar (Hahne 2014, S. 39). Können zudem Kennzahleninstanzen nicht nur auf Granularität der Blätter, sondern aller Knoten des Hierarchiebaumes zugeordnet werden, wird vom Konzept der *bebuchbaren Knoten* gesprochen (Hahne 2014, S. 39). Im Beispiel aus Abb. 3.7 ist so beispielsweise die Budgetzuordnung für jede Organisationseinheit ungeachtet der Ebene möglich.

Aggregierbarkeit von Kennzahlen entlang der Dimensionshierarchien

Kennzahlen werden zur qualitativen Analyse mit Dimensionen assoziiert und entlang der Verdichtungspfade aggregiert. Jedoch führen nicht alle Aggregationen notwendigerweise zu sinnvollen und korrekten Ergebnissen (Lehner 2003, S. 70).

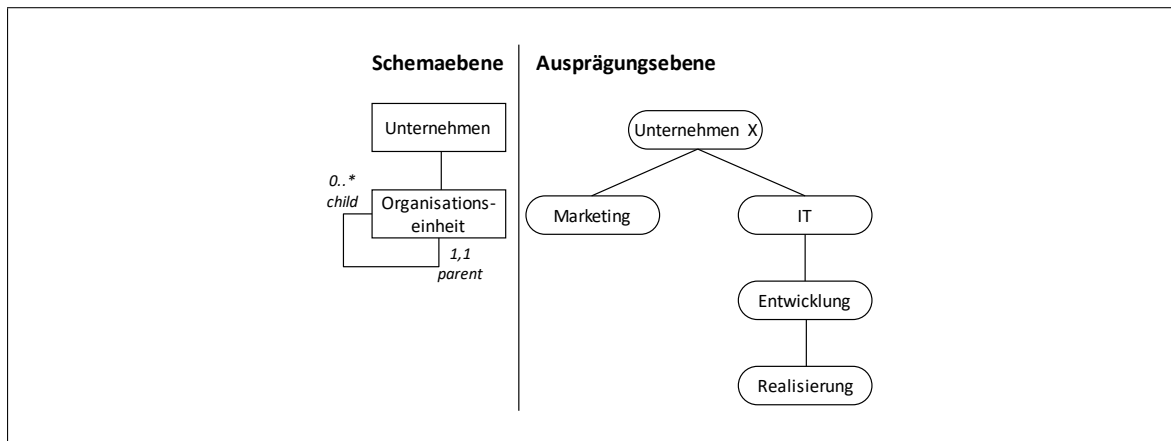


Abb. 3.7: Beispiel einer Dimension mit rekursiven Strukturen

Disjunktheit, Vollständigkeit und Typverträglichkeit stellen hierbei Voraussetzungen für die sinnvolle Aggregierbarkeit dar (Lenz & Shoshani 1997). *Disjunktheit* beschreibt die Forderung, eine Kennzahleninstanz nur einmalig in die Berechnung des Aggregats einfließen zu lassen. Auf niedrigster Stufe ist die Kennzahlenausprägung eindeutig einem Dimensionselement zuzuordnen. Zudem sind Dimensionselemente eindeutig durch Elemente der nächsthöheren Stufe bei Aggregationen zu erfassen (Lehner 2003, S. 70; Lenz & Shoshani 1997). *Vollständigkeit* beschreibt die Anforderung, auf niedrigster Dimensionsstufe alle Kennzahlenausprägungen zu erfassen sowie innerhalb einer Dimension alle Elemente einer Ebene einem Element nächsthöherer Ebene zuordnen zu können (Lenz & Shoshani 1997). *Typverträglichkeit* fordert die Verträglichkeit der Aggregationsoperation für eine bestimmte Kennzahl entlang einer Dimension (Lenz & Shoshani 1997). Die Summation von Bestandsgrößen entlang der Dimension *Zeit* stellt beispielsweise eine Unverträglichkeit dar. Bei Aggregation eines Lagerbestandes über mehrere Perioden finden Gegenstände, die sich über einzelne Perioden hinweg im Lager befinden, mehrfach Berücksichtigung. Die Aggregation entlang der Dimension *Zeit* stellt somit einen Verstoß gegen die Forderung der Disjunktheit dar (Lehner 2003, S. 71). Bei Kimball & Ross (2002, S. 17) erfolgt eine Klassifikation in additive, semi-additive und nicht-additive Kennzahlen. Während für semi-additive Kennzahlen die Summierung nicht für alle in Beziehung stehenden Dimensionen möglich ist, besteht für additive Kennzahlen in dieser Hinsicht keine Einschränkung. Für nicht-additive Kennzahlen ist keine Summierung entlang ihrer Dimensionen möglich. Eine Einbeziehung dieser Werte in Aggregationsoperationen könnte über Durchschnittsbildung oder Zählen erfolgen (Kimball & Ross 2002, S. 17).

3.3.2 Operatoren des multidimensionalen Datenmodells

Nachfolgend werden die auf Grundlage der Struktur des multidimensionalen Datenmodells definierten, für die Arbeit wesentlichen Operatoren vorgestellt.

Die Auswahl multidimensionaler Daten kann über die Operatoren *Selection*, *Slice* oder *Dice* erfolgen. Der Operator *Selection* beschreibt generell die Filterung von Würfel­daten anhand bestimmter Kriterien, die damit verbundene Einschränkung der Datenmenge sowie die Auswahl bestimmter Daten. *Slice* und *Dice* stellen Spezialfälle der *Selection* dar (Böhnlein 2001, S. 145, Jarke et al. 2003, S. 90). Der Operator *Slice* beschreibt, bildlich gesprochen, die Selektion einer Scheibe des multidimensionalen Würfels. Die Dimensionalität des ursprünglichen Würfels wird um eine Dimension verringert (Hahne 2014, S. 30). Entlang einer Dimension erfolgt die Einschränkung auf einen bestimmten Wert einer Hierarchiestufe (Bauer & Günzel 2013, S. 124f.). Die Selektion einer Teilmenge (Auswahl mehrerer Elemente) in mindestens einer Dimension wird als *Dice* bezeichnet und entspricht bildlich der Extraktion eines Teilwürfels aus dem *Hypercube*. Die Dimensionalität des Teilwürfels bleibt, bezogen auf den ursprünglichen Würfel, unverändert (Bauer & Günzel 2013, S. 125, Hahne 2014, S. 31).

Der Wechsel zwischen Hierarchie- bzw. Verdichtungsstufen innerhalb einer Dimension erfolgt über die komplementären Operatoren *Roll-Up* und *Drill-Down* (Hahne 2014, S. 31). Der Operator *Roll-Up* beschreibt die Verdichtung von Kennzahlenwerten entlang eines vorgegebenen Konsolidierungsweges einer Dimension, der Operator *Drill-Down* in entgegengesetzter Richtung die Erhöhung des Detailgrades auf granulareren Ebenen (Hahne 2014, S. 33). Für jedes Element einer Dimension kann eine entsprechende Verdichtungsstufe erzeugt werden (Bauer & Günzel 2013, S. 124). Sowohl bei *Drill-Down* als auch bei *Roll-Up* wird die Dimensionalität nicht verändert (Böhnlein 2001, S. 144). Der Operator *Rotation* (*Pivotierung*) ermöglicht die Analyse aus unterschiedlichen Perspektiven durch Drehen des Würfels durch vertauschen von Dimensionen (Bauer & Günzel 2013, S. 123). Die Dimensionalität des Würfels bleibt dabei ebenfalls unverändert (Böhnlein 2001, S. 147).

3.3.3 Realisierungsformen des multidimensionalen Datenmodells

Das multidimensionale Datenmodell kann bzgl. der eingesetzten Basistechnologie auf unterschiedliche Weise realisiert und gespeichert werden. Die drei etablierten Formen ROLAP, MOLAP und HOLAP werden dazu nachfolgend kurz charakterisiert.

Die Realisierung des multidimensionalen Datenmodells auf der Basis relationaler Datenbanken, die dem Relationenmodell nach Codd (1970) folgen, wird als *Relationales OLAP* (ROLAP) bezeichnet. Im Mittelpunkt steht die Abbildung multidimensionaler Konzepte, wie Dimensionen oder Klassifikationshierarchien auf zweidimensionale relationale Tabellen. Durch die relationale Abbildung multidimensionaler Strukturen ist auf einen möglichst geringen Verlust multidimensionaler Anwendungssemantik (z.B. Klassifikationshierarchien), die Effizienz bei Übersetzung und Abarbeitung multidimensionaler Anfragen sowie die gute Wartbarkeit der relationalen Strukturen zu achten (Bauer & Günzel 2013, S. 242f.). Eine direkte Abbildung multidimensionaler Konzepte auf relationale Strukturen ist mangels Unterstützung durch das Datenbankmodell nicht möglich (Golfarelli & Rizzi 2009, S. 38). Als Strukturierungsalternativen haben sich für relationale Systeme das denormalisierte *Star-Schema* bzw. das normalisierte *Snowflake-Schema* bzw. Mischformen zwischen diesen beiden durchgesetzt. Das *Star-Schema* besteht aus einer Menge sog. *Dimensionstabellen*, wobei jede einzelne eine Dimension vollständig und denormalisiert abbildet. Kennzahlen werden in sog. *Fakttabellen* verwaltet. Der Primärschlüssel der Fakttabellen setzt sich aus den Fremdschlüsseln der referenzierten Dimensionstabellen zusammen (Golfarelli & Rizzi 2009, S. 221). Die Strukturierung nach dem *Snowflake-Schema* sieht die normalisierte Abbildung der Dimensionen vor. Je Dimensionsstufe wird eine Tabelle gebildet, die über den Fremdschlüssel auf die Tabelle der nächst höheren Stufe verweist (Bauer & Günzel 2013, S. 243). Wesentliche Performance-Einschränkungen bei relationalen Systemen ergeben sich zum einen durch die Übersetzung von OLAP-Anfragen in die Zugriffssprache *Structured Query Language* (SQL), zum anderen durch aufwendige *Join*-Operationen zwischen Tabellen. Die Denormalisierung im *Star-Schema* stellt hier einen Ansatzpunkt zur Reduktion der *Join*-Operationen dar (Golfarelli & Rizzi 2009, S. 38).

Als *Multidimensionales OLAP* (MOLAP) wird das Konzept der direkten Realisierung des multidimensionalen Datenmodells ohne die Abbildung auf relationale Konzepte und ohne die Performance beeinflussende Übersetzung entsprechender Anfragen verstanden. Im Wesentlichen basiert die Abbildung auf dem Datentyp *Array* (Bauer & Günzel 2013, S. 265f.). Geschwindigkeitsvorteile entstehen v.a. durch die direkte Abbildung des multidimensionalen Datenmodells (Golfarelli & Rizzi 2009, S. 39). Dem gegenüber stehen jedoch speicherbezogene Grenzen der multidimensionalen Systeme. Diese sind unter anderem durch das Problem der Dünnesetztheit (*Sparsity*) bedingt (Bauer & Günzel 2013, S. 281f.). Das bedeutet, dass nur Teile des multidimensionalen Würfels mit Nutzdaten belegt werden. Die übrigen Würfelzellen reservieren Speicherplatz für nicht stattfindende Dimensionkombinationen (Golfarelli & Rizzi 2009, S. 219). Multidimensionale Systeme sind zudem auf das schnelle Lesen ausgerichtet. Das Laden neuer bzw. das Aktualisieren bestehender Daten zieht häufig eine vollständige Reorganisation der Speicherstrukturen mit Indexberechnung nach sich (Bauer & Günzel 2013, S. 281f.).

Bei Bauer & Günzel (2013, S. 283f. u. 297) wird die sinnvolle Kombination von ROLAP und MOLAP zur Realisierung der Vorteile beider Realisierungsarten motiviert. *Hybrides OLAP* (HOLAP) bezeichnet nach Bauer & Günzel (2013, S. 283f.) einen Ansatz, der die Stärken relationaler Systeme (u.a. Skalierbarkeit), eingesetzt als Basisdatenbank, und multidimensionaler Systeme (u.a. gute Anfrageperformance), eingesetzt in abhängigen Data Marts, verbindet. In großer Anzahl vorliegende Detaildaten werden in relationalen, häufig benötigte Daten (meist höheren Aggregationsgrades) in multidimensionalen Strukturen abgelegt. Anfragen an HOLAP-Systeme werden für den Nutzer transparent abgearbeitet und nach Möglichkeit aus den multidimensionalen Strukturen heraus beantwortet. Für den Fall, dass benötigte Daten dort nicht verfügbar sind, werden Anfragen an das relationale System durchgestellt. Die Aufteilung der Daten auf multidimensionale und relationale Datenstrukturen erfolgt anwendungsspezifisch (Bauer & Günzel 2013, S. 283f.).

lein (2001, S. 265), erweitert um die Beziehung von Dimension und Basiskennzahl basierend auf der *Basiskennzahlensicht* nach Böhnlein (2001, S. 273), dargestellt. Anhand der Ausführungen nach Böhnlein (2001, S. 264f.) wird das Metamodell nachfolgend vorgestellt: In der *Dimensionssicht* wird die Strukturierung qualitativer Daten in Form von Dimensionen und deren Strukturen beschrieben. Eine *Dimension* besteht aus mindestens zwei *Dimensionshierarchiestufen* (DH), die die verschiedenen Verdichtungsstufen einer Dimension spezifizieren. Jeder DH ist eine Stufe zugeordnet und kann über *Dimensionsattribute* näher beschrieben werden. *Sub-Dimensionshierarchiestufe* (SDH) und *Gesamt-Dimensionshierarchiestufe* (GDH) stellen Spezialisierungen einer DH dar. Je Dimension ist genau eine GDH vorgesehen und beschreibt die ggf. mehrstufige Verdichtung über alle SDHs hinweg. Über SDHs werden übrige Verdichtungsstufen einer Dimension spezifiziert. Zur Modellierung einer mehrstufigen hierarchischen Struktur zwischen Hierarchiestufen (GDH, SDH) dienen *Aggregationsbeziehungen*, die sich je nach Art der in Beziehung gesetzten Stufen unterscheiden lassen. Die Verdichtung erfolgt stets von niedrigerer SDH zu höherer SDH bzw. GDH. Während von der niedrigeren zur höheren Stufe stets eine eindeutige Zuordnung von Instanzen angenommen wird, kann die Gegenrichtung durch Angabe einer *Kardinalität* in (min, max)-Notation präzisiert werden. Abb. 3.10 zeigt dem Metamodell folgend die Dimensionen *Studienfach*, *Geographie* und *Zeit*. Während die Dimension *Zeit* eine einfache Hierarchie mit der SDH *Semester* darstellt, ist die Dimension *Geographie* mit dem Verdichtungsweg *Ort-Bundesland-Land* auf Instanzebene unbalanciert (vgl. Abschn. 3.3.1, Abb. 3.6). Die Dimension *Studienfach* bildet eine parallele Dimension mit alternativen Verdichtungspfaden über die SDHs *Fachbereich* oder *Fachgruppe*. Zur näheren Beschreibung der SDH *Universität* dient das Dimensionsattribut *Präsident*. Die Beziehung zwischen Dimensionen (SDHs der Dimensionen) und *Basiskennzahlen* erfolgt über *Dimensionsschnittstellen*. *Rollenbezeichner*, die *Aggregierbarkeit* (semi- und uneingeschränkte Aggregierbarkeit) sowie eine *Dimensionslinie* (Spezifikation der niedrigsten gültigen Granularitätsstufe einer Dimension für eine Kennzahl) können die Beziehung zwischen Kennzahl und Dimension näher beschreiben. Für das Beispiel in Abb. 3.10 wird die Kennzahl *Anzahl eingeschriebener Studierender* definiert, die als Bestandsgröße hinsichtlich der Dimension *Zeit* nicht aggregiert werden kann (vgl. Abschn. 3.3.1 zur Aggregierbarkeit von Kennzahlen).

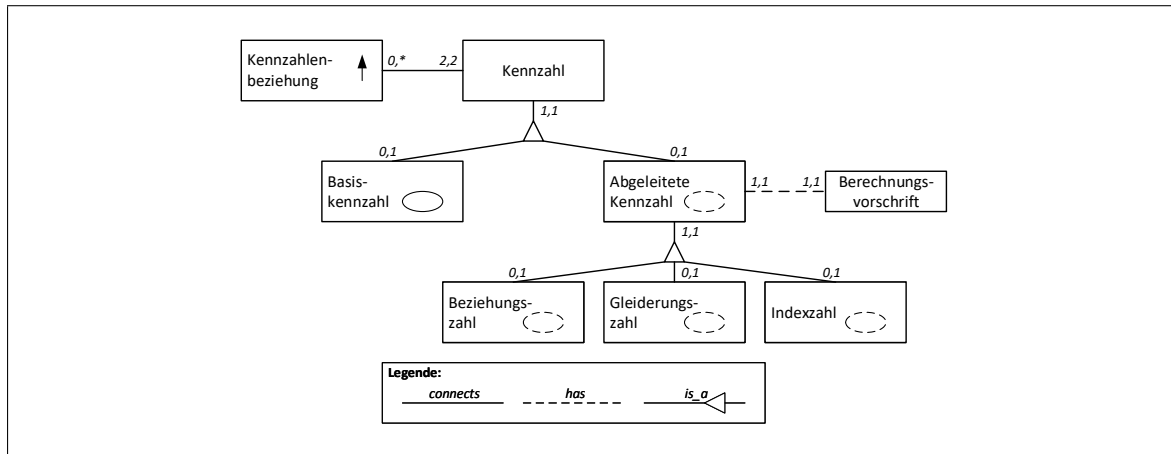


Abb. 3.9: Metamodell der Sicht auf Kennzahlensysteme des Semantischen Data-Warehouse-Modells (Böhnlein 2001, S. 277) (eigene, verkürzte Darstellung)

Die *Sicht auf Kennzahlensysteme* ist in Abb. 3.9 aufbereitet und wird nachfolgend auf Basis der Ausführungen nach Böhnlein (2001, S. 276f.) vorgestellt: Die Kennzahlensicht fokussiert, unabhängig von Dimensionen, auf die Modellierung von Kennzahlensystemen. Als Basiskennzahlen (*Anzahl eingeschriebener Studierender* im Beispiel Abb. 3.10) werden grundsätzlich absolute Kennzahlen verstanden, die die Grundlage zur Berechnung abgeleiteter Kennzahlen bilden, wobei letztere auch selbst Ausgangspunkt weiterer Ableitungen sein können. Zur Modellierung der Beziehungen zwischen Kennzahlen wird die *Kennzahlenbeziehung* sowie *Berechnungsvorschriften* als Eigenschaften der abgeleiteten Kennzahl verwendet. Als Spezialisierungen abgeleiteter Kennzahlen werden *Beziehungszahlen*, *Gliederungszahlen* und *Indexzahlen*, wie in Abschn. 3.3.1 vorgestellt, durch das Metamodell des SDWM unterstützt. Im Beispiel (Abb. 3.10) wird die Berechnung der *Gliederungszahl Studierendenteil* (Anzahl der eingeschriebenen Studierenden eines Studienfaches bezogen auf die Anzahl der eingeschriebenen Studierenden eines Fachbereiches) samt Berechnungsvorschrift modelliert.

3.5 Grundlagen prozessorientierter Data-Warehouse-Systeme

Das Konzept *prozessorientiertes Data-Warehouse-System* wird zunächst allgemein abgegrenzt (Abschn. 3.5.1) und anhand seiner Anwendungsszenarien eingeführt (Abschn. 3.5.2). Die Erläuterung des Begriffes der *Latenz* im Kontext von pDWH-Systemen in Abschn. 3.5.3 schließt das Kapitel.

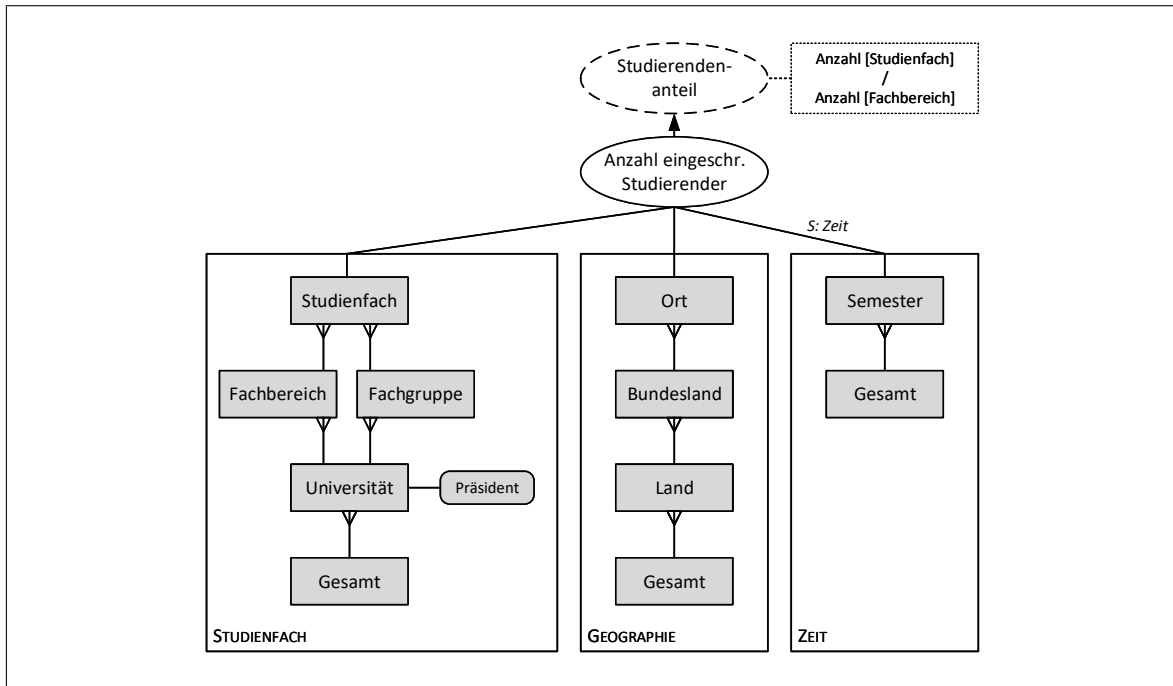


Abb. 3.10: Beispielhafte Anwendung des Semantischen Data-Warehouse-Modells im universitären Kontext (Böhnlein 2001, S. 279ff.) (eigene Darstellung)

3.5.1 Abgrenzung prozessorientierter Data-Warehouse-Systeme

In der Literatur wird das pDWH-Konzept, auch in Abhängigkeit von der intendierten Anwendung (vgl. hierzu Abschn. 3.5.2), unterschiedlich bestimmt und gegenüber dem etablierten DWH-Verständnis (vgl. Abschn. 3.1.2) abgegrenzt. Baars & Sun (2013) ordnen einem pDWH-System im Kontext des übergeordneten Konzeptes der *Business Process Intelligence* eine zentrale Bedeutung bei der Erfassung von Ist-Daten zu realen Prozessverläufen, deren Integration und Aufbereitung zu. Die Datenbereitstellung zu Prozessen und deren Verhalten als Gegenstand eines pDWH-Systems wird bei Becker & Chamoni (2008) als spezifische DWH-Anwendung bzgl. Datenstrukturen, Konfigurationen und zugehöriger Abläufe verstanden. Von zentraler Bedeutung ist dabei die Gestaltung des Datenmodells. Bei Schiefer et al. (2004) werden die vier Charakteristika (*subject-oriented, integrated, time-variant, non-volatile*) nach Inmon (2005, S. 29ff.) (vgl. Abschn. 3.1.2) aufgegriffen und erweitert. Über die zusätzliche Eigenschaft der *Prozessorientierung* wird eine Spezialisierung auf Prozesse und deren Verhalten definiert. Analysethemen eines pDWH-Systems sind auf die Prozessspezifikation, Ressourcen, fachliche Geschäftsobjekte, die Laufzeitumgebung sowie die Prozessumgebung ausgerichtet. Daten werden dabei sowohl auf Granularität von Prozessinstanzen als auch aggregiert angeboten (Schiefer et al. 2004). Zusammenfassend kann ein pDWH-

System als Spezialisierung eines DWH-Systems verstanden werden. Bei Baars & Sun (2013) wird explizit darauf hingewiesen, dass innerhalb eines Unternehmens ein pDWH-System hinsichtlich der Aussagenkonsistenz stets mit bestehenden DWH-Lösungen abgeglichen werden sollte.

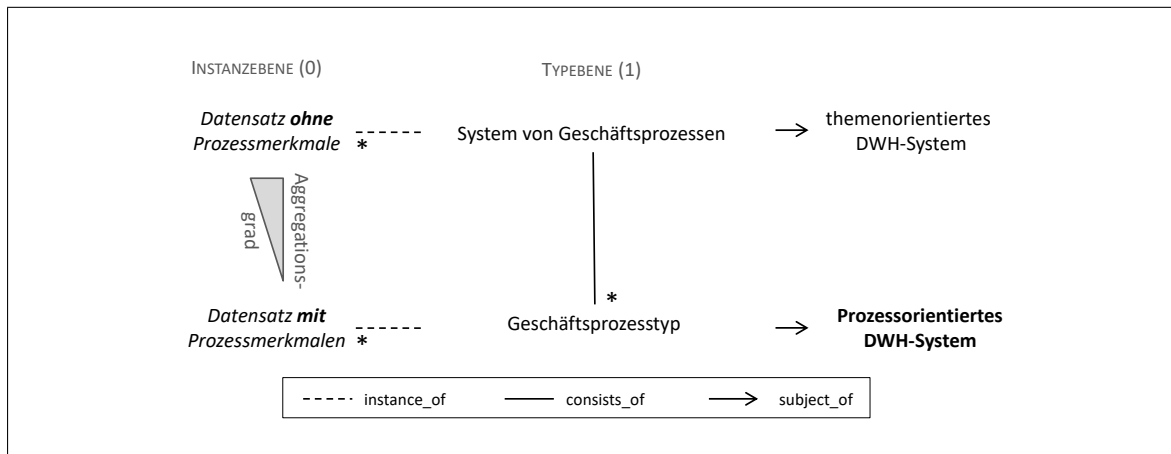


Abb. 3.11: Schematische Abgrenzung themen- und prozessorientierter Data-Warehouse-Systeme

Ein konsolidiertes Verständnis eines pDWH-Systems sowie die Abgrenzung zum klassischen Verständnis ist in Abb. 3.11 gezeigt. Klassische themenorientierte DWH-Systeme stellen GP-übergreifend aggregierte Daten über die Ergebnisse von Prozessen zur taktischen und strategischen Planung bereit. Die regional- und produktspezifische Analyse von Verkaufszahlen stellt dabei einen möglichen Themenbereich dar. Die zugehörige Datenbasis (Instanzebene) weist einen tendenziell höheren Aggregationsgrad auf und abstrahiert damit von einzelnen Geschäftstransaktionen (Bauer & Günzel 2013, S. 10f.). Prozessorientierte DWH-Systeme bieten detaillierte Daten zu Ablauf und Ergebnissen einzelner GP-Typen auf der Granularität einzelner Instanzen. In den zugrunde liegenden Datensätzen werden dazu zusätzlich Prozessmerkmale verwaltet. Als Prozessmerkmale werden z.B. das Prozessschema, Ablaufdaten oder Ereignisse in Prozessinstanzen verstanden. Aus pDWH-Systemen abgeleitete Ergebnisse sollten in konsistenter Form in GP-übergreifenden Analysen eines Systems von Geschäftsprozessen berücksichtigt werden. Kennzahlen eines pDWH-Systems sind dabei auf die Bewertung von Qualität, Kosten und Zeit ausgerichtet (Bange & Fuchs 2011).

3.5.2 Anwendungsszenarien prozessorientierter Data-Warehouse-Systeme

Grundsätzlich leiten sich die Anwendungsszenarien von pDWH-Systemen aus der Zielsetzung der effizienteren und innovativeren Durchführung und Gestaltung von Geschäftsprozessen ab. In der Literatur finden sich hierbei dennoch Unterschiede. In einigen Arbeiten wird die Anwendung prozessorientierter DWH-Systeme lediglich für *operative* Szenarien, also zur Unterstützung von Entscheidungen während der Prozessausführung (*Runtime*), motiviert (z.B. Bucher & Dinter 2008a, Bucher & Dinter 2008b). Andere Beiträge (z.B. Schwingel 2010, Bange & Fuchs 2011) beziehen zudem mittelfristige, auf die Prozessgestaltung bezogene Fragestellungen (*Buildtime*) mit ein. Die vorliegende Arbeit setzt auf der letztgenannten Abgrenzung auf. Basierend auf einer unabhängigen Datengrundlage lassen sich Informationen sowohl für operative als auch für gestaltungsbezogene Entscheidungssituationen ableiten. Die beiden resultierenden Anwendungsszenarien von pDWH-Systemen (im Weiteren als pDWH-Szenarien bezeichnet) werden in Tab. 3.2 charakterisiert und gegenübergestellt.

	verhaltensorientiert (<i>Runtime</i>)	strukturorientiert (<i>Buildtime</i>)
Ausrichtung	operativ	gestaltungsbezogen
Gegenstand	laufende Instanz	beendete Instanzen
Einfluss auf laufende Instanz	... nachfolgende Instanzen
Zielsetzung	aktive Steuerung	Verbesserung / Gestaltung
Entscheidungsfindung	teil-/vollautomatisiert	teilautomatisiert
Informationsbedarf	Ereignisse Prozessablaufkennzahlen	Ereignisse Prozessablaufkennzahlen Prozessergebniskennzahlen

Tab. 3.2: Anwendungsszenarien prozessorientierter Data-Warehouse-Systeme

Im *verhaltensorientierten*, auf die Ausführung von Prozessen ausgerichteten Szenario (häufig auch als *operativ* charakterisiert) bildet die laufende Instanz eines Geschäftsprozesses den zentralen Untersuchungsgegenstand. Zielsetzung ist die aktive Steuerung dieser Prozessinstanz. Entscheidungssituationen ergeben sich v.a. aufgrund von Ereignissen in Geschäftsprozessen bzw. deren ausführbaren Instanzen. Die Entscheidungsfindung findet dabei teil- oder vollautomatisiert statt. Als Entscheidungsgrundlage dienen mehrheitlich Ereignisse im Geschäftsprozess und Prozessablaufkennzahlen (Schwingel 2010). Für das verhaltensorientier-

te Szenario kann ein pDWH-System bzw. dessen Datenbasis, als prozessorientiertes Data-Warehouse (pDWH) bezeichnet, ebenfalls in den betrieblichen Regelkreis eingeordnet werden. Grundlage bildet hierzu die regelkreisorientierte Einordnung themenorientierter DWH-Systeme nach Sinz (2002b) (vgl. Abschn. 3.1.2). Abb. 3.12 zeigt die erweiterte Darstellung. Das pDWH-System bildet die Hilfsregelstrecke auf Geschäftsprozessebene. Neben operativen Daten werden auch Ablaufdaten der Prozessinstanzen erfasst und in aufbereiteter Form als Ereignisse und Ablaufanalysen bzw. -berichte der *GP-Lenkung* zur Verfügung gestellt. Die Regelstrecke des übergeordneten Regelkreises, das themenorientierte Unternehmens-DWH, sollte hinsichtlich der Konsistenz aus dem pDWH-System mit weiter aufbereiteten und aggregierten Daten bedient werden. Die direkte Extraktion und Transformation ergänzender Daten aus den operativen Datenbanken für das übergeordnete DWH ist dabei nicht ausgeschlossen.

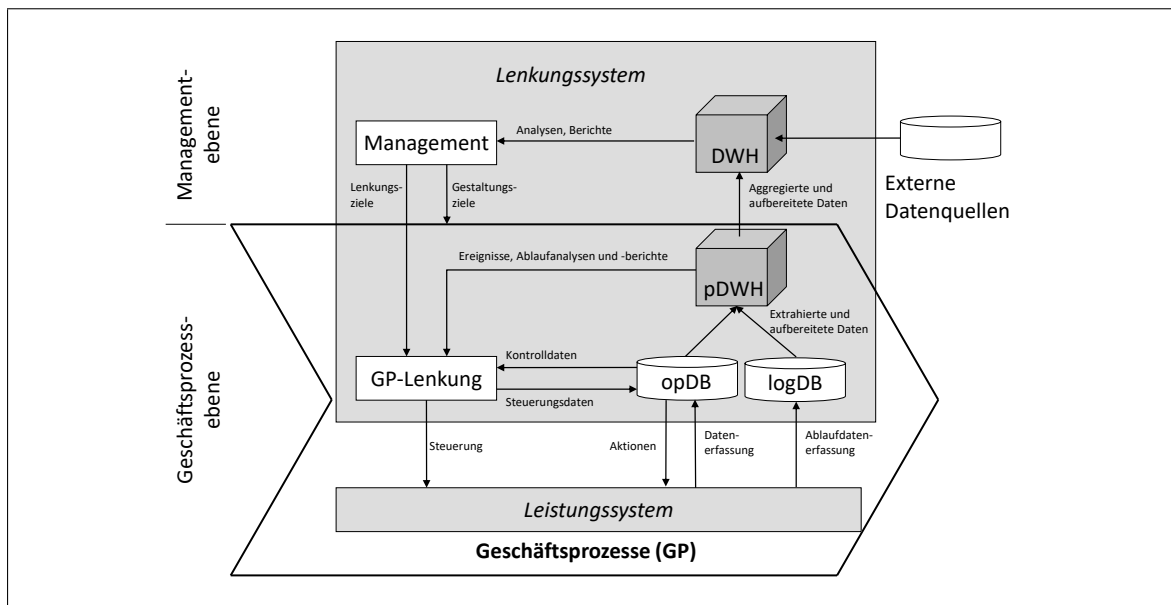


Abb. 3.12: Regelkreisorientierte Einordnung eines prozessorientierten Data-Warehouse-Systems

Im auf die Gestaltung von Geschäftsprozessen ausgerichteten pDWH-Szenario (*gestaltungsorientiert*) fließen neben Ereignissen und Ablaufkennzahlen auch Prozessergebniskennzahlen in die teilautomatisierte Entscheidungsfindung mit ein (Schwengel 2010). Zielsetzung ist die Gestaltung bzw. Verbesserung von Geschäftsprozessen. Resultierende Maßnahmen können dabei die Rahmenbedingungen (z.B. Ressourcenausstattung, technische Ausstattung maschineller Aufgabenträger) zur Prozessdurchführung oder auch den Geschäftsprozess auf Schemaebene (Felden et al. 2010) betreffen. Unter teilautomatisierter Entscheidungsfindung

wird die Bereitstellung des multidimensionalen Entscheidungskontextes durch ein AwS verstanden, wobei die eigentliche Entscheidungsfindung durch eine Person erfolgt. Bei vollautomatisierter Entscheidungsfindung werden beide Teilaufgaben von einem AwS durchgeführt.

3.5.3 Latenzen im Kontext prozessorientierter Data-Warehouse-Systeme

Gerade im verhaltensorientierten pDWH-Szenario ist die Reaktion auf Ereignisse in Geschäftsprozessen mit geringer zeitlicher Verzögerung (*Latenz*) wichtig (Russom 2011). Zur Gestaltung einer Reaktion werden von Russom (2011) vier Schritte vorgeschlagen, die auch auf das weniger zeitkritische strukturorientierte pDWH-Szenario Anwendung finden können:

1. *Wahrnehmung (recognize)* eines relevanten Ereignisses in einem Geschäftsprozess;
2. *Erfassen (capture)* von Informationen zu dem Ereignis und auslösenden Entitäten sowie *Aufbereitung* eines Analysekontextes, ggf. basierend auf historisierten Daten;
3. *Analyse (analyze)* der erfassten und aufbereiteten Informationen hinsichtlich der Ableitung bestmöglicher Maßnahmen;
4. Zeitnahe und angemessene *Reaktion (react)* auf das Ereignis durch (teil-)automatisierte Einflussnahme auf die Quelle.

Entlang dieser vier Schritte kann die Latenz vom Auftreten eines Ereignisses bis zu einer adäquaten Reaktion in vier Intervalle aufgeteilt werden. Diese sind z.T. in Abb. 3.13 dargestellt. Als *Recognition latency* wird das zeitliche Intervall zwischen Eintreten eines Ereignisses und dessen Wahrnehmung (vgl. Schritt 1, nicht dargestellt) verstanden. *Capture Latency (Bereitstellungslatenz)* beschreibt die Zeitspanne zur Erfassung und Aufbereitung von Informationen zur Analyse (vgl. Schritt 2) (Hackathorn 2004). Als *Analysis Latency (Analyselatenz)* wird die für Schritt 3, Analyse der Informationen und Bereitstellung der Ergebnisse, benötigte Zeitspanne verstanden (Hackathorn 2004). *Decision Latency (Entscheidungslatenz)* beschreibt das Intervall zur Entscheidungsfindung sowie Gestaltung und Initiierung einer Reaktion (Schritt 4) (Hackathorn 2004).

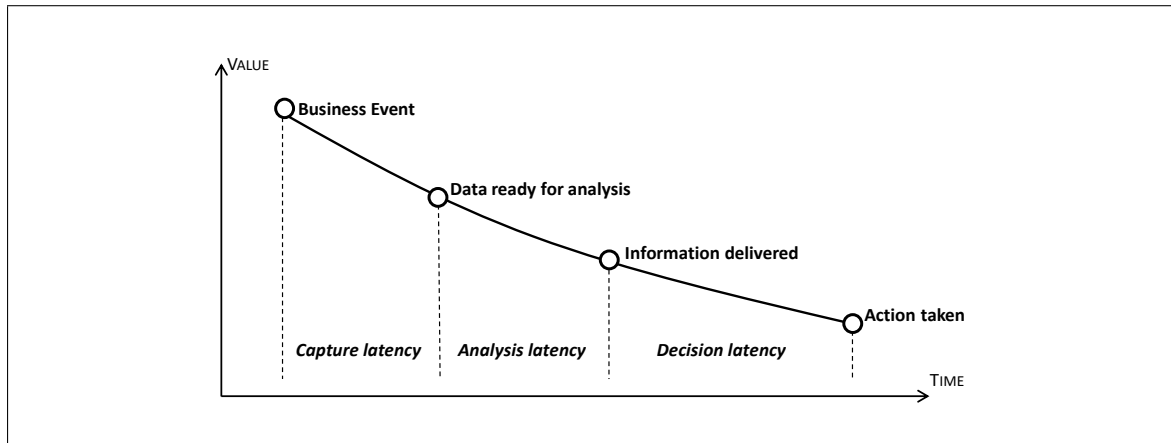


Abb. 3.13: Zusammenhang von Wertbeitrag und Latenz der Reaktion auf Prozessereignisse (*Time-Value-Curve* nach Hackathorn 2004, angepasst von Russom 2011)

In Abb. 3.13 sind die beschriebenen Formen der Latenz auf einer abfallenden Kurve (sog. *Time-Value-Curve* nach Hackathorn (2004)) dargestellt. Durch die Kurve wird das Verhältnis zwischen geschäftlichem Wertbeitrag und Latenz zur Reaktion auf Geschäftsprozessereignisse beschrieben. Die Grundannahme bzgl. des Kurvenverlaufs kann wie folgt zusammengefasst werden: Je länger die Latenz zwischen Ereignis und adäquater Reaktion, desto höher der Wertverlust für das Unternehmen. Der Verlauf der Kurve ist nach Hackathorn (2004) für viele, v.a. kundenbezogene Szenarien gültig. Jedoch sind auch Szenarien mit variierenden, z.T. nur bedingt vorhersagbaren Kurvenverläufen realistisch. Eine hohe Ausschöpfung der Potenziale gerade im operativen, eher zeitkritischen pDWH-Szenario ergibt sich durch Abstimmung von Latenzen und Technologien mit dem jeweils gültigen Kurvenverlauf (Hackathorn 2004, Russom 2011).

4 Einführung der durchgehenden Fallstudie e-Car Net

Kapitelzusammenfassung: Das Kapitel führt das Beispielunternehmen *e-Car Net* mit dem Teil-Szenario *Mobility Provision* ein. Dieses dient in den nachfolgenden Kapiteln zur Erarbeitung und Demonstration von Ergebnisartefakten. Der Dienstleistungsprozess *Mobility Provision* beschreibt die Vermietung von e-Cars. Die Spezifikation erfolgt auf Basis des Semantischen Objektmodells (SOM). Hierzu wird zunächst allgemein die Geschäftsprozessmodellierung mit SOM eingeführt (Abschn. 4.1) und anschließend auf das Szenario *Mobility Provision* angewendet (Abschn. 4.2 u. 4.3). Abschn. 4.4 grenzt schließlich das zu entwickelnde Anwendungssystem zur Automatisierung des Vertriebsprozesses anhand seiner funktionalen und nicht-funktionalen Anforderungen ab.

4.1 Grundlagen der Geschäftsprozessmodellierung mit SOM

Die Modellierung des Geschäftsprozesses der Fallstudie erfolgt auf Basis der SOM-Methodik. Zunächst werden die dazu benötigten Grundlagen eingeführt. Bei der Spezifikation von Geschäftsprozessen unterstützt die SOM-Methodik drei Sichten (Ferstl & Sinz 1995):

- **Leistungssicht:** In der Leistungssicht ist der Leistungsaustausch eines Geschäftsprozesses strukturorientiert beschrieben. Ein Geschäftsprozess übergibt Leistungen an beauftragende Geschäftsprozesse, beauftragt aber auch seinerseits Prozesse mit der Erstellung von Leistungen (Güter, Dienstleistungen oder Zahlungen).
- **Lenkungssicht:** In der Lenkungssicht wird, ebenfalls strukturorientiert, die Koordination betrieblicher Objekte durch Transaktionen zur Leistungserstellung und -übergabe beschrieben. Die Koordination erfolgt entweder hierarchisch oder auch nicht-hierarchisch.

- **Ablaufsicht:** Mit der Ablaufsicht wird das Verhalten eines Geschäftsprozesses als ereignisgesteuerter Ablauf von Vorgängen (Durchführung von Aufgaben betrieblicher Objekte) beschrieben.

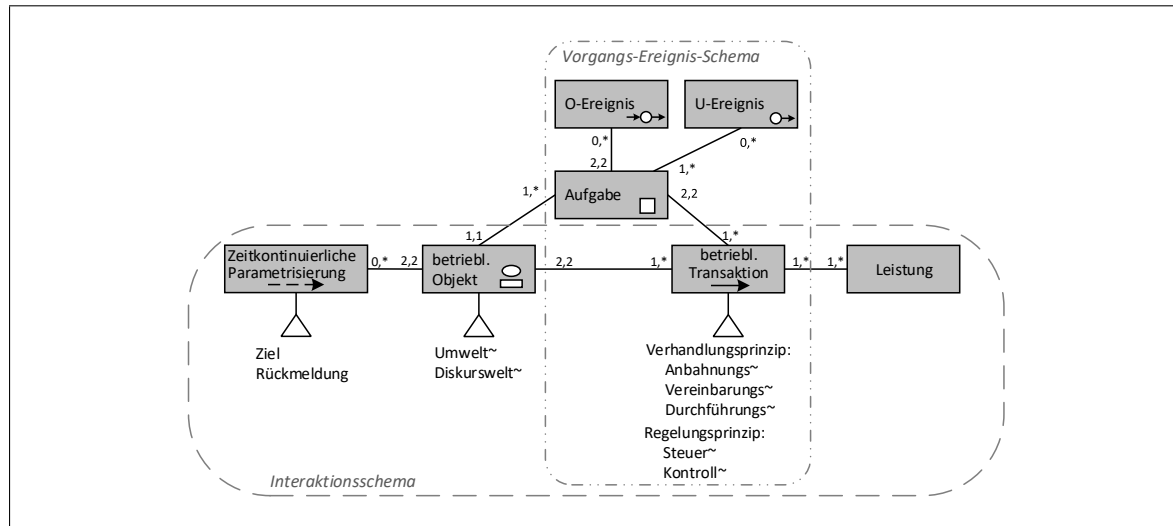


Abb. 4.1: Metamodell und Sichten für SOM-Geschäftsprozessmodelle (Ferstl & Sinz 1995, Ferstl & Sinz 2013, S. 219)

Die Modellierung von Geschäftsprozessen im Rahmen der SOM-Methodik erfolgt, basierend auf einem integrierten Metamodell aus struktur- und verhaltensorientierter Sicht. Das Metamodell ist zusammen mit den Sichten in Abb. 4.1 dargestellt. Die Modellierung der Struktur (Leistungs- und Lenkungssicht) eines Geschäftsprozesses erfolgt mit dem *Interaktionsschema* (IAS), die dazu konsistente Verhaltensspezifikation (Ablaufsicht) mit dem *Vorgangs-Ereignis-Schema* (VES). Das IAS beschreibt einen Geschäftsprozess als verteiltes System lose gekoppelter, autonomer betrieblicher Objekte aus Diskurs- und Umwelt, die über betriebliche Transaktionen interagieren (Lenkungsnachrichten und Leistungspakete) (Ferstl & Sinz 1995). Transaktionen dienen der Übertragung oder Koordination von Leistungen und können diesbezüglich näher typisiert werden. Zur Ausführung der betrieblichen Transaktionen dienen Aufgaben, je eine in Quell- und Zielobjekt (Ferstl & Sinz 2013, S. 219). In der verhaltensorientierten Spezifikation (VES) werden diese Aufgaben, durchgeführt als Vorgänge, über Ereignisse (*objektinterne Ereignisse*) gekoppelt. Ein Vorgang kann ausgeführt werden, wenn die entsprechenden Vorereignisse vorliegen, und produziert seinerseits ein bis beliebig viele Nachereignisse. Umweltereignisse dienen beispielsweise der Spezifikation von Zeitpunkten oder dem Ablauf von Zeitintervallen und stellen nicht in Transaktionen erfasste Ereignisse dar (Ferstl & Sinz 1995).

Die Geschäftsprozessmodellierung mit SOM folgt hierarchischen Zerlegungsschritten. In Abb. 4.2 sind Ersetzungsregeln für Objekte und Transaktionen nach Ferstl & Sinz (1995) und Ferstl & Sinz (2013, S. 208) dargestellt. Zusammen mit dem Metamodell spezifizieren diese Regeln Syntax und formale Semantik eines SOM-Geschäftsprozessmodells (Ferstl & Sinz 1995). Durch Anwendung der Regeln kann die Lenkung von Leistungserstellung und -übergabe aufgedeckt werden. Regel (1) beschreibt die Zerlegung betrieblicher Objekte nach dem *Regelungsprinzip*. Die resultierenden Objekte (Regel- und Regelstreckenobjekt) stehen in einer *hierarchischen* Beziehung und koordinieren sich über *Steuerungs- und Kontrolltransaktionen*. T_S stellt die Steuerungstransaktion ausgehend vom Regler (Planungsaufgabe) hin zur Regelstrecke (Durchführungsaufgabe) dar. Die entgegengesetzte Kontrolle (T_K) ist optional (Ferstl & Sinz 1995). Die weiteren Regeln zur Objektzerlegung beschreiben die Zerlegung eines Objektes in zwei Teilobjekte mit optionaler Transaktion (Regel 2), die Spezialisierung von Objekten (Regel 3) sowie die zeitkontinuierliche Parametrisierung von Aufgaben betrieblicher Objekte (Regel 5) (Ferstl & Sinz 2013, S. 208).

Ersetzungsregeln zur Objektzerlegung:		Regel Nr.		
O	$::= \{ O', O'', T_S(O', O''), [T_K(O'', O')] \}$	(1)		
O	$::= \{ O', O'', [T(O', O'')] \}$	(2)		
O	$::= \{ \text{spez } O' \}^+$	(3)		
$O' \mid O''$	$::= O$	(4)		
O	$::= \{ O', \{ O'', P_2(O', O''), [P_R(O'', O')] \}^+ \}$	(5)		
Ersetzungsregeln zur Transaktionszerlegung:				
$T(O, O')$	$::= [[T_a(O, O') \text{ seq }] T_v(O', O) \text{ seq }] T_d(O, O')$	(6)		
T_x	$::= T_x \{ \text{seq } T_x'' \}^+ \mid T_x \{ \text{par } T_x'' \}^+$ (für $x = a, v, d, s, k$)	(7)		
T_x	$::= \{ \text{spez } T_x'' \}^+$ (für $x = a, v, d, s, k$)	(8)		
$T_a \mid T_v \mid T_d$	$::= T$	(9)		
Legende zur verwendeten Notation (Backus-Naur-Form BNF)				
$::=$	Ersetzung	$\{ \}$ Menge	seq	sequenzielle Beziehung
$[]$	Option	$\{ \}^+$ Wiederholung (1,*)	par	parallele Beziehung
$ $	Alternative	$\{ \}^*$ Wiederholung (0,*)	spez	Spezialisierung

Abb. 4.2: Ersetzungsregeln für SOM-Geschäftsprozessmodelle (Ferstl & Sinz 1995, Ferstl & Sinz 2013, S. 208)

Das *Verhandlungsprinzip* (Regel 6) beschreibt die *nicht-hierarchische Koordination* gleichrangiger betrieblicher Objekte. Eine Transaktion zwischen diesen Objekten wird in die sequentiellen Transaktionen *Anbahnung* (optional), *Vereinbarung* (optional) und *Durchführung* zerlegt. In der *Anbahnungsphase* tauschen die be-

trieblichen Objekte unverbindlich Informationen bzgl. Leistungen und Rahmenbedingungen zur Leistungsbereitstellung aus. Während der *Vereinbarungsphase* erfolgt die für beide Seiten verbindliche Vereinbarung zum Leistungsaustausch, der in der *Durchführungsphase* erfolgt (Ferstl & Sinz 1995). Die weiteren Regeln zur Transaktionszerlegung spezifizieren die typerhaltende Zerlegung in sequenzielle oder parallele Transaktionen (Regel 7) sowie die Spezialisierung von Transaktionen (Regel 8) (Ferstl & Sinz 2013, S. 208). Die Regeln (4) und (9) sind zur wiederholten Anwendung der beschriebenen Ersetzungsregeln notwendig (Ferstl & Sinz 2013, S. 208).

4.2 Einführung in die Fallstudie e-Car Net

Die Fallstudie *e-Car Net* basiert auf der Arbeit von Leunig et al. (2011, S. 15ff.). Abb. 4.3 zeigt die Leistungssicht des Gesamtszenarios *e-Car Net* als IAS. Als Teil-Szenarien sind *Produktion*, *Absatz*, *Mobile Maintenance* sowie *Mobility Provision* spezifiziert.

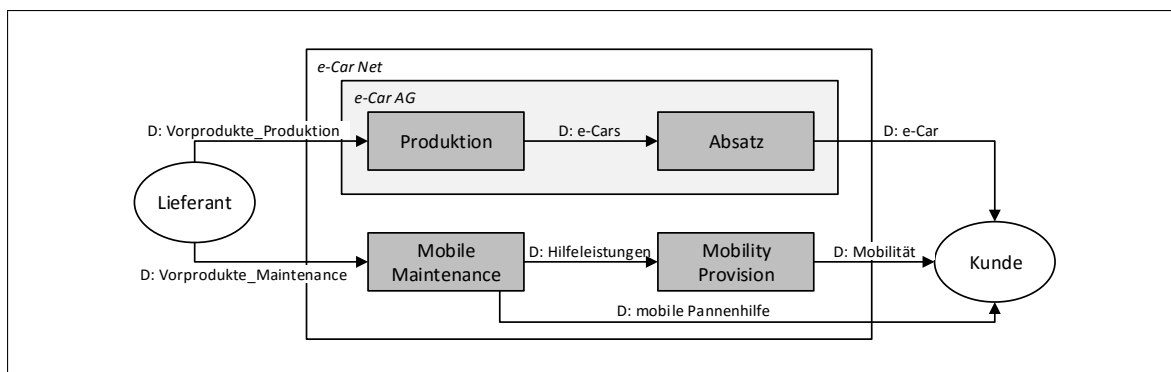


Abb. 4.3: Geschäftsprozessmodell der Fallstudie *e-Car Net* (Leunig et al. 2011) (eigene Darstellung)

Das Unternehmen *e-Car AG* produziert und vermarktet e-Cars (Elektroautos). Die Vorprodukte werden über Lieferanten bezogen. Im Bereich *Produktion* erfolgt die Fertigung der e-Cars, im Bereich *Absatz* die Vermarktung an den Endkunden. Weitere Leistungen um das Objekt e-Car werden von externen Dienstleistern erbracht. Zusammen mit der *e-Car AG* bilden die Dienstleister aus den Bereichen *Mobility Provision* sowie *Mobile Maintenance* das Wertschöpfungsnetzwerk *e-Car Net*. Der Bereich *Mobility Provision* beschreibt die Vermietung von e-Cars. *e-Car Net* unterhält hierzu Servicestationen in größeren Städten. *Mobile Maintenance* bezeichnet den Bereich der mobilen Instandhaltung der Elektroautos. Wartungsleis-

tungen werden gegenüber Endkunden (mobile Pannenhilfe) sowie gegenüber von Servicestationen (Hilfeleistungen, z.B. Wartung der Flotte) erbracht. Auch *Mobile Maintenance* unterhält entsprechende Lieferanten-Beziehungen zum Bezug von Vorprodukten.

4.3 Einführung in den Geschäftsprozess *Mobility Provision*

Der Geschäftsprozess *Mobility Provision* dient als Grundlage zur Entwicklung und Demonstration von Teilergebnissen der nachfolgenden Kernkapitel. Unter Berücksichtigung dieser Zielsetzung erfolgen Transaktions- und Objektzerlegung nach eigener Anwendung der Regeln von Ferstl & Sinz (2013, S. 207ff.). Teile sind bestehenden Zerlegungen von Leunig et al. (2011, S. 27ff.) sowie von Krücke & Sinz (2011, S. 300ff.) entnommen.

4.3.1 Struktursicht auf den Geschäftsprozess *Mobility Provision*

Die Leistung von *Mobility Provision* gegenüber seinen Kunden lässt sich als Mobilität durch Vermietung von e-Cars beschreiben. In Abb. 4.4 sind die durchgeführten Zerlegungsschritte der Transaktion *D: Mobilität* sowie des betrieblichen Objektes *Mobility Provision* in Form eines Zerlegungsbaums dokumentiert. Das aus der Zerlegung resultierende IAS (Lenkungsicht) ist in Abb. 4.5 dargestellt und wird nachfolgend erläutert.

Das betriebliche Objekt *Vertriebskanal* stellt die Schnittstelle zum Kunden dar. In der Fallstudie nach Leunig et al. (2011, S. 27f.) sind heterogene Vertriebskanäle vorgesehen. Für die vorliegende Fallstudie werden als mögliche Ausprägungen der Vertrieb über *Call Center* sowie der *Online-Vertrieb* übernommen. Aufgabe des Vertriebskanals ist die Durchführung der Leistungsvereinbarung mit dem Kunden bestehend aus Anfrage des Kunden, Erstellen eines individuellen Angebots sowie Entgegennahme und Bestätigung der Buchung. Hierzu werden die Dienste zur Prüfung verfügbarer e-Cars sowie zur Vorreservierung eines e-Cars des betrieblichen Objektes *Flottenlenkung* in Anspruch genommen. Der Dienst zur Buchungsdurchführung wird durch das betriebliche Objekt *Auftragsverwaltung* bereitgestellt. Das betriebliche Objekt *Auftragsverwaltung* ist für die Lenkung des Buchungsablaufs bis zum Zahlungseingang verantwortlich. Hierzu zählt zunächst

die verbindliche Reservierung eines e-Cars gegenüber der *Flottenlenkung* mit Bereitstellung der Buchungsunterlagen über den *Vertriebskanal* hin zum Kunden. Weiterhin wird die Abrechnung (Aufgabe des *Finanzwesens*) auf Basis eines Fahrtenprotokolls beauftragt und überwacht. Die Bereitstellung sowie die Rückgabe des e-Cars sind Aufgaben des Leistungsobjektes *Flottenbereitstellung*. Die hierarchische Koordination mit der Anweisung zur e-Car-Bereitstellung erfolgt durch das Lenkungsobjekt *Flottenlenkung*, der auch Rückgabe und Fahrtenprotokoll zurückgemeldet werden.

<p>TRANSAKTIONSZERLEGUNG: D: Mobilität V: Buchung_Mobilität Vseq: Anfrage Vseq: individuelles Angebot Vseq: Buchung Vseq: Buchungsbestätigung D: Nutzung_Mobilität Dseq: Mobilitätsleistung Dseq: Bereitstellung e-Car V: Anforderung e-Car D: e-Car Dseq: Rückgabe e-Car Dseq: Mobilitätsabrechnung V: Rechnung D: Zahlung</p>	<p>OBJEKTZERLEGUNG: Mobility Provision Vertrieb Buchungsdurchführung Auftragsverwaltung Finanzwesen D: Zahlungsabwicklung V: Forderung D: Zahlungsmeldung Vertriebskanal D: Buchungsabwicklung V: Buchungsauftrag D: Buchungsdokumentation Mobilitätsleistung Flottenlenkung Flottenbereitstellung S: Anweisung_Bereitstellung K: Rückmeldung_Bereitstellung D: Flottenverwaltung D: Verfügbarkeit V: Verfügbarkeitsprüfung D: Verfügbarkeitsliste D: Vorreservierung V: Auftrag_Vorreservierung D: Bestätigung_Vorreservierung D: Reservierung V: Auftrag_Reservierung D: Bestätigung_Reservierung D: Fahrtenprotokoll</p>
--	---

Abb. 4.4: Protokoll der Transaktions- und Objektzerlegung zum Szenario *Mobility Provision*

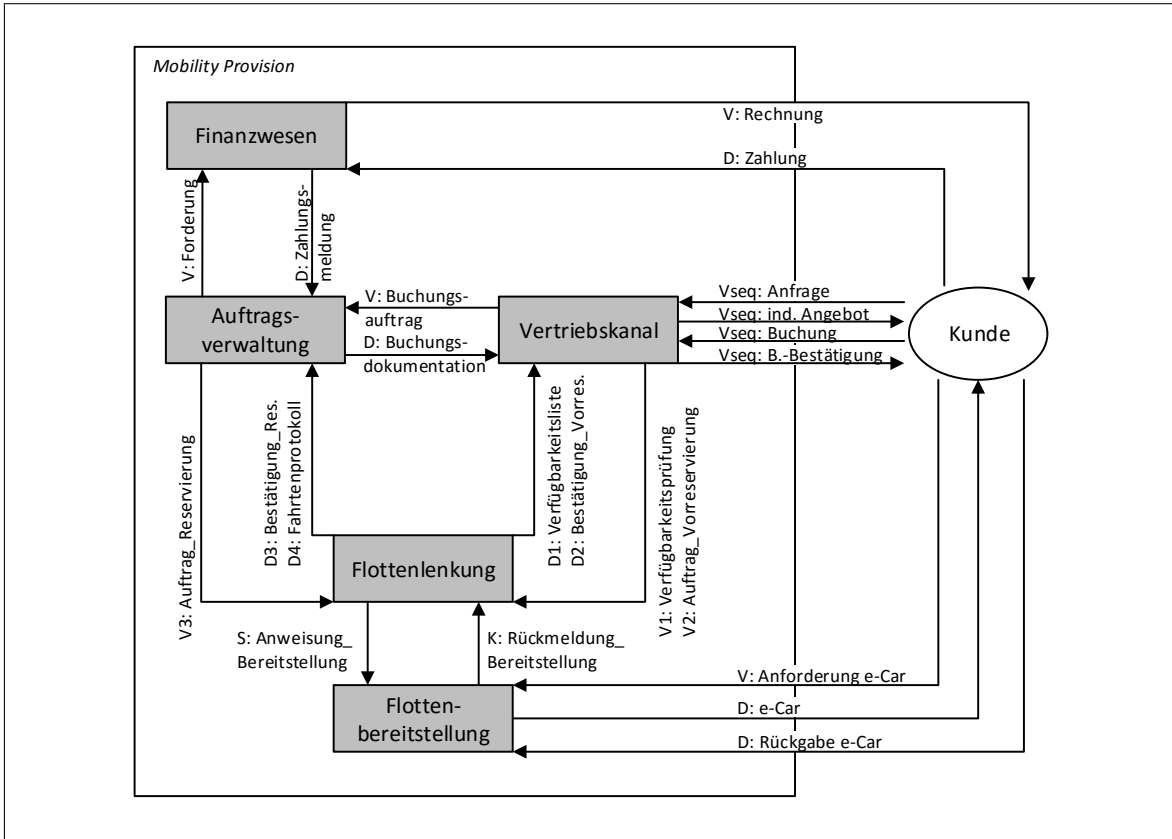


Abb. 4.5: Interaktionsschema zum Szenario Mobility Provision

4.3.2 Verhaltenssicht auf den Geschäftsprozess Mobility Provision

Das IAS des Geschäftsprozesses wird nachfolgend anhand des VES um eine ablauforientierte Sicht ergänzt. In Abb. 4.6 ist dazu der Ablauf bis zur verbindlichen Buchung eines e-Cars dargestellt. Nach Anfrage eines Kunden wird, initiiert durch den *Vertriebskanal*, beim betrieblichen Objekt *Flottenlenkung* die Verfügbarkeit eines e-Cars ermittelt, die Vorreservierung veranlasst und ein kundenindividuelles Angebot erstellt. Dieses kann der Kunde durch verbindliche Buchung annehmen.

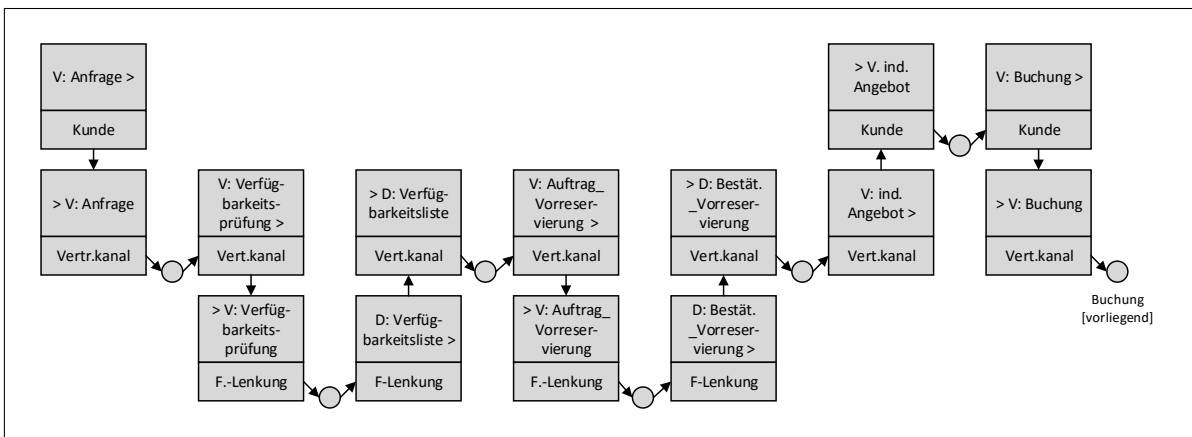


Abb. 4.6: Vorgangs-Ereignis-Schema zur Buchung im Szenario Mobility Provision

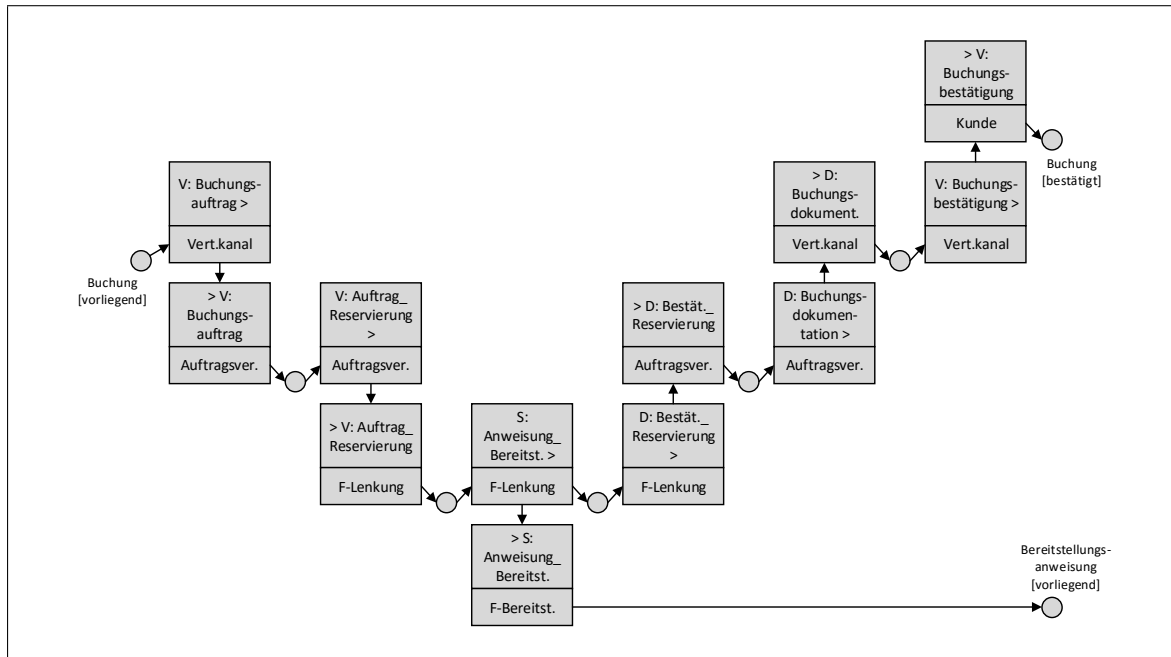


Abb. 4.7: Vorgangs-Ereignis-Schema zur Buchungsbestätigung im Szenario *Mobility Provision*

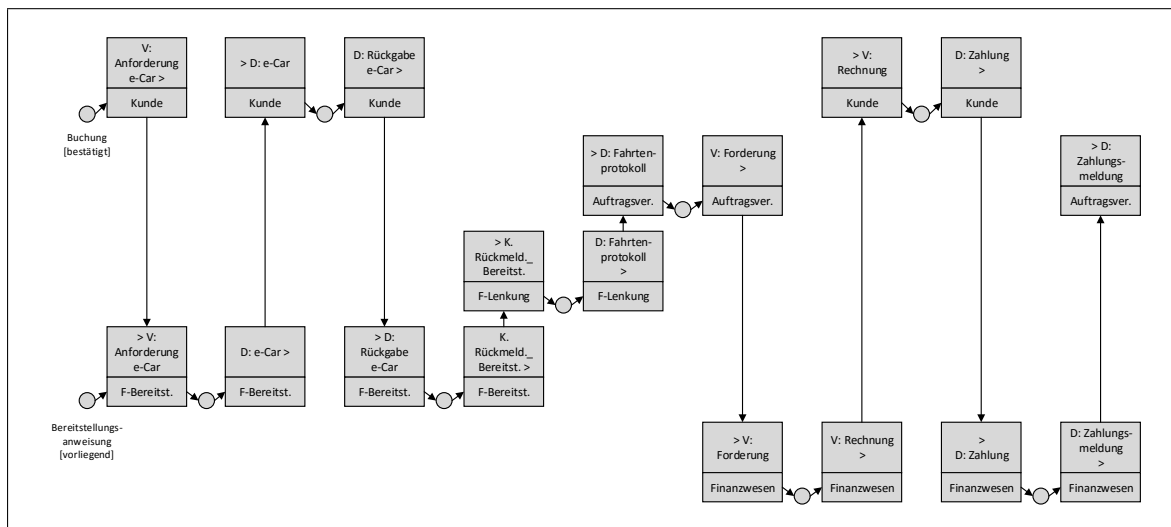


Abb. 4.8: Vorgangs-Ereignis-Schema zur Leistungserbringung im Szenario *Mobility Provision*

In Abb. 4.7 ist die interne Abwicklung der Buchung bis hin zur Bestätigung gegenüber dem Kunden dargestellt. Durch die *Auftragsverwaltung* wird die Reservierung des e-Cars bei der *Flottenlenkung* beauftragt. Sobald diese das betriebliche Objekt zur *Flottenbereitstellung* entsprechend beauftragt hat, erhält die *Auftragsverwaltung* eine Bestätigung, die über den *Vertriebskanal* an den Kunden weitergegeben wird. In Abb. 4.8 ist die eigentliche Durchführung der Leistung beschrieben. Voraussetzung für die Übernahme des Fahrzeuges ist die übermittelte Buchungs-

bestätigung an den Kunden sowie die Anweisung an die *Flottenbereitstellung*. Nach Aus- und Rückgabe des Fahrzeugs an bzw. durch den Kunden wird das Fahrtenprotokoll an die *Auftragsverwaltung* übermittelt. Dieses bildet die Grundlage für die Forderungen gegenüber dem Kunden im Teil-Szenario *Mobility Provision*. Das betriebliche Objekt *Finanzwesen* wird abschließend mit Rechnungsstellung und Zahlungsabwicklung beauftragt.

4.4 Automatisierung des Mobility Provision-Vertriebsprozesses

Das zu entwickelnde AwS wird anhand der zu automatisierenden Aufgaben sowie der für die Fallstudie relevanten nicht-funktionalen Anforderungen eingeführt.

Funktionale Abgrenzung Der Teil-Prozess *Vertrieb* des Geschäftsprozesses *Mobility Provision* soll durch ein AwS in geeigneter Form unterstützt werden. Die Verwaltung der e-Car-Flotte sowie die Abrechnung sind Aufgaben benachbarter AwS. Zu automatisieren sind die Aufgaben der betrieblichen Objekte *Vertriebskanal* und *Auftragsverwaltung*. Als Zielsetzung wird ein möglichst hoher Automatisierungsgrad angestrebt. Die betrieblichen Aufgaben werden, zusammen mit den möglichen Automatisierungsgraden ([a]utomatisierbar), in Tab. 4.1 dargestellt. Bei Normalverhalten wird die volle Automatisierbarkeit unterstellt.

Aufgaben des betrieblichen Objekts Vertriebskanal	Aufgaben des betrieblichen Objekts Auftragsverwaltung
Anfrage empfangen [a] Verfügbarkeitsprüfung senden [a] Verfügbarkeitsliste empfangen [a] Auftrag_Vorreservierung senden [a] Bestät._Vorreservierung empfangen [a] individuelles Angebot senden [a] Buchung empfangen [a] Buchungsauftrag senden [a] Buchungsdokumentation empfangen [a] Buchungsbestätigung senden [a]	Buchungsauftrag empfangen [a] Auftrag_Reservierung senden [a] Bestätigung_Reservierung empfangen [a] Buchungsdokumentationen senden [a] Fahrtenprotokoll empfangen [a] Forderung senden [a] Zahlungsmeldung empfangen [a]

Tab. 4.1: Automatisierungsgrade der Aufgaben von Vertriebskanal und Auftragsverwaltung

Zwischen den Vertriebskanälen e-Shop und Call Center wird hinsichtlich der Automatisierungsgrade der Aufgaben nicht unterschieden. Der Unterschied liegt in der Automatisierung der Transaktionen. Während im e-Shop von vollautomatisierbaren Transaktionen ausgegangen wird, können diese im Call Center auch nicht-automatisierbar sein. So erfolgt die Angebotsübermittlung beispielsweise mündlich durch einen Call Center-Mitarbeiter.

Neben der Durchführung des Vertriebsprozesses sind auch nachfolgend beschriebene Entscheidungsszenarien zu berücksichtigen. Ein möglichst umfassender Datenbestand soll hierzu als Grundlage dienen und die Qualität der Entscheidungen positiv beeinflussen.

- **Prozessgestaltung Vertriebskanal:** Hinsichtlich zeit- und kosteneffizienter Abläufe in Vertriebskanälen sollen eine permanente Analyse abgeschlossener Prozessinstanzen vorgenommen sowie Ansatzpunkte zur Neugestaltung aufgedeckt werden.
- **Angebotserweiterung:** Zur Erhöhung der Kundenbindung sowie zur Maximierung der Rentabilität sind Kunden während der Leihphase Erweiterungen ihrer Buchung anzubieten. Anhand des Mietverhaltens vergleichbarer Kunden soll über die Angebotserweiterungen entschieden werden.
- **Auslastungsgestaltung:** Es soll flexibel auf die jeweilige Flottenauslastung (z.B. zu hoch oder zu niedrig) an den jeweiligen Leihstationen reagiert werden können. Auf Basis abgeschlossener sowie laufender Buchungen soll entweder über Anpassung von Kundenangeboten oder Serviceprozesse die Auslastung an Leihstationen optimiert werden.

Nicht-funktionale Anforderung Die *Wartbarkeit* stellt ein wesentliches Qualitätsmerkmal für das zu realisierende AwS dar. Die Anpassung an veränderte bzw. neue fachliche Anforderungen soll flexibel möglich sein. Nach Balzert (2008, S. 464) (basierend auf der Norm ISO/IEC 9126-1:2001¹) ist die Wartbarkeit durch die Teilmerkmale Analysierbarkeit, Modifizierbarkeit, Stabilität und Testbarkeit charakterisiert. Möglichst niedriger Wartungsaufwand sowie eine hohe Stabilität bzgl. unerwarteter Seiteneffekte der Wartung sind für das zu realisierende AwS anzustreben.

¹seit 2011 als ISO/IEC 25010:2011 geführt

5 Referenzarchitektur prozessorientierter operativ-dispositiver Systeme

Kapitelzusammenfassung: Die Konzeption einer Referenzarchitektur für prozessorientierte operativ-dispositive AwS bildet einen Teil des Untersuchungsgegenstandes der Arbeit. Ausgehend von einer Anforderungsanalyse (Abschn. 5.1) werden nachfolgend etablierte DWH-Architekturen sowie verwandte Architekturkonzepte anhand der extrahierten Anforderungen gespiegelt (Abschn. 5.2) und die Notwendigkeit einer Referenzarchitektur aufgezeigt (Abschn. 5.2). Die Konzeption der Referenzarchitektur erfolgt zweistufig (Abschn. 5.3): In einem ersten Schritt wird mit der *flexiblen Data-Mart-Architektur* eine Referenzarchitektur für pDWH-Systeme vorgestellt. Im zweiten Schritt wird, basierend auf dem Paradigma der Komponentenorientierung, die Integration dieser mit dem operativen und dem EU-System betrachtet. Diese Ausbaustufe bildet die Referenzarchitektur für operativ-dispositiven AwS. Für Komponenten des pDWH-Systems werden zudem eine generische funktionsorientierte Spezifikation der Innensicht (Abschn. 5.4) sowie Integrationsmöglichkeiten mit unternehmensweiten DWH-Systemen (Abschn. 5.5) vorgestellt. Die Spezifikation von Datenstrukturen als Teil der *flexiblen Data-Mart-Architektur* ist in Kap. 6 ausgelagert. Zur Begründung und Demonstration der Ergebnisse findet die Fallstudie *Mobility Provision* (Kap. 4) Anwendung.

Das Kapitel basiert in Teilen auf den Veröffentlichungen Benker (2013, 2015).

5.1 Anforderungen an die Architektur operativ-dispositiver Anwendungssysteme

Zunächst werden Anforderungen an die Architektur operativ-dispositiver AwS, im Speziellen des zentralen pDWH-Systems spezifiziert. Diese werden anhand der Literatur sowie auf Basis der in Kap. 3.5.2 vorgestellten pDWH-Szenarien (*verhaltens-* u. *strukturorientiert*) abgeleitet.

Anforderung A1 *Datenbereitstellung und Reaktion mit angemessener Latenz:* Zur aktiven Steuerung laufender Prozessinstanzen im verhaltensorientierten pDWH-Szenario ist eine angemessene Latenz zur Datenbereitstellung, Entscheidungsfindung und Reaktion entscheidend, um flexibel auf Ereignisse in Geschäftsprozessen reagieren zu können. Nach Russom (2011) geht in vielen Situationen mit verzögerter Reaktion auf bestimmte Ereignisse auch der zunehmende finanzielle Verlust des Unternehmens einher. Die Datenbereitstellung muss nicht zwingend in Echtzeit stattfinden. Wichtig ist, dass zum Zeitpunkt der Entscheidungsfindung ein aktueller und konsistenter Datenbestand für die jeweiligen Nutzer vorliegt (Russom 2011). Für das strukturorientierte Szenario ist die Latenz der Entscheidungsfindung unkritisch. Maßnahmen zur effizienteren und innovativeren Prozessgestaltung werden nicht für laufende Prozessinstanzen, sondern mittelfristig, basierend auf Daten abgeschlossener Prozesse, realisiert.

Anforderung A2 *Informationsversorgung unterschiedlicher Nutzergruppen:* Durch die Realisierung eines pDWH-Systems wird einem größeren Nutzerkreis der Zugang zu entscheidungsrelevanten Informationen ermöglicht als dies bei „klassischen“, themenorientierten DWH-Systemen der Fall ist (Eckerson 2007). Letztere stehen einem kleinen Nutzerkreis im strategischen oder taktischen Management zur Verfügung. Nutzer von pDWH-Systemen können während der Prozessausführung (verhaltensorientiertes Szenario) hingegen Mitarbeiter und Verantwortliche auf operativer Ebene (aus unterschiedlichen Abteilungen), aus dem taktischen und strategischen Management, Analysten aber auch Geschäftspartner wie Kunden und Lieferanten sein (ElegantJ BI 2008, White 2006, Hrach & Alt 2012). Für das strukturorientierte Szenario leitet sich die Nutzergruppe aus der Zielsetzung der Prozessanalyse und -gestaltung ab (Felden et al. 2010) und ist als vergleichsweise klein zu bezeichnen (Linden et al. 2011). Jede der genannten Nutzergruppen verfolgt mit der Nutzung des pDWH-Systems eine eigene, prozessbezogene Zielsetzung, die es durch Bereitstellung bedarfsgerechter Informationen zu unterstützen gilt.

Anforderung A3 *Flexible Anpassbarkeit:* DWH-Systeme unterliegen im Zeitverlauf inhaltlichen (bezogen auf das Datenschema) sowie technischen Veränderungen und sind somit nicht stabil (Bauer & Günzel 2013, S. 228). Ursächlich dafür können beispielsweise Anpassungen an operativen Quellsystemen sein oder

neue bzw. veränderte Anforderungen, die sich aus der täglichen Arbeit mit den DWH-Systemen ergeben (Bauer & Günzel 2013, S. 233f.). Beide Ursachen verstärken sich mit zunehmender Dynamik der Unternehmensumwelt. Nach Dayal et al. (2009) ist, bezogen auf den ETL-Prozess, gerade im verhaltensorientierten Szenario von rapiden Veränderungen der Anforderungen auszugehen. Diese Aussage kann auch bzgl. der weiteren Bestandteile der DWH-Architektur (vgl. Kap. 3.2) verallgemeinert werden. Zudem ist bei dynamischen AwS mit vielen verschiedenen Nutzergruppen nicht nur während der Entwicklungs-, sondern auch während der Betriebsphase von einer kontinuierlichen Anpassung der Anforderungen auszugehen (Casati et al. 2007). Hinsichtlich neu-ausgerichteter Geschäftsprozesse oder veränderter Anforderungen sind neben operativen Systemen auch die DWH-Systeme zur Bereitstellung von entscheidungsrelevanten Prozessdaten flexibel anpassbar zu gestalten. Der optimale Grad der flexiblen Anpassbarkeit eines Systems ist dabei von der Realisierung weiterer, mit einem System assoziierter Ziele beeinflusst. Diese können funktionaler (Sachziel), nicht-funktionaler (z.B. Verfügbarkeit) oder auch betriebswirtschaftlicher (z.B. Wirtschaftlichkeit) Natur sein (Mertens 2005, Wagner et al. 2011b).

Anforderung A4 *Integration mit bidirektionalem Datentransfer:* Für operative Entscheidungssituationen (verhaltensorientiertes pDWH-Szenario), also zur aktiven Steuerung von Prozessinstanzen, wird bei Russom (2011) die Anforderung erhoben, operative und analytische Systeme enger zu integrieren, um einen schnelleren, häufigeren und v.a. bidirektionalen Datentransfer zwischen diesen zu ermöglichen. Entscheidungsrelevante und vom DWH-System bereitgestellte Informationen können wiederum als Input für operative Prozesse dienen (Bucher & Dinter 2008a). Um neben der Latenz zur Datenbereitstellung auch die Latenz zur Entscheidungsfindung und Reaktion zu reduzieren, können hier Software-Komponenten zur automatisierten Entscheidungsfindung integriert werden (White 2006). Hinsichtlich der aktiven Steuerung von Geschäftsprozessen ist es notwendig, Ergebnisse personell oder automatisiert getroffener Entscheidungen zurück an das operative System zu transferieren.

Anforderung A5 *Integration mit bestehenden DWH-Lösungen:* BI-Lösungen zur strategischen und taktischen Entscheidungsunterstützung sind i.d.R. prozessübergreifend an unternehmensweiten Analysethemen orientiert. Diese Lösungen sind

nicht nur für große Unternehmen relevant, nach Bange et al. (2012) haben 2011/12 auch 83 Prozent der befragten kleinen und mittelgroßen Unternehmen bereits BI-Lösungen im Einsatz. Prozessorientierte, auf einzelne Geschäftsprozesse ausgerichtete DWH-Systeme müssen demzufolge i.d.R. in eine bestehende DWH-Systemlandschaft eingebettet und unter Berücksichtigung der Aussagenkonsistenz mit unternehmensweiten DWH-Systemen integriert werden (Baars & Sun 2013).

Die *Datenbereitstellung mit angemessener Latenz*, die *Informationsversorgung verschiedener Nutzergruppen* mit sich stetig verändernden Anforderungen sowie die enge *Integration mit operativen Systemen* im Rahmen operativ-dispositiver AwS stellen z.T. erweiterte, z.T. neuartige Anforderungen an DWH-Systeme dar. Im nachfolgenden Kap. 5.2 wird überprüft, inwieweit bestehende Architekturkonzepte diese Anforderungen bereits berücksichtigen.

5.2 Stand der Literatur: Architekturen prozessorientierter Data-Warehouse-Systeme

Das pDWH-System stellt das zentrale Teilsystem operativ-dispositiver AwS zur Realisierung der Anforderungen aus Abschn. 5.1 dar. Zunächst werden aus diesem Grund etablierte Architekturformen von DWH-Systemen auf ihre Eignung zur Anwendung in operativ-dispositiven Systemen hin überprüft (Abschn. 5.2.1). Zudem werden in Abschn. 5.2.2 Ansätze verwandter Forschungsarbeiten anhand der unterstützten Anforderungen an ein pDWH-System zur Realisierung des struktur- und des verhaltensorientierten Szenarios vorgestellt.

5.2.1 Diskussion der Eignung klassischer Data-Warehouse-Architekturen

Die Diskussion von etablierten Architekturkonzepten themenorientierter DWH-Systeme kann ausschließlich auf abstrakter Ebene geführt werden. Die Erfüllungsgrade der einzelnen Anforderungen können in konkreten Architekturausprägungen stark variieren. Als Grundlage der Diskussion wird das generische Schichtenmodell (vgl. Kap. 3.2.2) mit möglichen Extensionen nach Sinz & Ulbrich-vom Ende (2010) aufgegriffen.

Die Unterstützung des verhaltensorientierten pDWH-Szenarios durch Architekturkonzepte klassischer DWH-Systeme ist in der Regel nicht gegeben (Vogt et al. 2008, Eckerson 2007). Als nicht berücksichtigte Anforderung ist die Integration des operativen und des dispositiven Teilsystems zu nennen (Eckerson 2007): Zum einen sind die Bereitstellung entscheidungsrelevanter Daten sowie die Realisierung von Reaktionen mit für das verhaltensorientierte Szenario angemessener Latenz im Systementwurf nicht berücksichtigt (Anforderung A1). Zum anderen stellt der bidirektionale Datentransfer (Anforderung A4) zur Rückkopplung abgeleiteter Reaktionen auf laufende Prozessinstanzen eine nicht berücksichtigte Anforderung dar. Die flexible Anpassbarkeit stellt grundsätzlich auch eine Anforderung an themenorientierte DWH-Systeme dar. Der erforderliche Grad der Flexibilität hängt vom jeweiligen Anwendungsfall ab. Im Falle von pDWH-Systemen sind Anpassungen des Geschäftsprozesses als Datenquelle und Ziel von Reaktion sowie sich stetig verändernde Nutzeranforderungen mögliche Einflussfaktoren. Mit Bezug zu den Architekturausprägungen nach Sinz & Ulbrich-vom Ende (2010) beeinflussen v.a. die Ausgestaltung der Schnittstellen und die gegenseitigen Abhängigkeiten zwischen Schichten die Anpassbarkeit sowie den Aufwand zur Realisierung neuer Anforderungen. So können Anpassungen einer Schicht, beispielsweise der Datenbereitstellungsschicht, sowohl niedrigere Schichten zur Befüllung des multidimensionalen Datenbestands als auch darüber liegenden Schichten betreffen. Die bedarfsgerechte Informationsversorgung verschiedener Nutzergruppen (Anforderung A2) findet über das Konzept der Data Marts in etablierten Architekturkonzepten hinreichend Berücksichtigung. Hierzu finden sich ebenfalls bei Sinz & Ulbrich-vom Ende (2010) entsprechende Vorschläge. Zur Integration prozess- und themenorientierter DWH-Systeme (Anforderung A5) bietet die ebenfalls dort vorgestellte *hierarchische Data-Warehouse-Architektur* einen möglichen Ansatz, um die hierarchische Beziehung zwischen prozess- und themenorientierten Systemen, wie in Kap. 3.5.1 vorgestellt, abzubilden.

Zusammenfassend begründen im Wesentlichen die nicht bzw. unvollständig berücksichtigten Anforderungen A1, A3 und A4 die Konzeption einer Referenzarchitektur für pDWH-Systeme. Etablierte Lösungsansätze themenorientierter DWH-Systeme, beispielsweise zur Realisierung der Anforderungen A2 und A5, sollen in die nachfolgende Konzeption integriert werden.

5.2.2 Diskussion verwandter Arbeiten zu pDWH-Architekturen

Im Weiteren werden zudem Forschungsarbeiten betrachtet, die sich mit Architekturen für pDWH-Systeme beschäftigen. Diese werden mit Bezug zu den in Kap. 5.1 erhobenen Anforderungen vorgestellt. Als Kriterium müssen verwandte Arbeiten auf wenigstens ein pDWH-Szenario aus Kap. 3.5.2 (verhaltens- oder strukturorientiert) ausgerichtet sein. In Tab. 5.1 sind die Ansätze anhand der unterstützten Anforderungen und Szenarien zusammengefasst.

		Anforderung					Szenario	
		A1: Latenz	A2: Nutzergruppen	A3: Flexibilität	A4: Datentransfer	A5: DWH-Integration	verhaltensorientiert	strukturorientiert
Technische Ansätze	Schaffner et al. (2009)	X					X	
	Kemper & Neumann (2011)	X					X	
	SAP HANA	X					X	
Erweiterungen etablierter Ansätze	Linden et al. (2010)	X	X		X		X	
	Christ (2009)	X					X	
	Schiefer et al. (2004)	X						X
	Casati et al. (2007)			X				X
	zur Muehlen (2001)		X					X
	Becker (2007)	X	X		X		(X)	X
service-orientierte Ansätze	Bucher et al. (2009)	X			X		X	
	Chan et al. (2011)		(X)	X				
	Dinter (2008)	X		X	X		(X)	
	Gordon et al. (2006)	X		X		X		
	Markarian et al. (2007)	X	X			X	X	
	Nguyen et al. (2005)	X			X		X	
	Oberhofer & Nijkamp (2009)	X	(X)				X	
	Pospiech & Felden (2013)	X		X	X		X	
	Vogt et al. (2008)	X	X	(X)	X		X	
Wu et al. (2007)	X			(X)				

X: explizit, (X): eingeschränkt berücksichtigte Anforderung/Szenario

Tab. 5.1: Berücksichtigte Anforderungen und pDWH-Szenarien in verwandten Arbeiten

Grundsätzlich können anhand des gewählten Lösungsansatzes drei Gruppen gebildet werden:

Technische Ansätze Eine erste Gruppe von Ansätzen beschreibt technische Konzepte (z.B. In-Memory-Technologien, spaltenorientierte Datenmodelle) für Datenbanksysteme zur Unterstützung hybrider operativer und analytischer Anwendungsfälle. Arbeiten dieser Gruppe sind im Wesentlichen auf die Reduktion der Latenz (A1) sowie die Vermeidung von Blockierungen operativer Datenbankzugriffe durch langlaufende Analysen bzw. Datenextraktion ausgerichtet. Die Problemlösung ist auf technischer Abstraktionsebene verortet, während in der vorliegenden Arbeit die konzeptuelle Architektur den Untersuchungsgegenstand bildet. Die Gruppe technischer Ansätze wird daher lediglich exemplarisch betrachtet. Stellvertretend können für die genannten Zielsetzungen die Arbeiten von Schaffner et al. (2009) und Kemper & Neumann (2011) angeführt werden. Die Arbeiten sind auf die Unterstützung des verhaltensorientierten Szenarios ausgerichtet. In der Praxis stellt das Datenbanksystem *SAP HANA* einen verbreiteten Vertreter dieser Kategorie dar. Die spaltenorientierte Hauptspeicherdatenbank bildet mit der umgebenden *SAP HANA Plattform* die erweiterbare Grundlage zur Integration operativer und analytischer Szenarien. Die Realisierung von Einblicken in laufende operative Prozesse mit geringer Latenz (A1) ist dabei die übergeordnete Zielsetzung (Färber et al. 2012, Word 2014).

Erweiterungen etablierter Architekturen Eine nächste Gruppe von Ansätzen erweitert etablierte Architekturkonzepte themenorientierter DWH-Systeme hinsichtlich der Unterstützung des verhaltens- oder des strukturorientierten pDWH-Szenarios: Bei Linden et al. (2010) wird die Erweiterung für den spezifischen Anwendungsfall des verhaltensorientierten Szenarios, der ereignisbasierten Steuerung von Logistikprozessen, vorgestellt. Gerade die pro-aktive Reaktion mit angemessener Latenz (A1) steht im Fokus. Ereignisse aus Logistikprozessen können gemeinsam mit multidimensionalen Daten des Kern-DWH-Systems ausgewertet werden. Ergebnisse können über den *Enterprise Service Bus* an die operativen Prozesse zurückgemeldet werden (A4). Über das Konzept der Data Marts stehen die Informationen allen Unternehmensebenen (strategisch, taktisch, operativ) entsprechend aufbereitet zur Verfügung (A2). Bei Christ (2009) wird BI-Funktionalität des verhaltensorientierten Szenarios in einer Schicht zwischen ope-

rativem AwS und DWH-System realisiert. Der Lösungsansatz basiert auf einer Erweiterung des Konzeptes des *Operational Data Store (ODS)* nach Inmon (1999). Als Anforderung wird ebenfalls explizit die Verkürzung der Latenz (**A1**) genannt. Daten werden auf Granularität der Geschäftstransaktion jedoch nur volatil mit kurzfristiger Historisierung vorgehalten. Bei Schiefer et al. (2004) wird die *Process Information Factory*, ein Architekturkonzept zur Verwaltung von Prozessdaten, vorgestellt. Eine klassische DWH-Architektur wird v.a. durch das Datenschema (vgl. Kap. 6) explizit um eine Prozessperspektive erweitert. Zur Realisierung des strukturorientierten Szenarios werden die Prozessdaten nicht-volatil, historisiert und multidimensional aufbereitet zur Verfügung gestellt. Auf die übrigen Anforderungen des strukturorientierten Szenarios, v.a. die Flexibilität zur Anpassung an sich verändernde Anforderungen, wird nicht explizit eingegangen. Zur Realisierung niedriger Latenz zur Datenbereitstellung wird zudem ein dem DWH-System vorgelagerter *Process Data Store (ODS mit Prozessdaten)* berücksichtigt. Die Daten liegen dabei jedoch nur volatil und nicht historisiert vor. Casati et al. (2007) schlagen ein generisches Konzept für ein pDWH-System, im Wesentlichen ausgerichtet auf das strukturorientierte pDWH-Szenario (Gestaltung von Geschäftsprozess-Spezifikationen), vor. Die Architektur ist auf die dynamische Anpassung des dispositiven Teilsystems an Veränderungen operativer Prozesse (**A2**) ausgerichtet. Bei zur Muehlen (2001) wird ein pDWH-System als DWH-Anwendung interpretiert. Dementsprechend finden sich Unterschiede des knappen Architekturentwurfs im Vergleich zu klassischen Architekturkonzepten vielmehr in der Ausgestaltung der Bestandteile (z.B. aufgezeichnete Ablaufinformationen von Workflows als Datenquelle, spezifisches Datenschema des DWHs) als in deren Art und Interaktion. Die Arbeit ist auf das *strukturorientierte* Szenario ausgerichtet. Zur Realisierung der Anforderungen unterschiedlicher Nutzergruppen (**A2**) sowie aus Performance-Gesichtspunkten wird die Nutzung von Data Marts motiviert. Ein umfassenderer Architekturansatz, der ebenfalls pDWH-Systeme als Anwendung etablierter Konzepte sieht, findet sich bei Becker (2007, S. 215ff.). Bestandteile eines etablierten Architekturkonzeptes werden als *Kernkomponenten*, die sich durch die Prozessorientierung ergebenden Bestandteile als *Zusatzkomponenten* bezeichnet. Grundlegende Architekturüberlegungen sind nicht Gegenstand der Betrachtung. Sowohl das *struktur-* als auch in Teilen das *verhaltensorientierte* Szenario werden explizit unterstützt, wenngleich die automatisierte Entscheidungsfindung sowie die Steuerung von Prozessen nicht thematisiert wer-

den. Als mögliche Reaktionen bzgl. laufender Prozessinstanzen werden das Auslösen von Ereignissen oder der Mail-Versand an verantwortliche Stellen genannt (A4) (Becker 2007, S. 223). Zur Realisierung niedriger Latenzen wird ebenfalls ein ODS eingesetzt (A1) (Becker 2007, S. 219). Zur Datenbereitstellung für verschiedene Nutzergruppen (A2) werden zusätzliche Präsentationswerkzeuge zur Visualisierung von Ergebnissen anhand von Prozessmodellen vorgeschlagen (Becker 2007, S. 220f.). Die Anforderung der Integration mit unternehmensweiten DWH-Systemen (A5) wird zwar erhoben (Becker 2007, S. 214), Lösungsansätze dazu werden jedoch nur auf Ebene des Datenschemas, nicht jedoch im Kontext der Architektur diskutiert.

Serviceorientierte Ansätze Die dritte Gruppe von Ansätzen setzt auf dem Paradigma Serviceorientierter Architekturen (SOA) auf. Bei Markarian et al. (2007) wird eine SOA-Referenzarchitektur für *Pervasive BI* vorgestellt. Darunter wird die Versorgung verschiedener Nutzergruppen (A2) auf strategischer, taktischer und operativer Ebene mit jeweils benötigten Daten unter Berücksichtigung der jeweiligen Latenzanforderungen (A1) verstanden. Das unternehmensweite DWH-System wird durch die Referenzarchitektur mit unterstützt (A5). Nguyen et al. (2005) beschreiben ein serviceorientiertes Konzept für das verhaltensorientierte Szenario. Zielsetzung ist die automatisierte Erkennung, Analyse und Entscheidungsfindung (A4) in laufenden Geschäftsprozessen zur Reduktion der zugehörigen Latenz (A1). Bei Oberhofer & Nijkamp (2009) wird die zeitnahe Versorgung operativer Mitarbeiter (A1 u. A2) mit entscheidungsrelevanten Informationen im verhaltensorientierten DWH-Szenario thematisiert. Auf die Heterogenität der Nutzer wird nicht explizit eingegangen. Von Pospiech & Felden (2013) wird eine serviceorientierte Referenzarchitektur für das verhaltensorientierte DWH-Szenario vorgestellt. Diese ist auf der Grundlage der Literatur im Bereich serviceorientierter BI (z.B. Vogt et al. 2008, Chan et al. 2011, Dinter 2008) konzipiert. Explizit bei der Gestaltung der Referenzarchitektur angesprochene Anforderungen sind neben der Reduktion der Latenz zwischen Geschäftsvorfall und Reaktion (A1) die flexible Anpassbarkeit (A3) sowie die Einbettung dispositiver Informationen in die Durchführung eines Geschäftsprozesses (A4). Bei Vogt et al. (2008) wird die Realisierung eines operativen DWH-Systems auf Basis einer serviceorientierten und ereignisgetriebenen Architektur vorgeschlagen. Transaktionale und analytische Dienste werden dabei über einen Enterprise Service Bus zur

Unterstützung des verhaltensorientierten pDWH-Szenarios flexibel koordiniert. Die Senkung der Latenz der Datenbereitstellung, Analyse und Reaktion (A1), die Unterstützung verschiedener Nutzergruppen (A2) sowie die enge Integration von operativem und analytischem System (A4) stellen die berücksichtigten Anforderungen dar. Die Flexibilität (A3) wird lediglich für operative Prozesse thematisiert. Bei Dinter (2008) sowie Bucher et al. (2009) wird keine vollständige Architektur spezifiziert, jedoch werden *Services* identifiziert, um die Potenziale serviceorientierter Architekturen in BI-Anwendungen auszuschöpfen. Als Ansatzpunkte werden dabei die Reduktion der Latenz (A1), die Flexibilität der Reaktion auf Veränderungen (A3) sowie die engere Integration operativer und analytischer Systeme (A4) thematisiert. Die Ausrichtung auf das verhaltensorientierte Szenario ist implizit gegeben. Zudem finden sich weitere Arbeiten, die die generelle Anwendung serviceorientierter Architekturen in BI-Anwendungen zum Gegenstand haben, jedoch die beiden pDWH-Szenarien nicht oder nur sehr eingeschränkt betrachten. Aus diesem Grund findet sich hier ebenfalls nur eine exemplarische Betrachtung. Chan et al. (2011) stellen die Realisierung eines *Enterprise-DWH-Systems*, basierend auf SOA-Konzepten, vor. Berücksichtigung finden dabei die flexible Realisierung veränderter Anforderungen (A3) sowie eine Zugriffsschicht mit heterogenen Formaten und Technologien (bedingt A2). Das *SOA-BI-Framework* nach Gordon et al. (2006) berücksichtigt *operational Reporting* in einer unternehmensweiten BI-Lösung. Das *Reporting* selbst wird dabei mit niedriger Latenz (A1) direkt auf operativen Datenquellen und AWS realisiert. Das DWH-System ist auf Nutzer aus dem Management ausgerichtet. Die Integration von DWH-System und operativem Reporting-System (A5) wird ebenso wie die Flexibilität (A3) betrachtet. Wu et al. (2007) schlagen eine Architektur zur Realisierung niedrigerer Latenzen (A1) auf Basis von SOA vor. Dazu trägt auch die automatisierte Nutzung von OLAP-Diensten durch die operativen Prozesse bei (vgl. A4 zur engeren Integration).

Zusammenfassende Würdigung Die vorgestellten Ansätze decken, auch aufgrund der unterschiedlich abgegrenzten Problemstellungen, die Anforderungen nicht vollständig ab. Wie zudem in Tab. 5.1 gezeigt, sind die Ansätze i.d.R. auf das verhaltens- oder das strukturorientierte Szenario ausgerichtet. Lediglich die Arbeit von Becker (2007) zielt auf beide Szenarien ab, wenn auch die erhobenen Anforderungen nicht vollständig abgedeckt sind. Ansonsten sind überwiegend Anforderungen des verhaltensorientierten Szenarios in den verwandten Architek-

turkonzepten berücksichtigt. Die innovativere und effizientere Prozessgestaltung im strukturorientierten Szenario haben nur wenige Architekturkonzepte zum Gegenstand. Das strukturorientierte Szenario wird, wie im nachfolgenden Kapitel (Stand der Literatur in Abschn. 6.3) gezeigt, in vielen Arbeiten ausschließlich durch Bereitstellung eines entsprechenden Datenschemas erreicht und als DWH-Anwendung interpretiert. Die Fähigkeit, das operative und das dispositive System konsistent zueinander flexibel anzupassen, wird nicht betrachtet.

Bei Analyse der Anforderungen ist festzustellen, dass nahezu alle Ansätze explizit die Bereitstellung entscheidungsrelevanter Daten mit angemessener Latenz realisieren (A1). Als Schlagworte aus den Arbeiten sind hierzu *real-time*, *near real-time* oder *right-time* zu nennen. Die übrigen Anforderungen werden in unterschiedlichem Maße von den verwandten Ansätzen abgedeckt. Serviceorientierte Ansätze berücksichtigen als eine zentrale SOA-Eigenschaft die Flexibilität zur Reaktion auf veränderte Anforderungen (A3). Die lose Kopplung zwischen Diensten stellt in SOA-Ansätzen das Lösungskonzept zur Realisierung des bidirektionalen Datenstransfers und damit der Integration analytischer Dienste in operativen Prozessen dar (A4). Die Informationsversorgung verschiedener Nutzergruppen (A2) findet nur in wenigen Arbeiten Berücksichtigung. Die Integration von prozess- und themenorientierten unternehmensweiten DWH-Systemen wird lediglich unter der Annahme betrachtet, dass beide Systeme SOA-basiert realisiert werden.

Die Gruppe der *serviceorientierten Ansätze* beschreibt Architekturkonzepte für DWH-Systeme basierend auf dem Paradigma der losen Kopplung. Diese Arbeit mit Konzeption einer Komponentenarchitektur setzt, bedingt durch das verwandte Funktionsparadigma, bei dieser Klasse von Ansätzen an. Neben Art und Zusammenspiel der Dienste wird aber auch die Innensicht der Komponenten mit Struktur und Verhalten betrachtet. Die Schnittstellen von Komponenten können als *Services*, z.B. nach Pospiech & Felden (2013), interpretiert werden und fließen als solche in die Referenzarchitektur ein. *Technische Ansätze*, insbes. das Datenbanksystem *SAP HANA* oder verwandte Systeme (z.B. NoSQL-Datenbanksysteme), stellen potenzielle Plattformen zur Realisierung der Datenverwaltung von Komponenten dar.

5.3 Konzeption einer komponentenorientierten Referenzarchitektur für operativ-dispositive Anwendungssysteme

Die Konzeption der Komponentenarchitektur für prozessorientierte operativ-dispositive AwS erfolgt als Referenzarchitektur. Zunächst wird die begriffliche Einordnung als solche in Abschn. 5.3.1 vorgenommen. Die Konzeption der Referenzarchitektur selbst erfolgt im Anschluss in zwei Ausbaustufen: In einem ersten Schritt wird eine Referenzarchitektur für flexible DWH-Systeme vorgestellt (Abschn. 5.3.2). Die zweite Ausbaustufe stellt schließlich die Erweiterung hin zur Referenzarchitektur für komponentenorientierte operativ-dispositive AwS dar (Abschn. 5.3.3). Die Spezifikation erfolgt plattformunabhängig und ermöglicht damit die Anwendbarkeit basierend auf unterschiedlichen Technologien.

5.3.1 Klassifikation als Referenzarchitektur

Für das in dieser Arbeit vorgestellte Architekturkonzept wird der Begriff der *Referenzarchitektur* verwendet. Der Begriff der Referenzarchitektur wird, in Anlehnung an Bauer & Günzel (2013, S. 37f.), als *Referenzmodell für die Architektur von AwS* interpretiert. Ein Referenzmodell ist ein Modell, das zur Unterstützung der Konstruktion konkreter Modelle (sog. Anwendungsmodelle) entwickelt bzw. eingesetzt wird. Bei der Entwicklung des Anwendungsmodells werden Inhalt und/oder Gegenstand des Referenzmodells wiederverwendet (vom Brocke 2003, S. 34). Thomas (2006) hebt explizit den Nutzungsaspekt von Referenzmodellen hervor. Die Nutzung eines Modells zur Unterstützung der Konstruktion anderer Modelle ist demnach entscheidend für die Klassifikation als Referenzmodell (Thomas 2006). Im vorliegenden Fall werden Referenzmodelle für die Architektur von pDWH-Systemen sowie für operativ-dispositive AwS konzipiert. Diese sind, auch mit Bezug zur Zielsetzung der Arbeit, explizit zur Unterstützung bei der Konstruktion konkreter Systeme vorgesehen. Nach Thomas (2006) sollte die Deklaration eines Modells als Referenzmodell im Idealfall dann erfolgen, wenn mindestens ein Anwendungsfall angegeben werden kann. Dieser liegt durch die begleitende Fallstudie *Mobility Provision* vor. Sie dient als Grundlage zur Demonstration der Anwendbarkeit der Referenzarchitektur in beiden Ausbaustufen. Den Anwendungskontext bilden die in Kap. 3.5.2 vorgestellten pDWH-Szenarien.

Nach Esswein (2016, S. 53) stellen die *Allgemeingültigkeit* sowie der *Empfehlungscharakter* intensionale Merkmale von Referenzmodellen dar. Der Aspekt der *Allgemeingültigkeit* wird bei Thomas (2006) weiter präzisiert. Demnach ist ein Referenzmodell nicht allgemein, sondern lediglich für eine Klasse von Problemen gültig, die sich im vorliegenden Fall anhand der Anforderungen aus Abschn. 5.1 bestimmen. In der ersten Ausbaustufe wird die Architektur von pDWH-Systemen beschrieben. Die Referenzarchitektur der zweiten Ausbaustufe ist für komponentenorientierte operativ-dispositive AwS gültig. Zwar ist die Einschränkung auf das komponentenorientierte Paradigma getroffen, jedoch ist Raum zur Ausgestaltung und Realisierung der Komponenten sowie für technologische Entscheidungen im Anwendungsmodell. Auch hinsichtlich der fachlichen Anwendungsdomäne ist die Architektur nicht festgelegt.

Zur Spezifikation von Anwendungsmodellen auf Basis von Referenzmodellen werden in der Literatur Konstruktionstechniken vorgeschlagen. Für eine vertiefende Betrachtung wird beispielsweise auf Esswein (2016, S. 55f.) oder vom Brocke & Buddendick (2004) verwiesen. Die zentrale Technik zur Spezifikation von Anwendungsmodellen auf Basis der nachfolgend vorgestellten Referenzarchitekturen stellt die *Analogiekonstruktion* dar. Die Bildung des Anwendungsmodells erfolgt derart, dass es in Bezug zur Referenzarchitektur hinsichtlich bestimmter Merkmale als übereinstimmend wahrgenommen werden kann (vom Brocke & Buddendick 2004). Zudem finden sog. *Platzhalter* (z.B. Esswein 2016, S. 56) zur Adaption der Referenzarchitektur an den konkreten Sachverhalt Anwendung.

5.3.2 Konzeption der flexiblen Data-Mart-Architektur

Die Referenzarchitektur für pDWH-Systeme ist auf die flexible Spezifikation von Data Marts zur Prozessanalyse ausgerichtet. Beide Kernkonzepte bilden den im Weiteren für die Referenzarchitektur genutzten Bezeichner der *flexiblen Data-Mart-Architektur*. Der Begriff der *Flexibilität* bestimmt sich dabei durch zwei Teilaspekte: Zum einen ermöglicht die pDWH-Realisierung auf Basis der Referenzarchitektur flexible Anpassung an sich verändernde Anforderungen. Zum anderen wird die Datenbereitstellung flexibel, mit variierender und für die pDWH-Szenarien angemessener Latenz ermöglicht. Die Referenzarchitektur zur Spezifikation flexibler Data Marts (*Flexible Data-Mart-Architektur*) ist in Abb. 5.1 dargestellt. Die wesentlichen Bestandteile stellen *Data Marts*, die *Datenintegration*

sowie *erweiterte Datenquellen* dar. Zum Erreichen der Flexibilitätsziele erfolgt die Aufteilung der Datenbereitstellung anhand der in Kap. 1.3 eingeführten Markt-Metapher von *Angebot und Nachfrage* in eine *angebotsorientierte* sowie eine *nachfrageorientierte* Datenbereitstellung. Die Konzeption erfolgt als Extension der generischen Schichtenarchitektur aus Kap. 3.2.

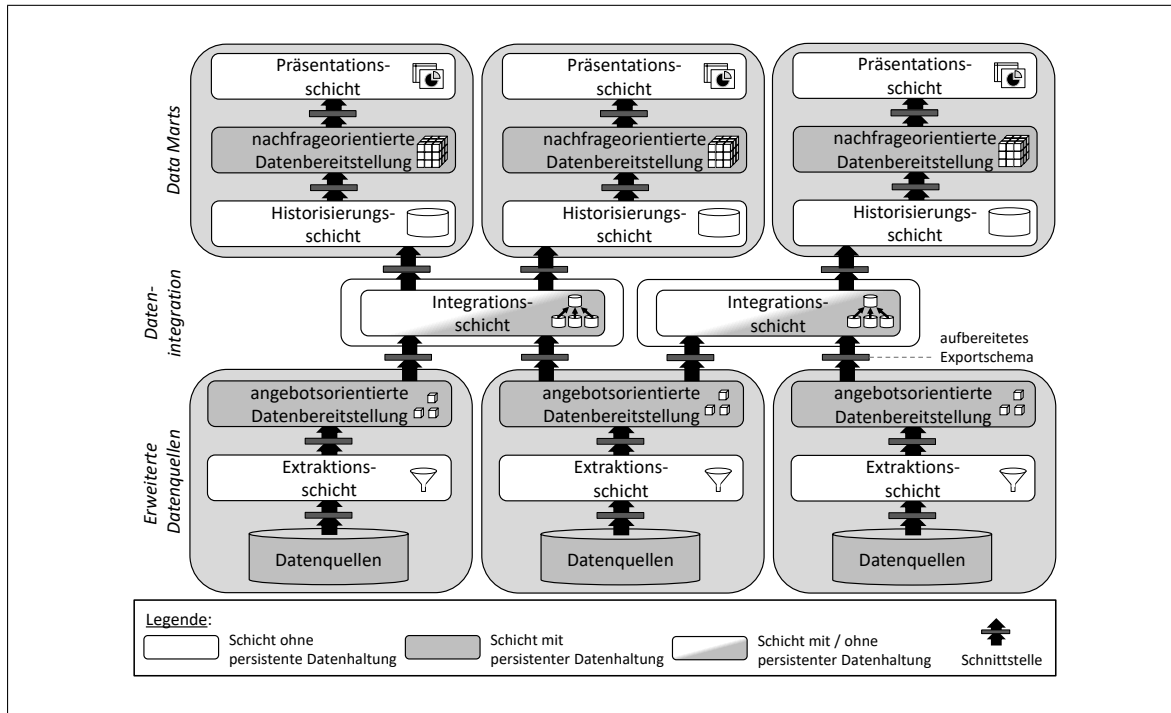


Abb. 5.1: Referenzarchitektur zur flexiblen Spezifikation von Data Marts (*Flexible Data-Mart-Architektur*)

Erweiterte Datenquellen Die Verantwortung der Extraktion von Daten (*Extraktionsschicht*) sowie der Bereitstellung dieser in aufbereiteter (nicht integrierter) Form wird den Datenquellen übertragen. Die Gestaltung des von den *erweiterten Datenquellen* bereitgestellten *aufbereiteten Exportschemas* erfolgt dabei angebotsorientiert. Zielsetzung ist es, möglichst umfassend aufbereitete entscheidungsrelevante Daten anzubieten, die flexibel zu heterogenen Analyseschemata integriert werden können. Struktur, Reichweite und Inhalt des aufbereiteten Exportschemas bestimmen sich aus der Quelle, unabhängig von der Nutzung im DWH-System bzw. den Data Marts. Als Beispiele für die Aufbereitung sind die Berechnung von Kennzahlen oder der Aufbau von Dimensionsstrukturen zu nennen. Die persistente Verwaltung der aufbereiteten Datenbestände erfolgt in der angebotsorientierten Datenbereitstellung, um flexibel und ohne Beeinflussung der Quellstrukturen Daten abgreifen zu können. Die Bereitstellung der Daten in generischer,

aufbereiteter Form trägt zur flexiblen Anpassung an neue bzw. veränderte Anforderungen bei. Durch inkrementelle Aktualisierung der aufbereiteten Daten nach dem *PUSH*-Prinzip, ausgelöst durch Veränderungen der operativen Instanzen, kann zur Realisierung kurzer Latenzzeiten beigetragen werden.

Data Marts Während erweiterte Datenquellen die Daten angebotsorientiert bereitstellen, erfolgt die Bereitstellung innerhalb der *Data Marts* nachfrageorientiert, in integrierter und multidimensionaler Form. Das Datenschema, die Granularität der Instanzen aber auch die Realisierungsform des multidimensionalen Datenmodells sowie die Präsentationswerkzeuge orientieren sich an den Anforderungen der Nutzer des *Data Marts*. Auch die Art der Historisierung hängt vom jeweiligen Analyse- und Berichtsbedarf der Nutzer ab (Hahne 2012). Als solches wird die Aufgabe der Historisierung, im Speziellen der Dimensionen, den *Data Marts* zugeordnet. Die persistente Datenhaltung in der Historisierungsschicht ist davon abhängig, ob diese als Grundlage mehrerer Schichten zur Datenbereitstellung genutzt wird (Sinz & Ulbrich-vom Ende 2010, S. 185f.). Im Rahmen der *flexiblen Data-Mart-Architektur* ist die Historisierung auf ein *Data Mart* ausgerichtet und damit nicht zwingend zu persistieren. Die *Historisierungsschicht* ist jedoch nicht zwingend auf das *Data Mart* festgelegt und kann auch der *angebotsorientierten Datenbereitstellung* vorgelagert werden. Als Teil des *Data Marts* wird die jeweilige Historisierungsstrategie anhand der Nutzeranforderungen festgelegt. Kennzahlen werden mit den für sie gültigen Dimensionsstrukturen an das *Data Mart* übergeben und nach Anwendung der jeweiligen Historisierungsstrategie in die Datenbasis eingefügt. Als Teil der angebotsorientierten Datenbereitstellung sind Dimensionen mit vollständiger Historie und Gültigkeit anzubieten. So wird die größtmögliche Flexibilität zur Verwendung der Datenstrukturen in nachgelagerten *Data Marts* gewährleistet (Hahne 2012).

Datenintegration Aufgabe der Datenintegration ist das Schließen der Lücke zwischen angebots- und nachfrageorientierter Datenbereitstellung. Nach dem Abgreifen der aufbereiteten Daten über die Exportschemata der verschiedenen Datenquellen werden diese nach erfolgter Integration an die relevanten *Data Marts* übergeben. Eine Systemkomponente zur Datenintegration kann dabei mehrere *Data Marts* mit vergleichbarer inhaltlicher Struktur beliefern. Die Datenintegration kann sowohl mit als auch ohne persistente Datenhaltung durchgeführt wer-

den. Der Verzicht auf eine persistente Datenhaltung kann sich positiv auf die flexible Anpassbarkeit und die flexible Datenbereitstellung auswirken. Anpassungen sind lediglich in der Integrationslogik durchzuführen. Die persistente Datenverwaltung von Integrations- und Transformationsschritten aus Gründen der Nachvollziehbarkeit und Rekonstruierbarkeit ist ebenfalls möglich.

Die *flexible Data-Mart-Architektur* wird anhand ausgewählter Aspekte nachfolgend weiter begründet und diskutiert:

Reichweite des DWH-Systems Die Anwendung der *flexiblen Data-Mart-Architektur* erscheint in klar abgrenzbaren, homogenen Domänen als geeignet. Homogene Strukturen gehen mit einem reduzierten Aufwand zur syntaktischen und semantischen Bereinigung der zu integrierenden Daten einher. Dies trägt zur flexiblen Datenbereitstellung mit niedriger Latenz bei. Bei der domänenübergreifenden Integration entscheidungsrelevanter Daten ist von einem höheren Aufwand zur Integration (z.B. Bereinigung heterogen codierter Begriffe) und damit einer höheren Latenz auszugehen.

Organisationsstruktur Durch die *flexible Data-Mart-Architektur* können Anforderungen verschiedener Nutzergruppen in Data Marts erfüllt werden. Dies kann sowohl auf fachlicher Ebene erfolgen aber auch durch geeignete Technologieauswahl unterstützt werden. Auf fachlicher Ebene wird das Analyseschema auf die Erfüllung des Informationsbedarfs ausgerichtet. Auf technischer Ebene kann die Realisierung nicht-funktionaler Anforderungen unterstützt werden. Als Beispiele sind die Realisierungsform des multidimensionalen Datenmodells, das Datenbankverwaltungssystem sowie die Zugriffstechnologie zur Realisierung, angepasst auf Datenvolumen und Latenzanforderungen, zu nennen. Das Konzept der *Polyglot Persistence* nach Sadalage & Fowler (2012, S. 133ff.) kann auf DWH-Systeme übertragen werden. Dieses sieht vor, in Abhängigkeit von technischen und strukturellen Anforderungen einer Anwendungsdomäne das passende Datenbanksystem zu wählen. Als Vorteile sind dabei u.a. die Reduktion der Komplexität während der Entwicklung der Persistenzfunktionalität und der Datennutzung zur Laufzeit sowie die technische Ausrichtung des Datenbanksystems auf die Anwendungsdomäne zu nennen (Sadalage & Fowler 2012, S. 133ff.). Die *flexible Data-Mart-Architektur* ist für Szenarien geeignet, in denen verschiedene

Nutzergruppen in einem gemeinsamen Entscheidungskontext nicht nur hinsichtlich des nachfrageorientierten Datenschemas, sondern auch bzgl. technischer und nicht-funktionaler Aspekte variierende Anforderungen haben.

Flexibilität der Gestaltung Durch die Bildung eigenständiger Teilsysteme mit klar abgegrenzten Aufgaben (*Seperation of Concerns*) können sich verändernde oder neue Anforderungen flexibel realisiert werden. Je nach Änderung ist die Anpassung bzw. Erstellung einer oder mehrerer Systemkomponenten (erweiterte Datenquelle, Datenintegration, Data Mart) notwendig. Durch parallele Anpassung mehrerer Systembestandteile sowie die Spezifikation stabiler Schnittstellen kann die Entwicklungszeit zudem reduziert werden.

Flexibilität der Datenbereitstellung Flexibilität besteht ebenfalls hinsichtlich der Latenz zur Datenbereitstellung in den Data Marts. Erweiterte Datenquellen können Änderungen operativer Daten unmittelbar in aufbereiteter Form an die Datenintegration weitergeben. Entsprechend der heterogenen zeitlichen Anforderungen erfolgen die Datenintegration sowie die nachfrageorientierte Bereitstellung. Innerhalb der Data Marts kann zudem durch geeignete Technologieauswahl die Latenz beeinflusst werden.

Datenquellen Die Erweiterung von Datenquellen zur angebotsorientierten Bereitstellung des aufbereiteten Exportschemas gestaltet sich bei externen Quellen durch mangelnden Zugriff auf die Implementierung problematisch. Durch Kapselung können diese um Extraktions- und Bereitstellungsfunktionalität erweitert werden. Die Realisierung der inkrementellen Extraktion von Daten nach dem *PUSH*-Prinzip erfordert jedoch Zugriff auf die Implementierung. Komplexität und Aufwand zur angebotsorientierten Bereitstellung entscheidungsrelevanter Daten in erweiterten Datenquellen sind von der Komplexität und der Größe des verwalteten Datenschemas abhängig. Klar abgrenzbare Datenstrukturen mit handhabbarer Reichweite erscheinen als geeignet.

Die *flexible Data-Mart-Architektur* ist, zusammengefasst, für Szenarien geeignet, in denen heterogene und über die Zeit variierende Nutzeranforderungen bzgl. Aufbereitung des zu analysierenden Datenschemas und der Latenzen zur Datenbereitstellung bestehen. Eine klar abgegrenzte und homogene Anwendungsdomäne ist vorteilhaft.

5.3.3 Spezifikation von Komponenten für operativ-dispositive Systeme

Als Grundlage für die Referenzarchitektur operativ-dispositiver AwS dient die in Abschn. 5.3.2 vorgestellte *flexible Data-Mart-Architektur*. Zielsetzung der zweiten Ausbaustufe der Referenzarchitektur ist die Unterstützung der pDWH-Szenarien (Kap. 3.5.2). Deshalb wird neben der Interpretation der *flexiblen Data-Mart-Architektur* als Komponentenarchitektur auch die Integration mit Komponenten des EU-Systems und des operativen Systems betrachtet.

Wie in Kap. 2.3 beschrieben, interagieren Komponenten basierend auf dem Konzept der losen Kopplung. Aufgrund dessen können komponentenorientierte AwS flexibel an sich verändernde Anforderungen angepasst und um neue Funktionalität erweitert werden (Sharp & Ryan 2005). Gerade zur Realisierung von angemessenen Latenzen zur Datenbereitstellung (Anforderung **A1**), der flexiblen Anpassbarkeit (Anforderung **A3**) sowie der engeren Integration von operativem und dispositivem Teilsystem (Anforderung **A4**) erscheint das komponentenorientierte Paradigma geeignet. Bei Panian (2008) wird ebenfalls das komponentenorientierte, im Speziellen das serviceorientierte Paradigma, für DWH-Systeme motiviert, wenn es darum geht, Analysen mit Fokus auf dem zugrundeliegenden Prozess durchzuführen und flexibel auf veränderte Anforderungen zu reagieren. Die Bestandteile *Erweiterte Datenquellen*, *Datenintegration* und *Data Marts* der *flexiblen Data-Mart-Architektur* erfüllen die Eigenschaften von Software-Komponenten und werden als Komponententypen verstanden (vgl. Kap. 2.3). Jeder Komponententyp stellt über eine wohldefinierte Schnittstelle Dienste zur Durchführung einer definierten Aufgabe bereit. Über die Schnittstellen werden das *erweiterte Exportschema*, das *integrierte Datenschema* bzw. das *Analyseschema* (nachfrageorientierte Datenschema) mit zugehörigen Operatoren angeboten. Die Komponententypen sind bzgl. ihrer Funktionalität abgeschlossen und können als Einheit wiederverwendet werden. Das resultierende Konzept der Komponentenarchitektur ist als abstraktes Beispiel in Abb. 5.2 dargestellt. Die Spezifikation der Komponenten erfolgt generisch auf Typ-Ebene. Die Instanziierung erfolgt durch Parametrisierung. Parameter können als *Platzhalter* (vgl. Abschn. 5.3.1) zur Anwendung von Referenzmodellen verstanden werden. Schnittstellen werden soweit möglich generisch spezifiziert. Die im Folgenden angegebenen Schnittstellen sind auf funk-

tionale Bestandteile beschränkt. Vor- und Nachbedingungen sowie qualitative Eigenschaften (z.B. Datensicherheit und Performance, vgl. Kap. 2.3.1) ergeben sich z.T. erst aus dem konkreten Anwendungsfall und sind für die vorliegende Problemstellung zweitrangig.

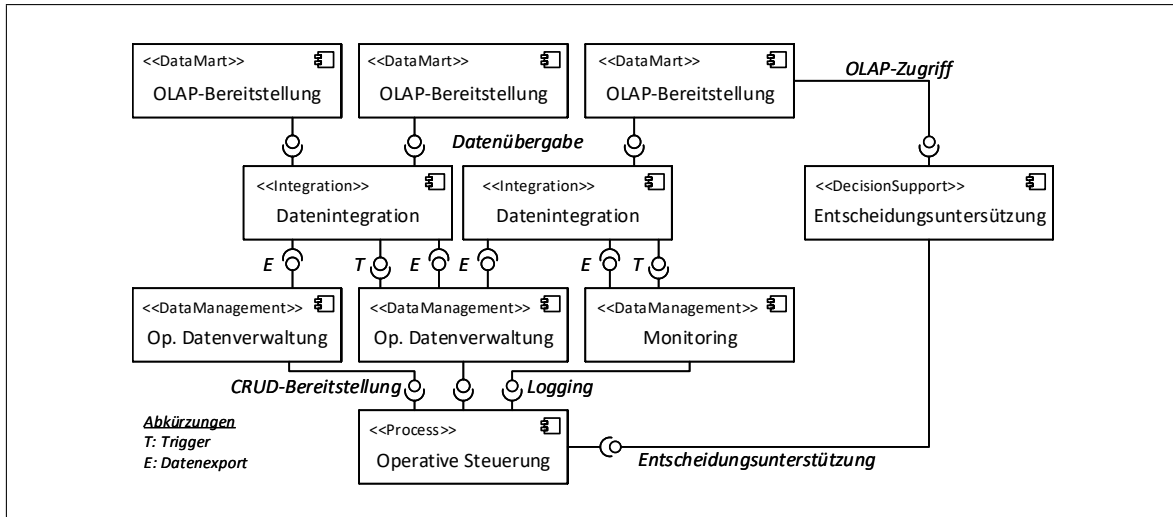


Abb. 5.2: Referenzarchitektur operativ-dispositiver Anwendungssysteme

Erweiterte Datenquellen werden um Verwaltungsfunktionen operativer Daten erweitert und zu Komponenten zur *Operativen Datenverwaltung* (<<DataManagement>>, Datenkomponenten). Über ihre Schnittstelle zur *CRUD-Bereitstellung* werden operative Funktionen bzgl. der verwalteten Datenstrukturen (i.d.R. CRUD-Operationen; *Create, Read, Update, Delete*) angeboten. Komponenten zur *operativen Vorgangssteuerung* (<<Process>>, Vorgangskomponenten) orchestrieren *Datenkomponenten* zur Realisierung höherer Geschäftsfunktionalität. Komponenten zur Verwaltung von Laufzeitinformationen (*Monitoring-Komponenten*) stellen einen Spezialfall der operativen Datenkomponente dar. Relevante Ablaufinformationen können über die *Logging*-Schnittstelle an die *Monitoring-Komponente* gegeben werden. Über die *Export*-Schnittstelle wird das *aufbereitete Exportschema* angeboten. Die Bereitstellung erfolgt, wie in der *flexiblen Data-Mart-Architektur* beschrieben, multidimensional aufbereitet, jedoch nicht integriert. Kennzahlen und Dimensionen werden auf Granularität einer Prozessinstanz zur Verfügung gestellt. Für den Zugriff auf das Exportschema wird die nachfolgende generische Schnittstelle definiert:

```
getOlapData(OlapDataType, OlapDataName, TID)
```

Über den Parameter *OlapDataType* wird der Zugriff auf Kennzahl oder Dimension festgelegt. *OlapDataName* identifiziert das OLAP-Datenobjekt über seinen Be-

zeichner. TID legt die Prozessinstanz (*Transaktions-ID*) fest, für die die Daten abzurufen sind.

Komponenten zur *OLAP-Bereitstellung* (<<DataMart>>) bilden sich aus den *Data Marts* der *flexiblen Data-Mart-Architektur*. Sie übernehmen die Aufgabe der nachfrageorientierten Datenbereitstellung und werden im Wesentlichen durch das multidimensionale Schema parametrisiert. Über die Schnittstelle zum *OLAP-Zugriff* werden das multidimensionale Analyseschema sowie die OLAP-Operationen (vgl. Kap. 3.3.2) angeboten. Nutzer der Schnittstelle sind zum einen Präsentationswerkzeuge zur interaktiven Datenanalyse durch Endanwender. Zum anderen kann durch das EU-System die Analyse von Daten auch (teil-)automatisiert erfolgen. Entscheidungsunterstützende Komponenten (<<DecisionSupport>>) bieten fachliche Entscheidungsfunktionen über die gleichnamige Schnittstelle an. Grundsätzlich können auch seitens der Vorgangskomponenten Schnittstellen zur Steuerung angeboten und von EU-Komponenten genutzt werden (in Abb. 5.2 nicht dargestellt).

Komponenten zur *Datenintegration* (<<Integration>>) übernehmen die Aufgabe der Integrationsschicht der *flexiblen Data-Mart-Architektur*. Sie nutzen die *Export-Schnittstelle* der Datenkomponenten zum Zugriff auf aufbereitete Datensätze. Durch die unabhängige, angebotsorientierte Datenbereitstellung kann die Datenintegration flexibel an neue Anforderungen angepasst werden. Nach Durchführung der instanzspezifischen Datenintegration werden die entscheidungsrelevanten Daten (*OlapData*) über die Schnittstelle zur *Datenübergabe* an die Data-Mart-Komponenten übergeben. Die Erweiterung des Datenbestandes von Data Marts erfolgt inkrementell. Die Schnittstelle zur Datenübergabe wird dazu wie folgt spezifiziert:

$$\text{pushOlapData}(\text{OlapData}, \text{TID})$$

Auslöser der Datenintegration sind Ereignisse in operativen oder in Data-Mart-Komponenten. In Datenkomponenten können hierzu fachliche (z.B. Abbruch einer Bestellung) und zeitbezogene Ereignisse (z.B. feste Zeitpunkte oder Zeitspannen), in Monitoring-Komponenten technische Ereignisse (z.B. Ausnahmeverhalten in oder Abbrüche von Prozessinstanzen) und in Data-Mart-Komponenten ad-hoc Ereignisse (ad-hoc Nutzeranfragen) definiert werden. Ebenso sind Ereignisse

in einer EU-Komponente (z.B. bestimmte Analyseergebnisse) oder im Ablauf von Vorgangskomponenten denkbar. Ereignisse bieten auf konzeptueller Ebene die Möglichkeit, den richtigen Zeitpunkt zur Datenbereitstellung zu definieren. Das Auslösen der Datenintegration erfolgt über die nachfolgend spezifizierte *Trigger-Schnittstelle*:

```
trigger(ProcessState, TID)
```

Neben der Prozessinstanz (TID) stellt der Prozesszustand (ProcessState) einen wichtigen Input dar. Anhand des Zustands ist die Verfügbarkeit der multidimensionalen Teilstrukturen ableitbar. Als Beispiel ist die Datenintegration aufgrund einer verzögerten Lieferung zu nennen. Datensätze bzgl. Abrechnung oder Zahlung sind zum Zeitpunkt der Datenintegration noch nicht verfügbar und somit nicht Teil des Integrationsprozesses.

5.3.4 Anwendung der Referenzarchitektur auf die Fallstudie *Mobility Provision*

Die Anwendung der Referenzarchitektur auf die Fallstudie *Mobility Provision* ist in Abb. 5.3 dargestellt und erfolgt im Wesentlichen durch *Analogiekonstruktion* (vgl. Abschn. 5.3.1). Das Anwendungsmodell zur Fallstudie stimmt so mit Bezug zur Trennung in angebots- und nachfrageorientierte Datenbereitstellung mit der *flexiblen Data-Mart-Architektur*, in Bezug auf Komponententypen sowie deren Interaktion mit der *Referenzarchitektur operativ-dispositiver AwS* überein. Schnittstellen werden durch Parametrisierung der *Platzhalter* angepasst.

Zur Steuerung der Durchführung der Aufgaben von *Vertriebskanal* und *Auftragsverwaltung* ist jeweils eine Vorgangskomponente (<<Process>>) definiert. Auf weitere Strukturierung dieser Komponenten wird verzichtet. Zur Verwaltung von Stammdaten sind die Komponenten *Kundendaten*, *Vertriebsdaten*, *Mietmodelle* und *eCar-Daten* spezifiziert. Die Verwaltung von Transaktionsdaten erfolgt über die Komponenten *Buchungsdaten*, *Mietprozessdaten* und *Forderungsdaten*. Ablaufinformationen zu den Prozessen werden in der Monitoring-Komponente *Ablaufdaten* verwaltet. Diese Datenkomponenten werden über ihre *CRUD-Schnittstelle* zur Realisierung der Vertriebs- bzw. Auftragsverwaltungsfunktionalität orchestriert (Auftragsverwaltung: gestrichelte Linie, Vertrieb: einfache Linie als Schnittstellenbeziehung).

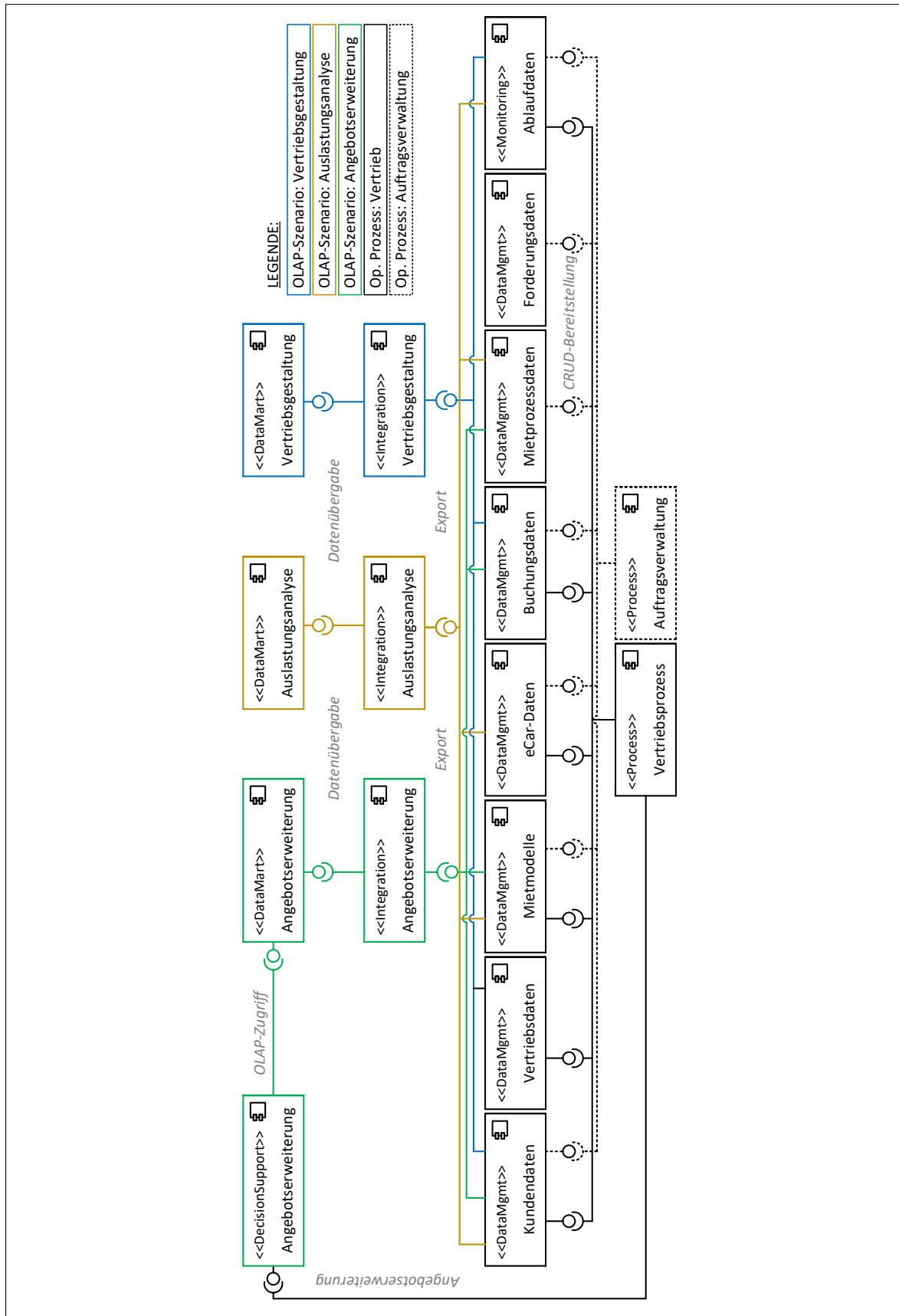


Abb. 5.3: Komponentenarchitektur zur Fallstudie *Mobility Provision*

Für die in Kap. 4.4 vorgestellten Entscheidungsszenarien sind im dispositiven

Teilsystem der Architektur jeweils ein Data Mart (<<DataMart>>) sowie eine zugehörige Komponente zur Datenintegration (<<Integration>>) spezifiziert. Nachfolgende Erläuterungen erfolgen auf Basis des Entscheidungsszenarios zur *Vertriebsgestaltung*: Für die Data-Mart-Komponente *Vertriebsgestaltung* werden OLAP-Datensätze der operativen Komponenten *Kundendaten*, *Vertriebsdaten*, *Mietmodelle*, *eCar-Daten* und *Buchungsdaten* benötigt. Diese werden über die Export-Schnittstellen abgegriffen. Exemplarisch werden hierzu die Schnittstellen der Komponenten *Vertriebsdaten* und *Buchungsdaten* beschrieben. Code 5.1 zeigt die Spezifikation der *Vertriebsdaten*-Komponente. Zulässige Parameter sind durch *Enumerations* (enum) angegeben. Es werden lediglich Dimensionsstrukturen (OlapDataType) bereitgestellt. Diese werden für Callcenter-Mitarbeiter (CCSTUFF) sowie für die Vertriebsart (SALESTYPE) gebildet.

```

1 % unterstützte Arten des OLAP-Datensatzes
2 public enum OlapDataType {DIMENSION}
3
4 % unterstützte Bezeichner des OLAP-Datensatzes
5 public enum OlapDataName {CCSTUFF, SALESTYPE}
6
7 % Methode der OLAP-Schnittstelle
8 public OlapData getOlapData(OlapDataType olapDataType,
9                             OlapDataName olapDataName,
10                            int tid)

```

Code 5.1: Export-Schnittstelle der Komponente *Vertriebsdaten*

In Code 5.2 ist die Schnittstellenspezifikation der Komponente *Buchungsdaten* dargestellt. Neben Dimensionen (Buchungszeit, BOOKINGTIME, Zeile 5) können auch Kennzahlen (FACT) abgerufen werden.

```

1 % unterstützte Arten des OLAP-Datensatzes
2 public enum OlapDataType {FACT, DIMENSION}
3
4 % unterstützte Bezeichner des OLAP-Datensatzes
5 public enum OlapDataName {BOOKINGTIME,
6                             NUM_OFFERS, NUM_BOOKINGS}
7
8 % Methode der OLAP-Schnittstelle
9 public OlapData getOlapData(OlapDataType olapDataType,
10                             OlapDataName olapDataName,
11                            int tid)

```

Code 5.2: Export-Schnittstelle der Komponente *Buchungsdaten*

Zeile 6 zeigt als Kennzahlen die Anzahl der getätigten Angebote in einer Transaktion (NUM_OFFERS) sowie die resultierende Anzahl der Buchungen (NUM_BOOKINGS).

Das Szenario *Vertriebsgestaltung* der Fallstudie ist strukturorientiert. Als solches wird die Latenz zur Datenbereitstellung als unkritisch eingestuft. Als Auslöser der Datenintegration wird der Abschluss einer Vertriebstransaktion festgelegt. Der Aufruf erfolgt mit dem Prozesszustand und der Transaktionsnummer (Code 5.3).

```
1 % unterstützte Prozesszustände
2 public enum ProcessState {SALES_FINISHED}
3
4 % Methode der Triggerschnittstelle
5 public void trigger(ProcessState processState, int tid)
```

Code 5.3: Schnittstelle der Datenintegration zum Szenario *Vertriebsgestaltung*

In analoger Form können Schnittstellen für das Szenario *Auslastungsgestaltung* angegeben werden. Unterschiede liegen v.a. in der Nutzung der operativen Komponenten. Grundlage des Szenarios bildet die Analyse der aktuell gebuchten Auslastung an den jeweiligen Leihstationen. Als Kennzahlen können die *Anzahl der Buchungen* oder daraus abgeleitet die *Buchungsquote* (Datenkomponente *Buchungsdaten*) in Bezug zu Dimensionen wie *Standortgeographie* (Datenkomponente *eCar-Daten*) oder *Buchungszeiträume* gesetzt werden. Letztere können beispielsweise Buchungen Urlaubs- oder Ferienzeiten zuordnen. Zur Analyse der Auslastung wird die Data-Mart-Komponente *Auslastungsanalyse* spezifiziert. Mögliche abgeleitete Reaktionen sind individuelle Angebote an Kunden, um das Fahrzeug an unterversorgten Stationen abzugeben, oder das Auslösen eines Serviceprozess zur Umverteilung der Flotte.

Die Komponenten zur *Angebotsenerweiterung* leiten sich aus dem Szenario *Angebotsenerweiterung* ab, das in Kap. 7.6.2 noch vertiefend eingeführt wird. Neben Data Mart und Datenintegration ist auch eine Komponente zur automatisierten Entscheidungsunterstützung vorgesehen (<<DecisionSupport>>). Die Entscheidung zur Angebotsenerweiterung erfolgt automatisiert. Die fachliche Schnittstelle der Entscheidungskomponente ist in Code 5.4 dargestellt. Aus dem automatisierten Ablauf des Vertriebsprozesses erfolgt der Aufruf mit Kundenparametern (Customer). Die Durchführung der Entscheidung, ob ein Kundenangebot erweitert wird, basiert auf einem multidimensionalen Datenbestand. Auf diesen wird über die Schnittstelle *OLAP-Zugriff* zugegriffen. Anhand von Kundendaten und abgeschlossenen Mietprozessen vergleichbarer Kunden wird die Entscheidung getroffen und an den Vertriebsprozess zurückgegeben. Die Ausgestaltung

der Schnittstellen der operativen Komponenten sowie der Datenintegration erfolgt in analoger Form zum Szenario *Vertriebsgestaltung*.

```
1 % Methode zur Angebotsindividualisierung
2 public Boolean decideOfferExtension(Customer customer)
```

Code 5.4: Schnittstelle der Entscheidungskomponente zur individuellen Angebotserweiterung

5.4 Konzeption der Komponenten des prozessorientierten DWH-Systems aus Innensicht

Nachfolgend wird die Funktionalität der in Abschn. 5.3.3 vorgestellten Komponententypen des pDWH-Systems aus Innensicht betrachtet. Die Struktur der Ausarbeitung orientiert sich am Prozess der Datenbereitstellung, startend mit der Verwaltung operativer Daten und der angebotsorientierten Datenbereitstellung in Datenkomponenten hin zur nachfrageorientierten Bereitstellung in den Data Marts.

5.4.1 Angebotsorientierte Datenbereitstellung durch operative Komponenten

Neben der Verwaltung operativer Daten zählen die Datenextraktion sowie die angebotsorientierte Datenbereitstellung zu den Aufgaben operativer Datenkomponenten. Ein Konzept zum Funktionsablauf von der Extraktion bis hin zu deren Bereitstellung in multidimensionaler, nicht integrierter Form ist in Abb. 5.4 dargestellt und Gegenstand der nachfolgenden Betrachtung. Der komponenteninterne Ablauf besteht aus den Phasen *Extraktion* und *angebotsorientierte Datenbereitstellung*.

Extraktion operativer Daten Im Rahmen der automatisierten Durchführung von Geschäftsprozessen übernimmt eine Datenkomponente die persistente Verwaltung operativer Daten, auf die von Vorgangskomponenten über *CRUD*-Operationen zugegriffen wird. Diese Nutzung operativer Daten bildet den Ausgangspunkt der Datenextraktion im pDWH-System, wie in Abb. 5.4 gezeigt. Relevant sind schreibende Zugriffe auf operative Daten. Werden diese erzeugt (*create*) oder aktualisiert (*update*), erfolgt auch die Aktualisierung des multidimensionalen Datenbestandes innerhalb der Datenkomponente. Prüfung der Zugriffsoperation,

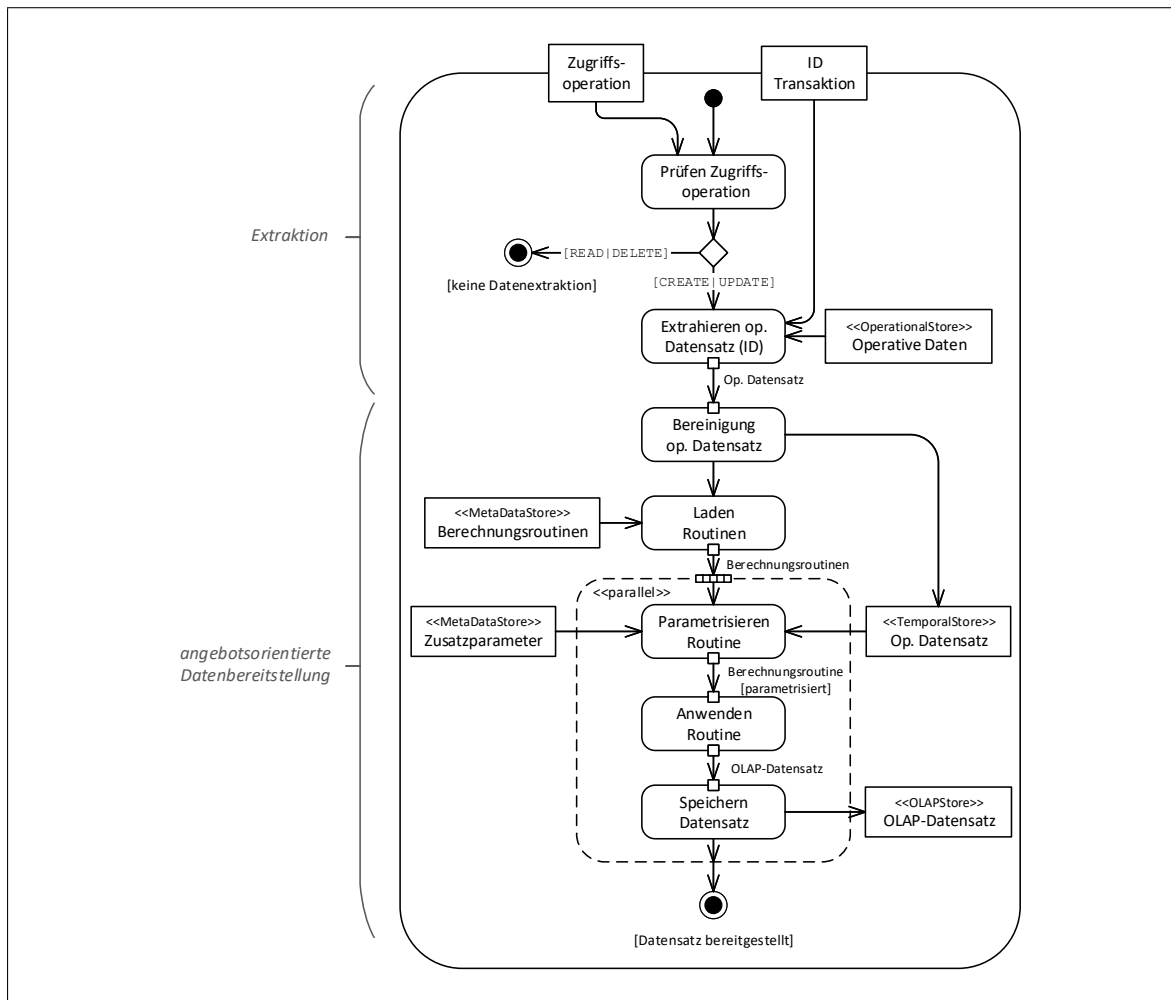


Abb. 5.4: Komponenteninterne, angebotsorientierte Datenbereitstellung

Extraktion und Bereitstellung der Daten erfolgt auf Ebene von Transaktionsinstanzen (TID) in einem separaten Prozess. Durch rein lesenden Zugriff (Aktivität: *Extrahieren des operativen Datensatzes*) auf den *operativen Datenbestand* (<<OperationalStore>>) wird der Einfluss auf die operative Datenverwaltung gering gehalten.

Angebotsorientierte Datenbereitstellung In die Phase der *angebotsorientierten Datenbereitstellung* fällt zunächst die Aufgabe der semantischen Datenbereinigung. In der generischen Schichtenarchitektur (Kap. 3.2.2) ist die semantische Bereinigung von operativen Daten der Integrationsschicht zugeordnet. Die Trennung in angebots- und nachfrageorientierte Datenbereitstellung der *flexiblen Data-Mart-Architektur* macht die Verlagerung in das angebotsorientierte Teilsystem notwendig. Als typische semantische Schwächen sind fehlende, fehlerhafte oder inkonsistente Werte innerhalb eines multidimensionalen Datensatzes zu nennen (Golfarelli & Rizzi 2009, S. 16f.). Auch zur Reduktion der Latenz müssen die

multidimensionalen Datensätze bei Zugriff der Integrationskomponente in bereinigtem, fehlerfreiem Zustand vorliegen. Nach der Bereinigung des operativen Datensatzes erfolgt die multidimensionale Aufbereitung. Hierzu werden ergänzende Metadaten (<<MetaDataStore>>) benötigt. Zunächst sind *Berechnungsroutinen* zur OLAP-Aufbereitung der operativen Daten notwendig. Nach dem Laden der Routinen werden diese einzeln, ggf. nebenläufig (<<parallel>>), parametrisiert mit dem operativen Datensatz angewendet. Zur Anwendung sind ggf. weitere *Zusatzparameter* notwendig. Ein Beispiel stellt der Aufbau einer Dimension *Kundengeographie* dar. Zwar ergibt sich der Wohnort eines Kunden aus dem operativen Datensatz, zur Zuordnung von Wohnort zu Landkreis oder zu Bundesland sind jedoch weitere Informationen notwendig. Nach erfolgreicher Berechnung eines OLAP-Datensatzes wird dieser, identifiziert über seine Transaktionsinstanz (TID), persistent gespeichert (<<OLAPStore>>). Zur Speicherung erscheint das *Key-Value*-Paradigma als geeignet.

5.4.2 Integration multidimensionaler Datenstrukturen

Auch die Ablauflogik einer Komponente zur Datenintegration kann, wie in Abb. 5.5 dargestellt, generisch beschrieben werden. In Kap. 3.2.2 wird neben der bereits thematisierten semantischen auch die syntaktische Bereinigung (auch Transformation genannt) in den Aufgabenbereich der Integrationsschicht eingeordnet. Zielsetzung ist es, Daten aus heterogenen Quellen in ein einheitliches Format zu transformieren, um u.a. deren Vergleichbarkeit zu ermöglichen (Bauer & Günzel 2013, S. 57). Die Reichweite eines pDWH-Systems wird durch einen Geschäftsprozessstyp begrenzt. Innerhalb dieser Grenzen wird von weitgehend homogenen Strukturen ausgegangen, so dass syntaktische Transformationen, wie das Angleichen von Datentypen oder Datumsangaben, nicht notwendig erscheinen. Die Phase der *Integration* ist somit auf das Zusammenführen relevanter OLAP-Datensätze fokussiert, die angebotsorientiert von operativen Datenkomponenten zur Verfügung gestellt werden. Voraussetzung ist das Vorliegen der Daten auf gleicher Granularität. Hier wird im Wesentlichen zwischen Kennzahlen bzw. Dimensionen unterschieden, die entweder auf Granularität des gesamten Prozesses (Prozessergebnis) oder einzelner Prozessschritte (Ablaufinformationen) vorliegen. Bei Auslösung des Prozesses der Datenintegration werden Prozesszustand und Transaktions-ID (vgl. Abschn. 5.3.3) als Parameter übergeben. Zur Integration wird zunächst das zu erstellende Ziel-Datenschema (<<MetaDataStore>> *Zielstrukturen*)

gelesen. Über den Prozesszustand wird die Verfügbarkeit bzw. die Notwendigkeit von Bestandteilen des zu erstellenden multidimensionalen Datensatzes geregelt. Der Zugriff auf die jeweiligen Bestandteile des multidimensionalen Datensatzes kann nebenläufig erfolgen. Über das Allokationsschema (<<MetaDataSet>> *Schemaallokation*) wird die Adressierung operativer Komponenten verwaltet, über die anhand der *TID* der jeweilige *OLAP-Datensatz* abgerufen wird. Nach Vorliegen aller multidimensionaler Teile werden diese zu einem *multidimensionalen Datensatz* integriert. Nach erfolgter Integration wird der multidimensionale Datensatz an die relevanten Data-Mart-Komponenten bzw. deren *multidimensionale Datenstruktur* übergeben. Adressen der Zielkomponenten werden hierzu aus den entsprechenden Metadaten (<<MetaDataSet>>) ausgelesen.

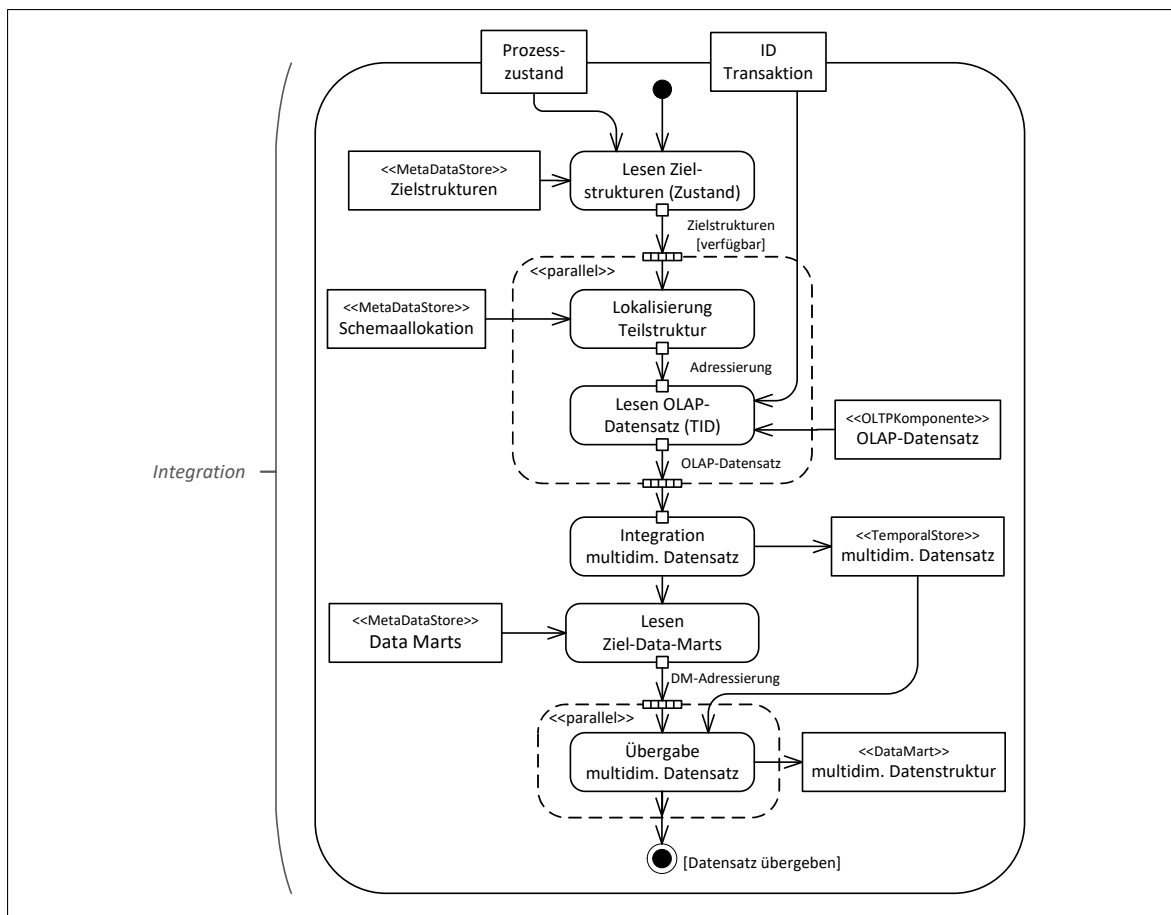


Abb. 5.5: Komponentenübergreifende Datenintegration

5.4.3 Nachfrageorientierte Datenbereitstellung in Data Marts

Data-Mart-Komponenten übernehmen einen integrierten multidimensionalen Datensatz und stellen diesen historisiert und nachfrageorientiert den zugeordneten Nutzern bereit. Die Phasen der *Historisierung* und der *Datenbereitstellung* werden nachfolgend allgemein erläutert.

Historisierung multidimensionaler Datensätze Zur Analyse von Kennzahlen im Zeitablauf werden diese mit einer oder mehrerer Zeitdimensionen in Beziehung gesetzt. Die Dimension Zeit wird im Integrationsprozess bereits zugeordnet und aus entsprechenden Datenkomponenten abgegriffen. Die Historisierung des Data Marts bezieht sich im Wesentlichen auf Dimensionsstrukturen. Auch für diese ergibt sich mitunter die Notwendigkeit, Veränderungen auf Instanzebene zu verwalten (Hahne 2012). Dadurch wird es ermöglicht, Kennzahlen im aktuellen Kontext, im Kontext ihrer Entstehung oder in einem vergangenen Kontext zu bewerten (Bauer & Günzel 2013, S. 233). Hierzu ist in der Phase der Historisierung die passende *Historisierungsstrategie* zu wählen und auf den multidimensionalen Datensatz anzuwenden. Die geeignete Strategie ist dabei vom Analyse- und Berichtsbedarf der jeweiligen Nutzer abhängig (Hahne 2012). Bei Kimball (2009, S. 389ff.) werden für das Star-Schema drei grundlegende Strategien unter dem Begriff *Slowly Changing Dimension* vorgestellt. Die Unterschiede liegen v.a. in der Dokumentation der Klassifikationshistorie. *Typ 1*-Dimensionen führen keine Historie und überschreiben bestehende Werte bei Veränderungen, während *Typ 2*-Dimensionen die vollständige Historie mit Gültigkeitsaussagen dokumentieren. *Typ 3*-Dimensionen bilden den aktuellen und den Vorgängerwert einer Hierarchiestufe ab (Kimball 2009, S. 389ff.).

Nachfrageorientierte Datenbereitstellung Der historisierte Datensatz wird zur nachfrageorientierten Datenbereitstellung in den entsprechenden persistenten Datenbestand übernommen. Die Erweiterung erfolgt inkrementell. Zur Realisierung der multidimensionalen Datenstruktur sowie der OLAP-Operationen stehen alternative Konzepte auf logischer und technischer Ebene zur Verfügung. Die Auswahl erfolgt in Abhängigkeit von den Nutzeranforderungen (vgl. Abschn. 5.3.2). Dies wird durch den Bezeichner *nachfrageorientiert* nochmals verdeutlicht. Nachfolgend sind nur einige Aspekte der Gestaltung der Innensicht einer Data-Mart-Komponente genannt. Zur Realisierung des multidimensionalen Datenmodells

sowie der OLAP-Zugriffe stellt neben ROLAP, MOLAP und HOLAP (vgl. Kap. 3.3.3) auch die Realisierung auf Basis neuartiger *NoSQL*-Datenbankverwaltungssysteme und -Technologien eine Alternative dar. Anhand der Anforderungen sowie anhand zu verwaltender Datenstrukturen ist hier eine passende Auswahl zu treffen. Das in einem Data Mart verwaltete Datenvolumen sowie damit assoziierte Anforderungen, beispielsweise an das Antwortzeitverhalten, bilden weitere Einflussgrößen auf die Gestaltung. Als Möglichkeit zur Verbesserung der Anfrageperformance sind materialisierte Sichten mit voraggregierten Daten, wie beispielsweise bei Golfarelli & Rizzi (2009, S. 257ff.) beschrieben, zu nennen. Ebenso kann die Verwendung von Hauptspeichertechnologien, wie beispielsweise in verwandten technischen Arbeiten unter Abschn. 5.2 ausgeführt, eine Verbesserung des Antwortzeitverhaltens bewirken. Je nach Realisierungsform des multidimensionalen Datenmodells stehen spezifische Möglichkeiten zur Realisierung nicht-funktionaler Anforderungen zur Verfügung. Auf eine detaillierte Betrachtung aller Gestaltungsparameter wird mit Verweis auf die Zielsetzung der Arbeit verzichtet.

5.5 Integration prozess- und themenorientierter Data-Warehouse-Systeme

In Abschn. 5.1 wird unter Anforderung A5 die Integrierbarkeit von pDWH-Systemen mit prozessübergreifenden themenorientierten DWH-Systemen gefordert. Den Hintergrund dieser Anforderung stellt die angestrebte Aussagenkonsistenz beider Systeme dar. In Kap. 3.5.1 ist die Abgrenzung sowie der Zusammenhang zwischen themen- und prozessorientiertem DWH-System dargestellt. Ein pDWH-System stellt einen Prozesstyp in den Mittelpunkt. Innerhalb dieses Rahmens werden durch Datensätze auf feiner Granularität und mit Prozessmerkmalen (z.B. Ablaufinformationen) sowohl das struktur- als auch das verhaltensorientierte pDWH-Szenario unterstützt. Das themenorientierte DWH-System betrachtet ein System von Prozessen und ist klassisch auf die strategische Entscheidungsunterstützung ausgerichtet. Die Betrachtung eines Systems von Prozessen und damit heterogener Quellsysteme in themenorientierten DWH-Systemen rückt die Notwendigkeit der syntaktischen und semantischen Bereinigung bei Integration der Daten verstärkt in den Fokus. Die Daten eines themenorientierten DWH-Systems werden zudem i.d.R. auf einem höheren Aggregationslevel sowie ohne Prozess-

merkmale verwendet. Der Unterschied zwischen beiden Systemen kann zusammenfassend über die (i) heterogene Reichweite, (ii) den heterogenen Aggregationsgrad sowie (iii) die Verwendung von Prozessmerkmalen beschrieben werden.

Konzepte zur Integration werden auf Basis der *flexiblen Data-Mart-Architektur* (vgl. Abschn. 5.3.2) eingeführt. Bei Sinz et al. (2001) wird die Ausrichtung eines DWH-Systems an der Führungs- und Organisationsstruktur eines Unternehmens als Gestaltungsmerkmal gefordert. Dies wird aufgegriffen und auf die Integrationslösung übertragen. Aus der organisationalen Nutzung der beiden DWH-Systemtypen kann eine hierarchische Struktur abgeleitet werden. Diese Beziehung wird auch durch die regelkreisorientierte Einordnung beider Systeme, wie in Kap. 3.5.2 beschrieben, deutlich. Während das pDWH-System als Hilfsregelstrecke der GP-Lenkung eingeordnet wird, bildet das themenorientierte DWH-System die Hilfsregelstrecke des hierarchisch übergeordneten Regelkreises. Diese Einordnung wird durch weitere Kriterien bestätigt: Während das unternehmensweite DWH-System damit nur wenigen Nutzern auf strategischer oder taktischer Managementebene zur Verfügung steht, entstammen die Zielgruppen eines pDWH-Systems mehrheitlich der untergeordneten operativen und taktischen Ebene. Auch die Betrachtung der Reichweite (System von Prozessen vs. einzelner Prozess) sowie der heterogenen Granularität der Daten beider Systeme unterstreichen die hierarchische Beziehung. Aufbauend auf dem Konzept des *hierarchischen DWH-Systems* nach Sinz & Ulbrich-vom Ende (2010) erfolgt die Integration der *flexiblen Data-Mart-Architektur* und themenorientierter DWH-Systeme. Die *hierarchische DWH-Architektur* ermöglicht die Zusammenführung heterogener Datenbestände und die Einbindung externer Datenquellen (Sinz & Ulbrich-vom Ende 2010). Die *flexible Data-Mart-Architektur* bietet mit der *angebotsorientierten* und der *nachfrageorientierten Datenbereitstellung* zwei Ansatzpunkte zur Integration.

In Abb. 5.6 ist die Integration über die *angebotsorientierte Datenbereitstellung* und damit über die Schnittstelle *aufbereitetes Exportschema* dargestellt. Die vom themenorientierten DWH-System benötigten (ggf. multidimensional aufbereiteten) Daten werden auf Granularität des prozessorientierten Systems abgegriffen. Syntaktische und semantische Bereinigung, Integration und Historisierung der Daten stellen zentrale Aufgaben des ETL-Prozesses (Extraktion-Transformation-Laden) des themenorientierten Systems dar.

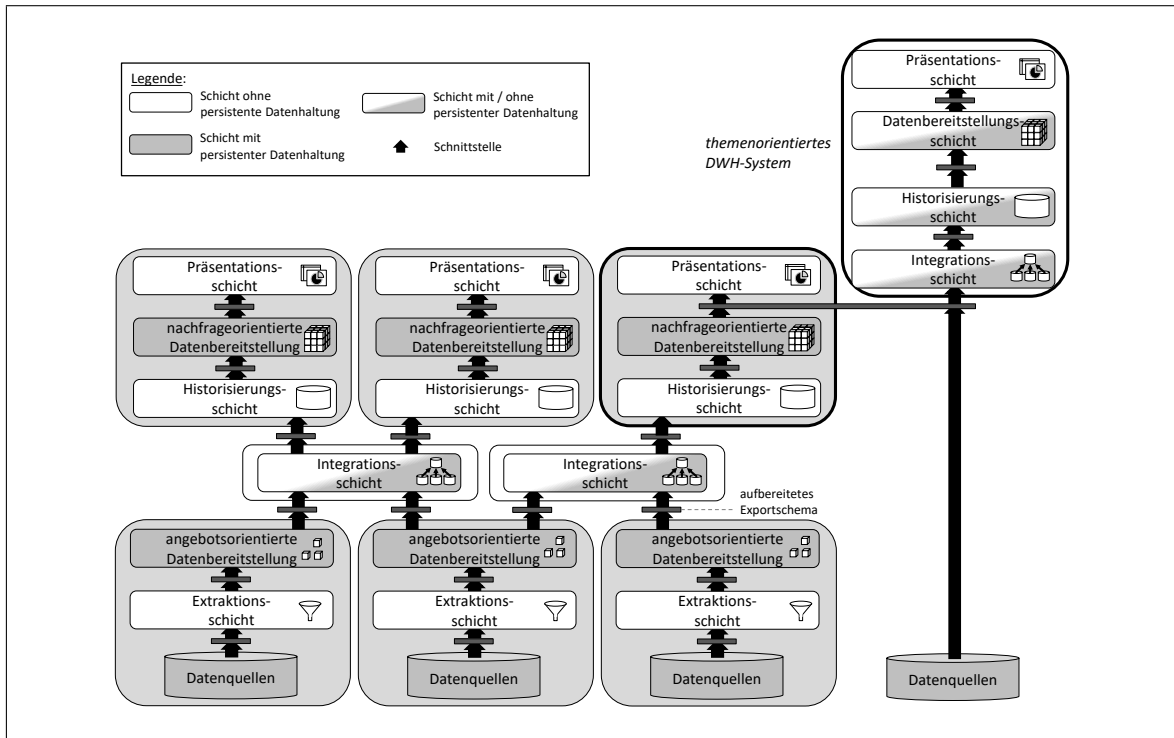


Abb. 5.7: Nachfrageorientierte Datenbereitstellung zur Integration themen- und prozessorientierter Data-Warehouse-Systeme

mit genannten Charakteristika zurückgegriffen werden. *Partitionierungs-* und *Allokationsschema* beschreiben die Aufteilung des *globalen Analyseschemas* sowie dessen Verteilung auf *erweiterte Datenquellen* (bzw. operative Datenkomponenten). In den lokalen *Komponentenschemata* liegen die multidimensionalen Daten in nicht integrierter, nicht historisierter und nicht voraggregierter Form vor. Zur angebotsorientierten Integration sind die benötigten Daten aus den jeweiligen Komponenten bzw. den multidimensionalen Komponentenschemata zusammenzuführen.

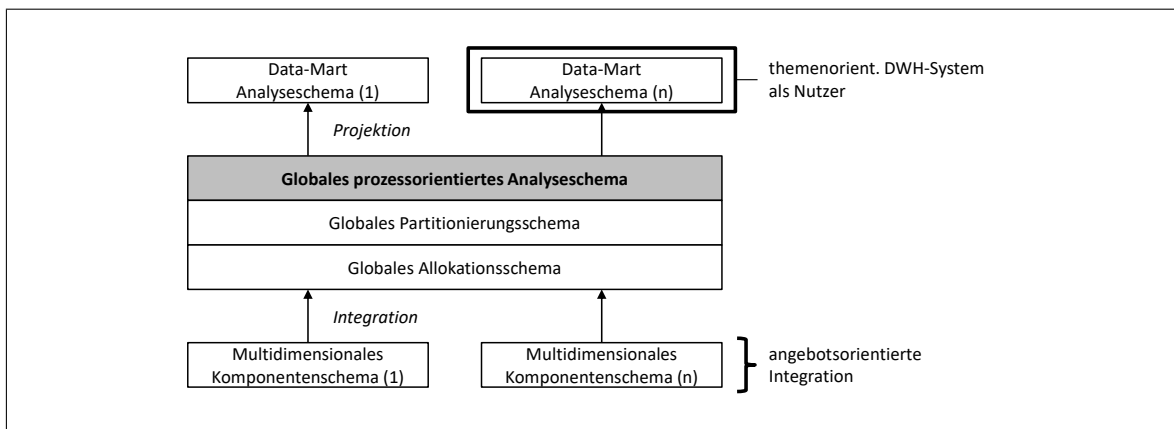


Abb. 5.8: Kartierung der Integration themen- und prozessorientierter Data-Warehouse-Systeme in der Architektur des Analyseschemas

Die Auswahl der passenden Form der Integration ist im Einzelfall in Abhängigkeit von der Spezifikation des ETL-Prozesses, weiteren Datenquellen und anderen Kontextfaktoren zu entscheiden. Beide Formen schlagen eine hierarchische Anordnung von themen- und prozessorientiertem DWH-System vor. Durch Verwendung des pDWH-Systems als Vorsystem übergreifender DWH-Systeme werden Nutzeranfragen in beiden Systemen aus konsistenten Datenbasen heraus beantwortet und damit das Risiko widersprüchlicher Aussagen minimiert.

5.6 Zusammenfassung und Einordnung der Ergebnisse

Das vorgestellte Architekturkonzept wird nachfolgend zusammengefasst und gegen die Anforderungen aus Abschn. 5.1 gespiegelt. Aufbauend auf einer generischen, funktionsorientierten Architekturbeschreibung für DWH-Systeme wird in einer ersten Ausbaustufe eine Referenzarchitektur für prozessorientierte DWH-Systeme (*flexible Data-Mart-Architektur*, Abschn. 5.3.2) konzipiert. In der zweiten Ausbaustufe erfolgt die Integration mit operativem und EU-System zur komponentenorientierten Referenzarchitektur für operativ-dispositive Systeme (Abschn. 5.3.3). Grundlage der Konzeption bilden Anforderungen, die anhand der Literatur sowie der pDWH-Szenarien (struktur- und verhaltensorientiert, Kap. 3.5.2) abgeleitet sind.

Die *flexible Data-Mart-Architektur* wird funktionsorientiert beschrieben. Zur Realisierung der Datenbereitstellung mit angemessener Latenz (A1) sowie der flexiblen Anpassbarkeit (A3) wird die Funktion der Datenbereitstellung anhand der Markt-Metapher in ein angebots- sowie ein nachfrageorientiertes Teilsystem geteilt. Zur angebotsorientierten Datenbereitstellung wird der Verantwortungsbereich von Datenquellen v.a. um Extraktions- und Transformationsaufgaben erweitert. Zur nachfrageorientierten Datenbereitstellung und somit zur Informationsversorgung verschiedener Nutzergruppen (A2) wird das etablierte Data-Mart-Konzept angewendet. Die Brückenfunktion zwischen angebots- und nachfrageorientiertem Teilsystem übernimmt die Datenintegration, ausgelöst über fachliche Ereignisse in den verschiedenen Systembestandteilen. Für alle Komponenten der *flexiblen Data-Mart-Architektur* wird ein generisches, parametrisierbares Lösungsverfahren angegeben, sodass die Anwendbarkeit in heterogenen betrieblichen Anwendungsdomänen im Sinne einer Referenzarchitektur möglich ist.

Das Paradigma der komponentenorientierten Systembeschreibung wird zur Realisierung der *flexiblen Data-Mart-Architektur* sowie zu deren Integration (u.a. bidirektionaler Datentransfer, Anforderung A4) mit operativem und EU-System angewendet. Diese resultierende zweite Ausbaustufe der Referenzarchitektur ist zur Realisierung operativ-dispositiver AwS geeignet. Für Aufgaben erweiterter Datenquellen, der Datenintegration und der Data Marts werden abgeschlossene Komponententypen gebildet. Diese sind durch zu unterstützende Datenstrukturen parametrisierbar und können über generische Schnittstellen miteinander integriert werden. Das Konzept der losen Kopplung fördert die flexible Anpassbarkeit des Systems. Der zentralen Zielsetzung der Komponentenorientierung, maximale Kohäsion bei minimaler Kopplung (siehe Kap. 2.3.2) wird auf Typ-Ebene durch klar definierte Zuständigkeiten der Komponententypen begegnet. Innerhalb von Datenkomponenten bilden sowohl das verwaltete Datenschema, operative CRUD-Zugriffe sowie die angebotsorientierte Datenbereitstellung die Abgrenzung. Weitere Komponententypen bilden sich anhand der Integrationsaufgabe, der Aufgabe zur nachfrageorientierten Bereitstellung, der Entscheidungsunterstützung sowie der operativen Vorgangsteuerung. Die Kopplung der Komponententypen bzw. deren Interaktion ist klar definiert und kann aus fachlicher Sicht als minimal angesehen werden.

Eine zentrale Anforderung an die operativ-dispositive Referenzarchitektur stellt die Datenbereitstellung und Reaktion mit angemessener Latenz (A1) dar. Angemessen ist nicht im Sinne von *real-time* zu verstehen, sondern richtet sich nach den Anforderungen der jeweiligen Anwendung. Die Realisierung angemessener, auch kurzer Latenz ist dabei im Wesentlichen von Entscheidungen auf technischer Ebene beeinflusst. Die fachliche Referenzarchitektur bietet hierzu einen geeigneten Rahmen, der jedoch bei Anwendung entsprechend technologisch auszugestalten ist. Die Nutzung von In-Memory-Systemen wie *SAP HANA* (vgl. Kap. 5.2) zur persistenten Datenverwaltung der Architektur stellt eine mögliche Überlegung dar. Je nach Anforderungen der konkreten Anwendung ergeben sich auch Potenziale durch Anpassung der Referenzarchitektur. So beeinflussen Komponenten mit persistenter Datenhaltung sowie deren Interaktion über Schnittstellen die Latenz. Die Trennung ist im Wesentlichen der Informationsversorgung verschiedener Nutzergruppen geschuldet. Sofern diese Anforderung entfällt, besteht Optimierungspotenzial der Latenz durch Zusammenlegen der pDWH-Funktionen.

Das vorgeschlagene Konzept zur Integration von prozess- und unternehmensweiten DWH-Systemen (Anforderung A5) basiert auf der *flexiblen Data-Mart-Architektur*. Auch hier bieten sowohl die angebots- als auch die nachfrageorientierte Datenbereitstellung Ansatzpunkte zur hierarchischen Integration. Das zentrale Unterscheidungskriterium ist die Auslagerung von Bereinigungs-, Historisierungs- und Integrationsaufgaben aus dem themenorientierten DWH-System in das Vor-system. Die Auswahl der vorgestellten Alternativen hängt vom konkreten Einsatzszenario ab. Die vollständige Realisierung eines themenorientierten DWH-Systems auf Basis der vorgeschlagenen Referenzarchitektur erscheint hingegen wenig sinnvoll. Dies kann durch abweichende Anforderungen, beispielsweise zur Latenz der Datenbereitstellung, Flexibilität der Anpassung oder Integration mit operativen Systemen begründet werden. Die Trennung in angebots- und nachfrageorientierte Datenbereitstellung ist in themenorientierten DWH-Systemen nicht notwendig. Auch die Reichweite themenorientierter DWH-Systeme mit steigender Komplexität der Integrationsaufgaben wirkt sich auf die Komplexität der *flexiblen Data-Mart-Architektur* aus. Etablierte Architekturen themenorientierter DWH-Systeme haben aus praktischer Sicht ihre Eignung hierzu nachgewiesen. Zudem ist von einem weitverbreiteten Einsatz themenorientierter DWH-Systeme auszugehen, so dass die Integration dieser mit einem pDWH-System anstatt die Neuentwicklung im Fokus der Arbeit steht.

6 Datenstrukturen prozessorientierter Data-Warehouse-Systeme

Kapitelzusammenfassung: Kap.6 beschäftigt sich mit drei zentralen Fragestellungen: (i) Welchem Strukturierungsprinzip sollte die Datenbasis eines pDWH-Systemes folgen, um sowohl für das verhaltens- als auch das strukturorientierte pDWH-Szenario anwendbar zu sein? (ii) Welche inhaltlichen Anforderungen müssen von deren Daten erfüllt werden, um Entscheidungen innerhalb dieser Szenarien angemessen zu unterstützen? (iii) Wie können die notwendigen Daten innerhalb der Referenzarchitektur aus Kap. 5 erfasst und bereitgestellt werden? Grundlage der Konzeption bildet die BPMN als aktueller de facto-Standard zur Modellierung von Workflows (Abschn. 6.1). Auf Basis der BPMN sowie anhand der Literatur werden Anforderungen an Inhalt und Struktur des pDWH-Datenschemas abgeleitet (Abschn. 6.2) und gegen bestehende Forschungsarbeiten gespiegelt (Abschn. 6.3). Ausgehend von diesen Anforderungen sowie auf Grundlage des Metamodells der BPMN wird ein generisches multidimensionales pDWH-Schema konzipiert, das neben Ablaufdaten auch Dimensionen fachlicher Domänen berücksichtigt (Abschn. 6.4). In Abschn. 6.5 erfolgt schließlich die Zuordnung der Datenstrukturen zu Komponententypen der Referenzarchitektur.

Das Kapitel basiert in Teilen auf der Veröffentlichung Benker (2016).

6.1 Grundlagen der Business Process Model and Notation

Die *Business Process Model and Notation* (BPMN) stellt, als Spezifikation der Object Management Group (OMG) (OMG 2011), eine verbreitete und von Modellierern sowie Werkzeugherstellern anerkannte Prozessmodellierungssprache dar (Silver 2011, S. V). Als eine der dominanten graphischen Modellierungssprachen für Prozesse (im Speziellen für Workflows) (Pütz & Sinz 2010) wird die BPMN in dieser Arbeit als Grundlage zur Ableitung multidimensionaler Schemata für pDWH-Systeme eingesetzt. Hierzu werden nachfolgend Grundlagen zu Diagrammart

(Abschn. 6.1.1), Konformitätsklassen (Abschn. 6.1.2) und Metamodell (Abschn. 6.1.3) eingeführt.

6.1.1 Diagrammarten der Business Process Model and Notation

Die BPMN sieht die vier Diagrammarten *Prozessdiagramm*, *Kollaborationsdiagramm*, *Konversationsdiagramm* und *Choreographiediagramm* vor (Weske 2012, S. 206). Gegenstand von pDWH-Systemen sind Ablaufdaten des Prozesses eines betrieblichen Objektes. In der Literatur wird dies durch den Begriff *Prozess Orchestrierung* (z.B. Weske 2012, S. 125) abgedeckt. Zur Modellierung von Orchestrierungen mit der BPMN sind Prozessdiagramm und Kollaborationsdiagramm geeignet (Weske 2012, S.206). Diese beiden Diagrammarten werden nachfolgend im Detail vorgestellt. Das Choreographiediagramm dient der Spezifikation des Interaktionsprotokolls zwischen betrieblichen Objekten (OMG 2011, S. 315), das Konversationsdiagramm der Darstellung von Interaktionsstrukturen (OMG 2011, S. 26). Beide Diagrammtypen liefern keinen zusätzlichen Beitrag zur Prozessanalyse.

- **Prozessdiagramm** (OMG 2011, S. 24): Ein Prozess ist nach dem Verständnis der BPMN einem betrieblichen Objekt (*Participant*, Teilnehmer) zugeordnet. Unterschieden wird dabei zwischen *private* und *public process*. Durch den *private process* wird der Arbeitsablauf (Workflow) bzw. das Lösungsverfahren des Teilnehmers aus Innensicht beschrieben. Der *public process* fokussiert auf die Außensicht des Prozesses und beschreibt damit lediglich die Prozessbestandteile, die zur Interaktion mit benachbarten Teilnehmern (anderen Prozessen) notwendig sind.
- **Kollaborationsdiagramm** (OMG 2011, S. 24, Weske 2012, S. 236ff.): Die Interaktion zwischen (mindestens zwei) Teilnehmern ist Gegenstand des Kollaborationsdiagramms. Die Interaktion erfolgt über Nachrichtenflüsse zwischen den Teilnehmern, die als *Pools* modelliert werden. Ein Kollaborationsdiagramm kann um die Prozessdiagramme (*private* oder *public process*) der einzelnen Teilnehmer ergänzt werden. Alternativ ist auch die Darstellung eines Teilnehmers ohne Prozessdetails (sog. *Black Box*) zulässig.

In Abb. 6.1 ist das Kollaborationsdiagramm zur Auftragsverwaltung aus der Fallstudie *Mobility Provision* (Kap. 4) dargestellt. Die Teilnehmer *Vertriebskanal*, *Auftragsverwaltung*, *Finanzwesen* sowie *Flotte* sind in Form von *Pools* modelliert und

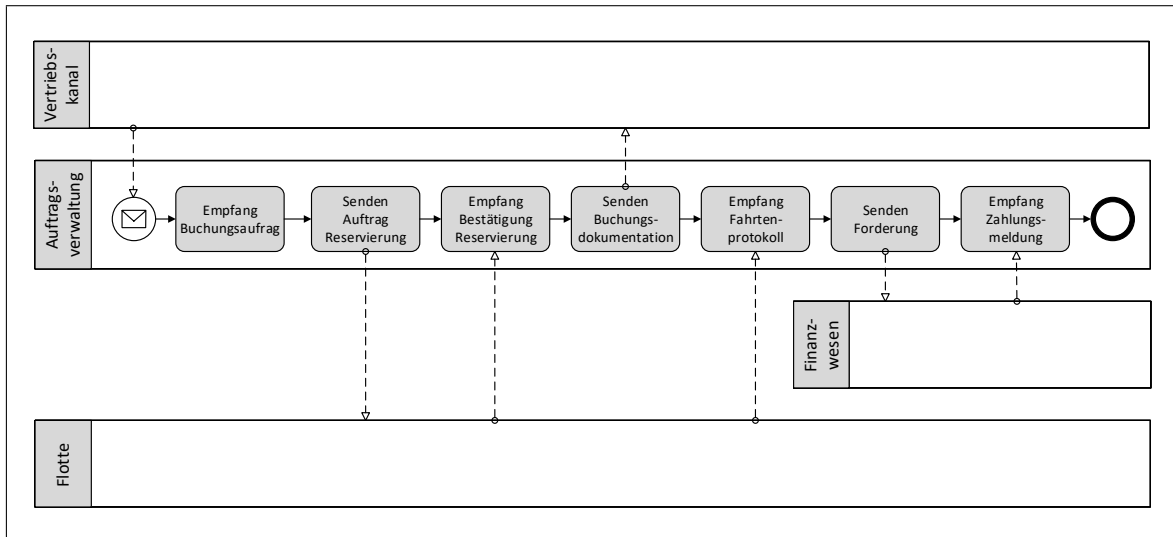


Abb. 6.1: Kollaborationsdiagramm (public process) zur Auftragsverwaltung

interagieren über Nachrichtenaustausch. Für den Teilnehmer *Auftragsverwaltung* ist das Prozessdiagramm als *public process* angegeben. Es beschreibt den Sequenzfluss der Aktivitäten zur Interaktion mit den übrigen Teilnehmern. Diese sind als *Black Box* spezifiziert.

6.1.2 Konformitätsklassen der Business Process Model and Notation

Die Spezifikation der BPMN sieht zudem drei Klassen vor, anhand derer sich die Konformität von Werkzeugen mit Blick auf die Spezifikation bestimmen lässt. Jede Klasse definiert als Mindestanforderung einen Ausschnitt des OMG-Standards, der von *konformen* Werkzeugen zu unterstützen ist. Nachfolgend werden die Konformitätsklassen kurz vorgestellt:

- **Process Modeling Conformance** (OMG 2011, S. 2, Weske 2012, S. 207): Mit dieser Klasse wird die Spezifikation von Prozess-, Kollaborations- und Conversationsdiagrammen abgedeckt. Alternativ zur vollen *Process Modeling Conformance* werden drei abgestufte Sub-Klassen unterschieden:
 - *descriptive*: enthält *grundlegende* Elemente der BPMN zur Prozessmodellierung.
 - *analytic*: erweitert die Elemente der Sub-Klasse *descriptive* und ist auf erfahrene Modellnutzer ausgerichtet.

- *common executable*: umfasst Elemente zur Spezifikation *ausführbarer* Prozesse.
- **Process Execution Conformance** (Weske 2012, S.207): In dieser Klasse werden Werkzeuge eingeordnet, die die Ausführungssemantik sowie den Lebenszyklus von BPMN-Prozessen unterstützen. Die Klasse *BPEL Process Execution Conformance* stellt einen Spezialfall dar. Hier verortete Werkzeuge unterstützen zusätzlich die Abbildung von BPMN-Prozessen auf die Implementierungssprache *Web Services Business Process Execution Language* (WS-BPEL).
- **Choreography Modeling Conformance** (Weske 2012, S.207): Die Unterstützung von Choreographie- und Kollaborationsdiagrammen stellt die Voraussetzung zur Modellierung von Choreographien und zur Einordnung in diese Konformitätsklasse dar.

6.1.3 Das Metamodell der Business Process Model and Notation

Das Metamodell der BPMN ist auf die für die Arbeit relevanten Aspekte beschränkt und anhand der Konformitätsklassen aus Abschn. 6.1.2 abgegrenzt. Zielsetzung eines pDWH-Systems stellt die Bereitstellung entscheidungsrelevanter Prozessdaten dar. Als relevant werden der Prozessablauf sowie die Interaktion zwischen Teilnehmern angesehen, wie in Kollaborations- und Prozessdiagrammen (vgl. Abschn. 6.1.1) beschrieben. Das Metamodell wird somit auf die Unterstützung der Elemente der Klasse *Process Modeling Conformance* eingegrenzt. Zur Beherrschung der Komplexität des BPMN-Metamodells erfolgt die Einschränkung zunächst auf die Sub-Klasse *Descriptive Conformance*. Diese enthält alle grundlegenden Elemente zur Abbildung eines Prozesses sowie dessen Interaktion mit in Beziehung stehenden Teilnehmern. Eine Diskussion zur Unterstützung der vollen *Process Modeling Conformance* erfolgt in Abschn. 6.4.4. Abb. 6.2 zeigt das resultierende, hinsichtlich der Zielsetzung der Arbeit verkürzte Metamodell der BPMN, Abb. 6.3 die zugehörigen graphischen Darstellungen. Die Bestandteile des Metamodells werden nachfolgend eingeführt.

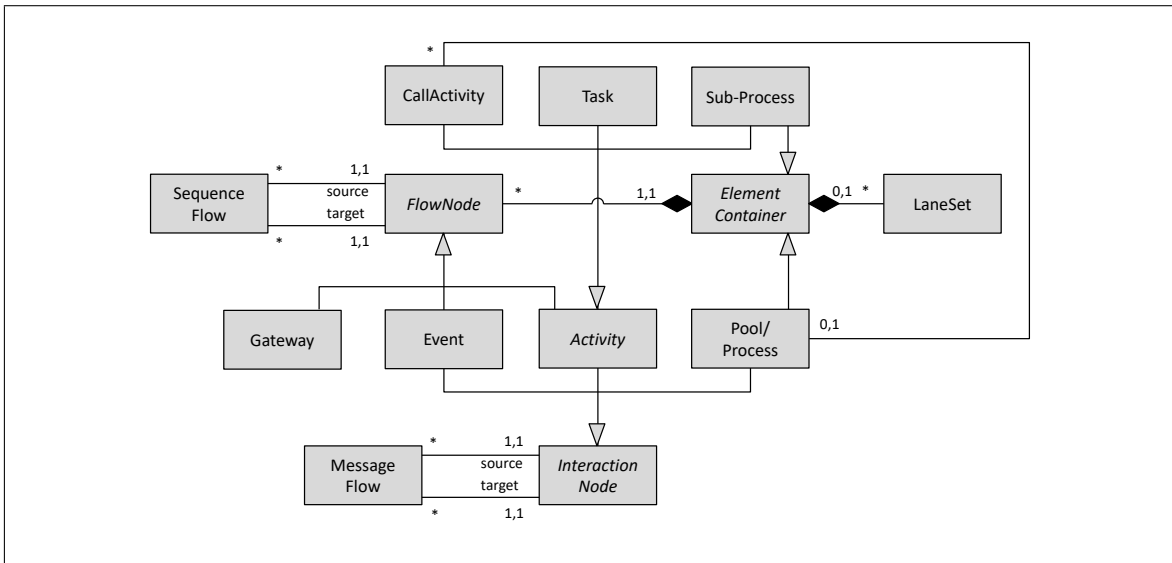


Abb. 6.2: Metamodell der Business Process Model and Notation (OMG 2011) (eigene, verkürzte Darstellung)

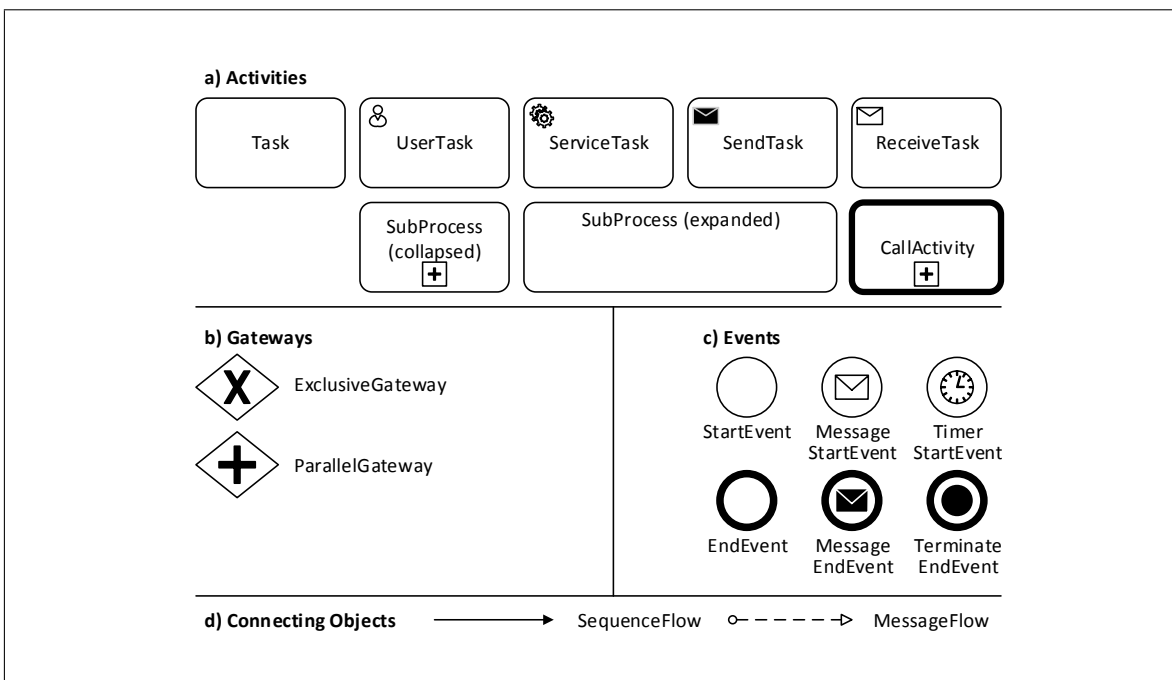


Abb. 6.3: Grafische Syntax der Business Process Model and Notation (OMG 2011) (eigene Zusammenstellung)

Flow Objects

Als *Flow Objects* (im Metamodell in Abb. 6.2 als *FlowNode* bezeichnet) werden die zentralen Elemente zur Modellierung des Prozessverhaltens verstanden. Es werden Aktivitäten (*Activities*), *Gateways* und Ereignisse (*Events*) differenziert:

Aktivitäten *Aktivitäten* stellen Arbeitsschritte innerhalb eines Prozesses dar. Dabei werden die Spezialisierungen *Task*, *Sub-Process* und *Call Activity* unterschieden. Die grafische Repräsentation der Elemente ist in Abb. 6.3a dargestellt. Atomare Aktivitäten (als *Task* bezeichnet) können nicht weiter verfeinert werden (Weske 2012, S. 212). Über Sub-Typen können sie näher spezifiziert werden. Die relevanten Sub-Typen der *Descriptive Modeling Conformance* werden nachfolgend vorgestellt:

- **UserTask** (Weske 2012, S. 38): Die Durchführung der Aktivität erfolgt teilautomatisiert. Der personelle Aufgabenträger arbeitet mit Unterstützung eines AwS bzw. eines Workflow-Management-Systems.
- **ServiceTask** (Weske 2012, S. 38): Die automatisierte Durchführung einer Aufgabe wird als *ServiceTask* beschrieben.
- **Send-/ReceiveTask** (OMG 2011, S. 160ff.): Über einen *SendTask* wird der Nachrichtenversand an einen Teilnehmer modelliert. Über einen *ReceiveTask* wird analog der Nachrichtenempfang spezifiziert.

Ein *Sub-Process* stellt eine Aktivität innerhalb eines übergeordneten Prozesses dar und kann selbst wiederum als Prozess beschrieben werden. Die Darstellung innerhalb des Prozessdiagramms kann erweitert (*expanded*) oder nicht erweitert (*collapsed*) erfolgen (Silver 2011, S. 21). Der Aufruf unabhängiger, beispielsweise wiederverwendbarer (Teil-)Prozesse wird über das Element *Call Activity* modelliert (Silver 2011, S. 38f.). Die Spezifikation der aufgerufenen Prozesse erfolgt dabei in separaten Diagrammen.

Gateways Die Steuerung des Prozessablaufs erfolgt über *Gateways* (Abb. 6.3b) (OMG 2011, S. 287). Ein *ExclusiveGateway* beschreibt die Aufspaltung (*split*) in bzw. die Zusammenführung (*join*) von alternativen Sequenzflüssen (OMG 2011, S. 290ff.). Ein *ParallelGateway* spezifiziert hingegen die Erzeugung bzw. die Synchronisation paralleler Abläufe (OMG 2011, S. 293ff.).

Ereignisse *Ereignisse* können anhand der Position im Prozess (Start-, Zwischen- und Endereignis) sowie anhand ihres Auslösers differenziert werden (Weske 2012, S. 216). Nachfolgend werden die in der *Descriptive Process Modeling Conformance* vorgesehenen Ereignistypen (Abb. 6.3c) eingeführt (Weske 2012, S. 216ff.):

- **(None) Start-/EndEvent:** Der unspezifizierte Ereignistyp wird verwendet, wenn die Art des Ereignisses nicht relevant oder nicht bekannt ist.
- **Message Start-/EndEvent:** Das *Message StartEvent* signalisiert das Starten eines Prozesses mit Eingang einer Nachricht. Das *Message EndEvent* signalisiert das Senden einer Nachricht an einen anderen Teilnehmer bei Terminierung des Prozesses. Das Ereignis wird aktiv durch den Prozess ausgelöst.
- **Timer StartEvent:** Zeitbezogene Ereignisse können auf Basis von Zeitspannen, Zeitpunkten oder Zeitdauern spezifiziert werden. Im Fall des *StartEvents* wird der Prozessstart mit Eintreten des zeitbezogenen Ereignisses modelliert.
- **Terminate EndEvent:** Das Ereignis signalisiert das *sofortige* Ende eines Prozesses.

Connecting Objects

Über den Sequenzfluss (*SequenceFlow*) wird der Prozessablauf, also die Reihenfolge der *FlowNodes* innerhalb eines Prozesses, beschrieben (OMG 2011, S. 97). Ein Sequenzfluss-Element verbindet dabei immer genau zwei *FlowNodes*. Die prozessübergreifende Kommunikation und damit der Nachrichtenaustausch zwischen Teilnehmern wird durch den *MessageFlow* spezifiziert (OMG 2011, S. 112). Die dabei ausgetauschten Nachrichten können an den Kanten des Nachrichtenflusses annotiert werden. Die graphische Darstellung der *Connecting Objects* ist in Abb. 6.3d gezeigt.

Swimlanes

Unter dem Begriff *Swimlane* werden die Modellelemente *Pool* und *Lane* zusammengefasst. Ein *Pool* stellt die graphische Repräsentation eines Teilnehmers einer Kollaboration dar und kann als *ElementContainer* den Prozess des jeweiligen Teilnehmers beinhalten (OMG 2011, S. 112). Dieser kann durch die Verwendung von *Lanes* (in Abb. 6.2 als *LaneSet* bezeichnet) weiter strukturiert werden (OMG 2011, S. 120). *Lanes* werden typischerweise zur Modellierung von Rollen bzw. Verantwortlichkeiten innerhalb eines Prozesses verwendet (Silver 2011, S. 20). Die grafische Darstellung von *Pools* kann dem Beispiel aus Abb. 6.1 entnommen werden.

Sonstige Elemente

Als weitere Modellelemente sind im *Descriptive Conformance Level* Datenobjekte und ergänzende Artefakte vorgesehen. Datenobjekte können mit Aktivitäten (Input, Output) assoziiert werden. Der Begriff *Data Object* ist in der BPMN jedoch weit gefasst. Er deckt neben digitalisierten Informationen ebenfalls physische Objekte und Produkte ab (Weske 2012, S.230). Entsprechend weit gefasst ist auch der Begriff *Data Store* zur Verwaltung von Datenobjekten. Das vorliegende Begriffsverständnis ist für die Arbeit nicht zielführend. Die ergänzenden Artefakte *Text Annotation* (beliebige textuelle Ergänzung) und *Group* (beliebige Gruppierung grafischer Elemente) sind bzgl. der Prozessausführung und damit in pDWH-Systemen nicht von Relevanz. Die beschriebenen Modellelemente werden aus genannten Gründen im Weiteren nicht berücksichtigt.

6.2 Anforderung an die Datenstrukturen prozessorientierter Data-Warehouse-Systeme

Nachfolgend werden Anforderungen an das Datenschema eines pDWH-Systems zur Realisierung der in Kap. 3.5.2 eingeführten Anwendungsszenarien erhoben. Hierzu wird in Abschn. 6.2.1 zunächst die Frage nach der grundsätzlichen Strukturierung auf Metaebene, also einem geeigneten Datenmodell, betrachtet. In Abschn. 6.2.2 werden im Weiteren generische Anforderungen zur Ausgestaltung darauf aufbauender Datenschemata beleuchtet.

6.2.1 Anforderungen an das Datenmodell

Für beide pDWH-Szenarien wird grundsätzlich die Bereitstellung entscheidungsrelevanter Daten in multidimensionaler Form, wie in Kap. 3.3 eingeführt, vorgeschlagen. Im noch folgenden Abschn. 6.3 werden verwandte Arbeiten vorgestellt, die den Einsatz von DWH-Systemen und multidimensionalen Datenstrukturen für das *strukturorientierte* Szenario motivieren. Als Datenquellen dienen hierzu i.a.R. *Log-Dateien*, die die Ausführung automatisierter Prozesse aufzeichnen. Durch die multidimensionale Strukturierung dieser Ablaufdaten wird, vor der Zielsetzung der effizienteren und innovativeren Prozessgestaltung, ein geeigneter Zugang zu den Prozessinformationen ermöglicht (z.B. Eder et al. 2002, Mansmann et al. 2007b, Schiefer et al. 2004). Auch für das *verhaltensorientierte* Szenario

rio wird in der Literatur die Anwendung des multidimensionalen Datenmodells motiviert. Bei Eckerson (2007) und Russom (2011) wird die Bereitstellung auch kurzfristig benötigter Daten in einem historisierten Datenbestand vorgeschlagen, um neben zeitpunktbezogenen Analysen auch Daten und Ereignisse im Zeitverlauf bewerten zu können. Mit der Bereitstellung der Daten in multidimensionaler und aggregierter Form wird zudem die Anwendbarkeit von OLAP-Operatoren ermöglicht und als Mehrwert genannt. Nach ElegantJ BI (2008) führt deren ggf. auch kurzfristige Anwendung auf Unternehmensdaten zu einem tieferen Verständnis in operativen Entscheidungssituationen. In bestimmten Fällen kann jedoch auch eine alternative Datenstruktur sinnvoll erscheinen, wenn beispielsweise Performance-Aspekte für die Realisierung entscheidend sind (Russom 2011). In der Arbeit wird dieser Fall nicht betrachtet und der Fokus auf das multidimensionale Datenmodell gelegt.

Zusammenfassend ergibt sich nachfolgende **Anforderung an das Datenmodell**: *Zur Strukturierung bzw. Bereitstellung der pDWH-Datenbasis sollte sowohl im struktur- als auch verhaltensorientierten pDWH-Szenario das multidimensionale Datenmodell angewendet werden.*

6.2.2 Anforderungen an das Datenschema

Anforderungen zur Ausgestaltung eines DWH-Schemas ergeben sich i.a.R. aus dem jeweiligen Anwendungskontext sowie der mit dem System verfolgten Zielsetzung. Für beide pDWH-Szenarien können anhand ihrer Zielsetzungen sowie auf Basis der Literatur Anforderungen unabhängig von der betrieblichen Anwendungsdomäne abgeleitet werden. Nachfolgend werden sowohl für beide Szenarien gültige als auch szenariospezifische Anforderungen eingeführt.

Szenarioübergreifende Anforderungen an das Datenschema

Zunächst werden Anforderungen zur *Granularität der Daten* sowie zur *flexiblen Anpassbarkeit* erläutert, die gleichermaßen für das *struktur-* als auch das *verhaltensorientierte* Szenario gelten.

Die **Anforderung an die Granularität** des Datenbestandes von pDWH-Systemen lässt sich wie folgt zusammenfassen: *Zur Spezifikation der Granularität der pDWH-*

Datenbasis ist die Datengranularität des Quellsystems, also des Prozess-ausführenden Systems, im Speziellen der davon bereitgestellten Ablaufdaten maßgebend. Grundsätzlich sollte der Datenbestand eines DWH-Systems nicht voraggregiert in der Granularität vorliegen, wie er von Prozessinstanzen (bzw. deren ausführenden Systemen) abgegriffen werden kann (Baars & Sun 2013, Inmon 2000). Dennoch muss auch die Möglichkeit zur Abstraktion von einzelnen Instanzen gegeben sein (Castellanos et al. 2009).

Die **Anforderung der flexiblen Anpassbarkeit** des Datenschemas leitet sich aus Anforderung (A3) der Referenzarchitektur (Kap. 5.1) ab. Diese beschreibt die flexible Anpassbarkeit der AwS-Architektur an sich verändernde Anforderungen im Prozessumfeld. Operative Systeme und Prozesse unterliegen im Zeitverlauf Anpassungen, beispielsweise ausgelöst durch verändertes Markt- und Kundenverhalten. Diese Veränderungen wirken sich ebenfalls auf die nachgelagerten DWH-Systeme sowie deren Datenstrukturen aus. Die abgeleitete Anforderung an das pDWH-Datenschema ist nachfolgend zusammengefasst: *Datenstrukturen eines pDWH-Systems sind so zu gestalten, dass neben der Verwaltung prozessbezogener Daten auch auf Veränderungen in Prozessabläufen sowie deren Umfeld flexibel reagiert werden kann.*

Anforderung an das strukturorientierte Datenschema

Für das *strukturorientierte* Szenario ergeben sich Anforderungen aus der Zielsetzung der effizienteren und innovativeren Prozessgestaltung und der hierzu notwendigen Prozessanalyse. Nach Bange & Fuchs (2011) sind Prozesse hinsichtlich Qualität, Kosten und Zeit zu analysieren. Bei Dumas et al. (2013, S. 213ff.) wird als viertes Kriterium zur Bewertung von Prozessen die Flexibilität genannt. Die Ableitung von Kennzahlen zur Prozessanalyse erfolgt analog zum Vorgehen nach Böhnlein (2001, S. S. 351ff.): Zur Bewertung von Geschäftsprozessen werden Kennzahlen auf Basis der erzeugten Leistung (Sachziel) und der damit assoziierten Zielsetzung (Formalziele) ermittelt. Übertragen auf das strukturorientierte pDWH-Szenario wird die Leistung der Prozessgestaltung mit relevanten Zielsetzungen betrachtet. Die von Praktikern verfolgten Zielsetzungen nach Fischermanns (2013, S. 309) können in sechs Hauptzielen zusammengefasst werden: (1) kurze Durchlaufzeiten; (2) niedrige Prozesskosten; (3) hohe Prozessqualität; (4) hohe Produktqualität; (5) hohe Kundenzufriedenheit; (6) hohe Mitarbeitermo-

tivation. Die Erreichung einer für den jeweiligen Prozess angemessenen Flexibilität nach Dumas et al. (2013, S. 213ff.) wird als weiteres Ziel aufgenommen. Zusammen mit Laufzeit, Kosten und Prozessqualität ergeben sich vier Kriterien, die auf Basis von Prozessmodellen bewertet und analysiert werden können. Zur Bewertung von Kundenzufriedenheit und Mitarbeitermotivation erscheinen beispielsweise Fragebögen oder Interviews als geeignet (Fischermanns 2013, S. 309). Produkt- und Prozessqualität können bei Produktionsprozessen getrennt bewertet werden (Fischermanns 2013, S. 311). Die Qualität eines Produktes lässt sich über Prüfverfahren der Qualitätssicherung oder aber anhand von Reklamationen bewerten. Bei Dienstleistungen beeinflusst der Prozess maßgeblich die Leistung und stellt selbst einen Teil dieser dar (Fischermanns 2013, S. 311). Die Produktqualität ist dabei weitestgehend über die Prozessqualität mit bestimmt. Die Ziele (4), (5) und (6) werden, wie beschrieben, nachfolgend nicht als Gegenstand eines pDWH-Systems betrachtet.

Die Analyse von *Qualität*, *Zeit*, *Kosten* und *Flexibilität* soll in einem möglichst umfassenden Kontext erfolgen. Es wird der Vorschlag von Shahzad & Johansson (2009) aufgegriffen, diesen Kontext über die vier grundlegenden Perspektiven der Prozessmodellierung nach Curtis et al. (1992) zu definieren:

- **Funktion** (*Functional*): In der funktionalen Perspektive werden auszuführende Prozesselemente, deren Strukturierung (z.B. in Form von Sub-Prozessen) sowie der Datenfluss zwischen den Elementen betrachtet (Curtis et al. 1992, Korherr 2007, S. 10).
- **Verhalten** (*Behaviorial*): In der Verhaltensperspektive wird der Kontrollfluss des Prozesses beschrieben. Als mögliche Kontrollflusselemente können Schleifen, Sequenzen, parallele oder alternative Pfade genannt werden (Curtis et al. 1992).
- **Organisation** (*Organizational*): Die organisatorische Perspektive beschreibt die Einbettung sowie zugeordnete Verantwortlichkeiten (Rolle u. Akteur) des Prozesses bzw. von Prozesselementen in einer Organisation (Curtis et al. 1992). Nach Korherr (2007, S. 10) kann zwischen Organisationseinheit, Rolle, Art des Aufgabenträgers (automatisiert, nicht-automatisiert) sowie der Zugehörigkeit des Aufgabenträgers (intern, extern) unterschieden werden.

- **Information (Informational):** Die Perspektive *Information* ist auf Entitäten (z.B. Daten, Produkte, Artefakte), die während des Prozessablaufs erzeugt bzw. manipuliert werden fokussiert. Die Struktur der Entitäten sowie deren Beziehungen zueinander sind dabei ebenfalls relevant (Curtis et al. 1992).

Die **Anforderung an das strukturorientierte Datenschema** wird nachfolgend zusammengefasst: *Durch das Datenschema sind entscheidungsrelevante Prozessdaten (zu Prozessablauf u. -ergebnis) so zu berücksichtigen, dass die Eigenschaften Zeit, Kosten, Qualität und Flexibilität im Kontext der Perspektiven der Prozessmodellierung bewertet werden können.*

Anforderung an das verhaltensorientierte Datenschema

Vor dem Hintergrund prozessorientierter Unternehmensstrukturen ist der Prozessbezug in DWH-Systemen für die strategische, taktische und operative Entscheidungsunterstützung zwingend notwendig (Bucher & Dinter 2008b). Ohne die Berücksichtigung des Prozessbezugs bleiben Informationen entweder ungenutzt oder werden z.T. außerhalb ihres Entstehungskontextes verwendet (Bucher & Dinter 2008b). Zielsetzung des *verhaltensorientierten* pDWH-Szenarios ist es, Zugang zu entscheidungsrelevanten Informationen aktiver Prozesse zu schaffen und ggf. steuernd in deren Durchführung einzugreifen. Aufgrund dessen ist es notwendig, betriebswirtschaftliche Daten zu Prozessaktivitäten zuordnen zu können (Panian 2008, Bucher & Dinter 2008b). Nutzer, beispielsweise aus Vertrieb oder Kundenbetreuung, benötigen Informationen mit Bezug zu ihren eigenen Aktivitäten (ElegantJ BI 2008). Von der Berücksichtigung betriebswirtschaftlicher bzw. fachlicher Prozessinformationen wird auch eine positive Rückwirkung auf die Prozessgestaltung, also das *strukturorientierte* Szenario, erwartet. Je breiter die Datengrundlage aufgestellt ist, desto mehr Informationen können in Entscheidungen zur effizienteren und innovativeren Gestaltung und Durchführung von Prozessen einfließen. Der Prozess wird bereits im strukturorientierten Datenschema zum zentralen Untersuchungsgegenstand.

Als **Anforderung an das verhaltensorientierte Datenschema** ergibt sich die *Erweiterung des multidimensionalen und strukturorientierten pDWH-Schemas um Daten der jeweiligen betrieblichen Anwendungsdomäne, die die Ableitung von Maßnahmen zur aktiven Prozesssteuerung unterstützen.*

6.3 Stand der Literatur: Datenstrukturen prozessorientierter Data-Warehouse-Systeme

In Abschn. 6.2 wird die Anwendung des multidimensionalen Datenmodells in pDWH-Systemen als Kernanforderung spezifiziert. Zielsetzung dieses Kapitels ist die Analyse und die Bewertung verwandter Forschungsarbeiten zu Datenschemata von pDWH-Systemen. Als verwandt gelten Arbeiten, die eine multidimensionale Datenstruktur zur Unterstützung des struktur- und/oder des verhaltensorientierten Szenarios zum Gegenstand haben. Diese Arbeiten werden zunächst allgemein kurz vorgestellt und anschließend anhand der in Abschn. 6.2.2 eingeführten Perspektiven der Prozessmodellierung nach Curtis et al. (1992) weiter charakterisiert. Abschließend wird eine Bewertung der Datenmodelle vorgenommen. Zum einen wird überprüft, ob die vorgestellten Datenmodelle die Anforderungen aus Abschn. 6.2 bereits hinreichend berücksichtigen. Zum anderen wird die Mächtigkeit der Datenmodelle zur umfassenden Verwaltung von Ablaufdaten aktueller Workflow-Spezifikationen (am Beispiel BPMN) diskutiert.

Allgemeine Vorstellung verwandter Arbeiten

Bei Becker (2007, S. 223ff.) wird ein generisches multidimensionales pDWH-Schema vorgestellt und durch das ARIS-Metamodell begründet. Als Eigenschaften werden die *flexible Wiederverwendbarkeit* sowie die *Anwendungsneutralität* angestrebt (Becker 2007, S. 225). Innerhalb der DWH-Strukturen werden generische Dimensionen und Kennzahlen (v.a. bzgl. Zeit, Kosten, Qualität und Flexibilität) sowie *Prozesstyp-spezifische* Dimensionen und Kennzahlen berücksichtigt. Zu überwachende *operative Geschäftsprozesse* werden direkt als Workflows instanziiert und auf Basis der *erweiterten Ereignisgesteuerten Prozesskette* spezifiziert. Dimensionen sowie deren Semantik sind somit eng mit Modellbausteinen der eEPK verknüpft. Als Quellsysteme zur Befüllung des Datenschemas werden Ablaufprotokolle von *Workflow-Engines* bzw. operative Datenquellen für geschäftsbezogene Daten verwendet. Bei Casati et al. (2007) wird ebenfalls ein generisches multidimensionales Datenmodell zur Prozessanalyse vorgestellt. Kennzahlen werden dabei je nach Granularität in einem multidimensionalen Würfel auf Prozess- oder auf Aktivitätsebene verwaltet. Semantische Kennzahlen und Dimensionen werden dem generischen Modell nicht berücksichtigt. Diese sind aus dem jeweiligen betrieblichen Anwendungsszenario abzuleiten. Eder et al. (2002) stellen eben-

falls eine multidimensionale Datenstruktur zur Analyse von aufgezeichneten Ablaufinformationen vor. Das zugrundeliegende, proprietäre Workflow-Metamodell wird in einem UML-Klassendiagramm spezifiziert. Ausschlaggebend für die Gestaltung des pDWH-Schemas sind angenommene, beispielhafte Abfragen, die es auf Basis des pDWH-Systems zu beantworten gilt. Neben Dimensionen zur Analyse des Workflow-Ablaufs wird auch die organisatorische Einbettung des Prozesses in die Unternehmung berücksichtigt. Bei Grigoria et al. (2004) wird ein pDWH-Schema als Teil einer *Business Process Intelligence Toolsuite* vorgestellt. Grundlage bildet die Workflow-Implementierung des *HP Process Managers*. Die vorgeschlagenen Dimensionen werden lediglich aus Außensicht betrachtet. Die Dimensionsstrukturen werden nicht vorgestellt. Um auch Fachanwendern den Zugang zu Workflow-Daten zu ermöglichen, werden die semantischen Dimensionen Prozessregion, Taxonomie und Verhalten eingeführt. Dimensionen und Kennzahlen werden nicht begründet. Kueng et al. (2001) stellen im Rahmen eines DWH-Konzeptes zur Performanceanalyse von Prozessen auch den Ausschnitt eines multidimensionalen Datenmodells ausgerichtet auf Dienstleistungs- und Vertriebsprozesse eines Beispielunternehmens vor. Ablaufdaten werden dabei mit operativen Daten der Anwendungsdomäne kombiniert. Die zugrundeliegende Workflow-Sprache wird nicht expliziert. List et al. (2000) stellen, basierend auf dem Workflow-Verständnis der Workflow Management Coalition (WfMC), ein multidimensionales Datenmodell zur Prozessanalyse vor. Dimensionen und Kennzahlen begründen sich aus den Anforderungen der Prozessanalyse (Instanzanalyse, Prozessverbesserung, ganzheitliche Prozessbetrachtung, organisationale Einbettung). Zur Erfüllung der Anforderungen werden multidimensionale Würfel auf Granularität von Prozessen, Aktivitäten und atomaren Aktionen gebildet. Bei Mansmann et al. (2007a,b) wird ein pDWH-Schema zur Analyse von klinischen Operationsabläufen vorgestellt. Die Grundlage zur Ableitung des Datenmodells bildet ein domänenspezifisches Workflow-Verständnis. Die Ableitung von Kennzahlen und Dimensionen begründet sich zum einen aus der Anwendungsdomäne, zum anderen aus den Perspektiven der Workflow-Modellierung (Funktion, Operation, Verhalten, Information, Organisation) nach Jablonski & Bussler (1996). Bei Pau et al. (2007) wird, mit der übergeordneten Zielsetzung der Evaluierung der Prozessperformanz, ein multidimensionales Datenmodell zur Ablaufanalyse von Workflows vorgestellt. Das Modell sieht separate multidimensionale Würfel zur Analyse von Ereignissen, der Performance von Aufgabenträgern so-

wie der Ablaufperformance des Workflows vor. Die benötigten Datensätze werden über Logging-Mechanismen von *Workflow Management Systemen* abgegriffen. Die Bestimmung der Dimensionen und Kennzahlen basiert nicht auf einem Metamodell bzw. einer konkreten Workflow-Spezifikation. Sun (2014) stellt ein szenario- und prototypingbasiertes Konzept zur Identifikation des Informationsbedarfs für Business-Process-Intelligence-Systeme vor. Der Prototyp als Grundlage zur Evaluierung des Informationsbedarfs basiert auf einem multidimensionalen Modell für Prozessdaten. Dabei handelt es sich zwar um ein umfassendes und generisches pDWH-Schema, von der Modellierungssprache wird jedoch abstrahiert. Das pDWH-Schema ist auf die Analysephase ausgerichtet und für diese ausreichend. Während der Laufzeit des operativen und des pDWH-Systems sind jedoch sprachspezifische Konstrukte notwendig, um Datenquellen zu spezifizieren und um detaillierte Analysen auch auf technischer Ebene zu ermöglichen. Bei zur Muehlen (2001) sowie erweitert bei zur Mühlen (2004, S. 175ff.) werden textuell sowie anhand eines beispielhaften Schemas (zeitorientierte Analysen auf Aktivitätsebene) Bestandteile prozessorientierter DWH-Strukturen erläutert. Als Grundlage der Beschreibung dient eine Referenzstruktur zur Verwaltung von Workflow-Ablaufdaten. Die Referenzdatenstruktur ist auf der Grundlage des WfMC *Workflow Audit Data Format* sowie der Datenstrukturen dreier *Workflow Management Systeme* definiert. Je nach Granularität bzw. Hierarchiestufe (Aktivität, Prozesssegment, Prozess) sowie Kennzahlentyp (Häufigkeiten, zeitbezogene Größen) wird die Bildung eigener *Data-Mart*-Strukturen empfohlen.

Charakterisierung verwandter Arbeiten anhand der unterstützten Prozessperspektiven

Zielsetzung des Abschnittes ist u.a. die Bewertung der Mächtigkeit von verwandten pDWH-Schemata in Bezug auf aktuelle Workflow-Sprachen. Hierzu werden nachfolgend zunächst die explizit unterstützten Dimensionen der vorgestellten verwandten Arbeiten extrahiert und den eingeführten Prozessperspektiven (Abschn. 6.2.2) zugeordnet. In Tab. 6.1 wird das Ergebnis zusammenfassend dargestellt (von variierenden Bezeichnungen und Dimensionsstrukturen wird abstrahiert). Bei der Bewertung wird differenziert, ob das jeweilige Datenmodell eine Dimension voll, nur in Teilen oder spezifisch für eine Anwendungsdomäne unterstützt. Die Semantik der Dimensionen sowie die Zuordnung wird zudem stichpunktartig eingeführt.

	Becker (2007)	Casati et al. (2007)	Eder et al. (2002)	Grigoria et al. (2004)	Kueng et al. (2001)	List et al. (2000)	Mansmann et al. (2007a)	Pau et al. (2007)	Sun (2014)	zur Mühlen (2004)
Prozessperspektive Funktion										
Prozesszuordnung	X	X	X	X	X	X	X	X	X	X
hierarchische Strukturen	X					X	X	X	X	X
GP-Semantik	X	D		X	D		D			
Prozessperspektive Verhalten										
Sequenz	X	X					X		X	
Prozess- / Aktivitätszustand	X	X		X		X	X	X	X	X
Zeit	X	X	X	X	X	X	X	X	X	X
Prozessperspektive Organisation										
Aufgabenträger	X	X	X	X	X	X	X	X	X	X
Organisation	X		X	X	X	X		X	X	X
Prozessperspektive Information										
Prozessdaten	X	D		X	D		D	X	X	X
Ereignisse	X						X	X		(X)

X - unterstützte Dimension, (X) - teilw. unterstützte Dimension, D - domänenspez. Lösung

Tab. 6.1: Berücksichtigte Dimensionen und Prozessperspektiven in verwandten Arbeiten

Dimensionen der Prozessperspektive *Funktion*:

- Die *Prozesszuordnung* beschreibt die einfache Zuordnung von Prozessbestandteilen (z.B. Entscheidungsknoten oder Prozessschritte) zu einem übergeordneten Prozess.
- Die *hierarchische Struktur* beschreibt mehrstufige Eltern-Kind-Beziehungen von Prozessen. Beispiele sind Sub-Prozesse, die aus Innensicht als Prozess spezifiziert sind, aus Außensicht aber selbst als Schritt in übergeordnete Prozesse eingebunden sind.
- Durch die *GP-Semantik* werden Prozessinformationen mit fachlicher Semantik der betriebswirtschaftlichen Anwendungsdomäne erweitert. So wird neben Entwicklern und Modellierern auch Fachanwendern der Zugang zur Prozessanalyse ermöglicht wird.

Dimensionen der Prozessperspektive *Verhalten*:

- Über die Dimension *Sequenz* wird die Reihenfolge von Prozessschritten, im Wesentlichen Vorgänger-Nachfolger-Beziehungen, abgebildet.
- Der *Prozess-* bzw. *Aktivitätszustand* spezifiziert den jeweiligen Ausführungszustand. Gestartet, laufend, abgeschlossen (erfolgreich, nicht erfolgreich) stellen Beispiele dar.
- Die Dimension *Zeit* ermöglicht Analysen im Zeitverlauf. Die Zeit kann dabei in unterschiedlichen Rollen (z.B. als Start- oder Abschlusszeit) Berücksichtigung finden.

Dimensionen der Prozessperspektive *Organisation*:

- Über die Dimension *Aufgabenträger* werden die für die Ausführung von Prozessen bzw. einzelnen Prozessschritten zuständigen Ressourcen im multidimensionalen Datenmodell aufgenommen. Unterschieden werden personelle und maschinelle Aufgabenträger (AwS).
- Die Dimension *Organisation* beschreibt die organisatorische Einbettung von Prozessen bzw. Prozessschritten. Häufig erfolgt die Beschreibung der organisatorischen Verantwortlichkeiten kombiniert mit den zuständigen Aufgabenträgern.

Dimensionen der Prozessperspektive *Information*:

- Über die Dimension *Prozessdaten* werden Datenobjekte, die im Prozessablauf (als Input/Output von Prozessschritten) bearbeitet werden, in ein pDWH-Schema integriert.
- Die Dimension *Ereignisse* erfasst im Prozessverlauf auftretende fachliche Ereignisse.

Zusammenfassende Würdigung

Anhand von Tab. 6.1 wird ersichtlich, dass die verschiedenen Ansätze zur Ableitung von Datenmodellen unterschiedliche Dimensionen berücksichtigen. Dies gilt sowohl für generische Ansätze sowie für Arbeiten mit einem eingeschränkten, informalen oder auch proprietären Workflow-Begriff. Konsens besteht weitgehend bzgl. der Verwaltung der Daten (Kennzahlen) auf Granularität von Ak-

tivitätsinstanzen – soweit verfügbar und sinnvoll interpretierbar. Die vorliegende Arbeit greift, aufgrund der Anwendungsnähe sowie der gegenwärtigen Verbreitung die BPMN als Sprache zur Workflow-Modellierung auf und leitet aus den Sprachbestandteilen ein pDWH-Schema ab. Verglichen mit der Mächtigkeit der BPMN sind die vorgestellten Arbeiten als eingeschränkt zu bezeichnen und erscheinen damit wenig operabel. Konzepte wie Sequenzflüsse (Verhalten) werden nur selten, die Interaktion von Prozessen über Nachrichtenflüsse (Organisation, Information) gar nicht berücksichtigt. Die hierarchische Strukturierung von Workflows (Funktion) wird ebenfalls nur eingeschränkt betrachtet. Zudem werden eine Workflow-Sprache mit grafischer Notation sowie die Ableitung der Datenstrukturen aus dieser Sprache zur Lokalisierung von Analyseergebnissen im Prozessschema als sinnvoll erachtet (Becker & Chamoni 2008). Zu beiden Aspekten bestehen Lücken im z.T. eingeschränkten sowie proprietären oder informalen Workflow-Verständnis der verwandten Arbeiten.

Gerade mit Blick auf das verhaltensorientierte aber auch bezogen auf die Entscheidungsqualität im strukturorientierten pDWH-Szenario ist die Einordnung der Analyseergebnisse in eine von Fachanwendern verständliche Semantik (Funktion: GP-Semantik) sowie die Berücksichtigung von Geschäftsdaten (Information: Prozessdaten) relevant. Die Berücksichtigung beider Aspekte erfolgt nur in wenigen verwandten Arbeiten. Stellenweise werden lediglich domänenspezifische Lösungen vorgeschlagen. Die Arbeit von Becker (2007) wählt hingegen mit der eEPK eine aufgabenorientierte Sprache mit GP-Semantik zur Ableitung der Datenstrukturen. Durch die gewählte Abstraktionsebene entsteht jedoch mit Blick auf die angebotenen Modellbausteine, deren Semantik sowie die Ausführung der Prozesse auf Aufgabenträgerebene eine größere semantische Lücke. So wird beispielsweise bei Thomas et al. (2007) die EPK als Sprache zur fachlichen Prozessgestaltung, die BPMN als anwendungsnahe Sprache mit Brückenfunktion zur Prozessausführung eingeordnet. Durch Verzicht auf diese Brückenfunktion ist eine wechselnde Betrachtung der Prozessabläufe aus *Aufgaben-* und *Aufgabenträger-Perspektive* nicht möglich und führt zu eingeschränkten Möglichkeiten der Prozessanalyse.

Die Anforderung der *flexiblen Anpassbarkeit* des Datenschemas an sich verändernde Prozessstrukturen bzw. an veränderte operative AwS wird im Wesentli-

chen in generischen Ansätzen (z.B. Casati et al. 2007) thematisiert. Ein generisches Datenmodell kann zum einen als unabhängig gegenüber der Prozesssprache und der Anwendungsdomäne, zum anderen aber auch als weitgehend stabil im Falle von Anpassungen der Quellstrukturen charakterisiert werden.

6.4 Gestaltung eines multidimensionalen Prozessdatenschemas

Der Argumentation vorangegangener Abschnitte folgend wird ein multidimensionales Datenmodell für entscheidungsrelevante Prozessdaten vorgestellt. Hierzu wird zunächst eine generische Datenstruktur erarbeitet. Generisch bedeutet dabei, multidimensionale Strukturen unabhängig vom fachlichen Kontext zu spezifizieren. Damit kann das Modell nicht nur in verschiedenen Anwendungsdomänen eingesetzt werden, es kann auch flexibel auf Anpassungen des Workflowschemas reagiert werden. In Kap. 5.1 wird die flexible Anpassbarkeit der Architektur explizit gefordert und soll sich auch im Datenschema widerspiegeln. In Abschn. 6.4.1 werden prozessspezifische Dimensionen auf Basis der BPMN erarbeitet. Die Spezifikation von Kennzahlen erfolgt in Abschn. 6.4.2. Abschn. 6.4.3 zeigt die Anwendung des Datenmodells auf die Fallstudie *Mobility Provision*. In Abschn. 6.4.4 wird die Unterstützung der vollen *Process Modeling Conformance* diskutiert. Die Erweiterung des Datenmodells um Kennzahlen und Dimensionen der betrieblichen Anwendungsdomäne sowie die Integration des Datenmodells in die Referenzarchitektur aus Kap. 5 wird schließlich im nachfolgenden Abschn. 6.5 beschrieben. Den Ausgangspunkt zur Ableitung der multidimensionalen Datenstruktur zur Prozessanalyse bildet das Metamodell der BPMN (Abschn. 6.1.3). Betrachtet wird ein multidimensionaler Würfel, der genau einen Prozesstyp (*private*) zum Gegenstand hat. In einem Kollaborationsdiagramm kann dieser Prozess selbst mit weiteren Teilnehmern/Prozessen interagieren. Die weiteren Teilnehmer werden als *Blackbox* betrachtet. Die Prozesse der Interaktionspartner sind in eigenen multidimensionalen Würfeln aufbereitet. In Abb. 6.4 ist das BPMN-Schema zum Vertriebskanal in einem Kollaborationsdiagramm dargestellt. Die Steuerung des Vertriebskanals ist dabei als *private process* beschrieben. Partnerprozesse sind jeweils als *Black Box* abgebildet. Im oberen Teil der Abbildung ist der vollständige Prozess dargestellt, der untere Teil zeigt beispielhaft den erweiterten Sub-Prozess *Empfang Verfügbarkeitsliste*.

6.4.1 Ableitung prozessorientierter Dimensionen auf Basis der BPMN

Für die generischen multidimensionalen Datenstrukturen werden Dimensionen aus dem Metamodell der BPMN (vgl. Abschn. 6.1.3) abgeleitet und mit dem SDWM nach Böhnlein (2001) (vgl. Kap. 3.4) modelliert. In Abschn. 6.2.2 wird die Anforderung aufgestellt, Daten auf feinsten Granularität zu erfassen, wie sie vom Quellsystem bereitgestellt werden. In BPMN-Prozessen stellen Knoten (*Task*, *Gateway*, *Event*) die atomare Einheit dar und werden als solche in das multidimensionale Datenmodell übernommen. Durch systematische Analyse des BPMN-Metamodells können die nachfolgend vorgestellten Dimensionen ausgehend vom Knoten (Metaobjekt *FlowNode*) abgeleitet werden.

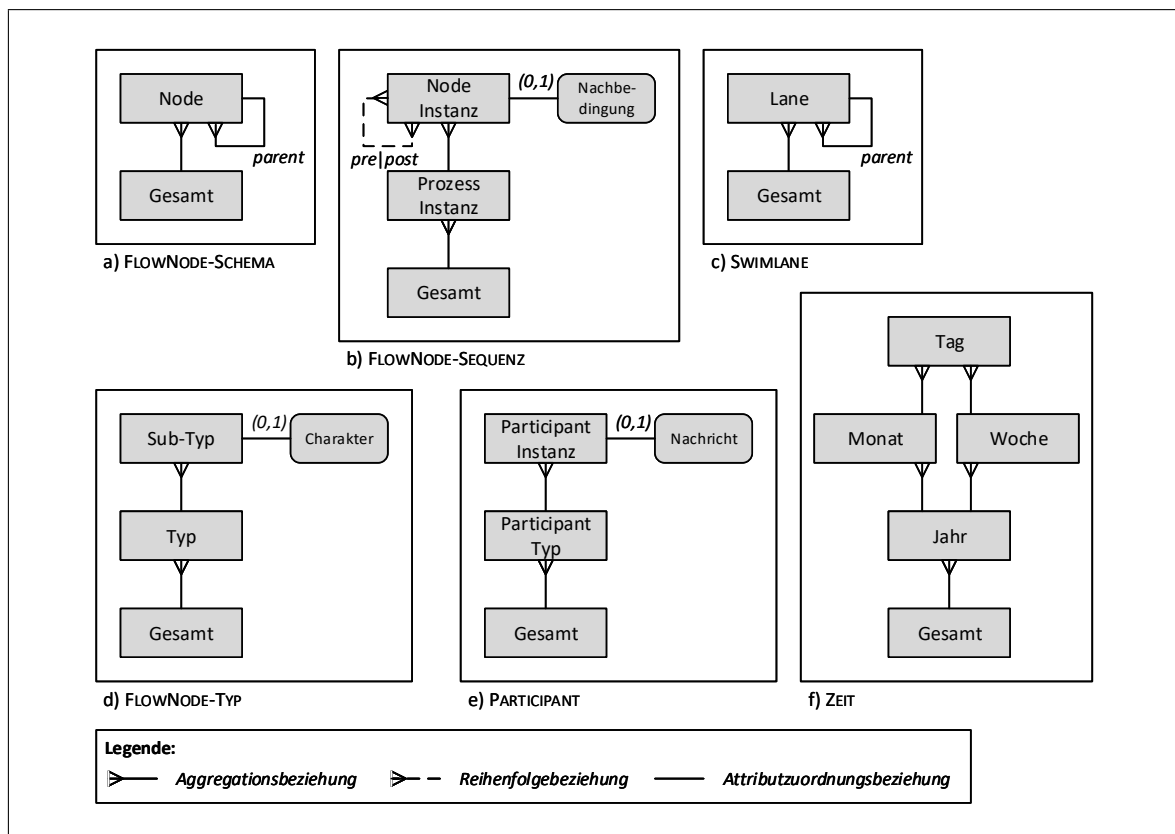


Abb. 6.5: Dimensionen für BPMN-Workflows

Dimension FlowNode-Schema Die Dimension *FlowNode-Schema* beschreibt die Knoten auf Schemaebene. In Abb. 6.5a ist die Dimension abgebildet. Ein Knoten ist auf Schemaebene durch einen semantischen Bezeichner gekennzeichnet. Nach Böhnlein (2001, S. 366ff.) sind einer Dimensionshierarchiestufe implizit eine eindeutige Kennung sowie ein semantischer Bezeichner (z.B. Aktivität *Emp-*

fang Anfrage) zugeordnet. Die Beziehungen zwischen den Hierarchiestufen der Dimension ergeben sich aus den möglichen hierarchischen Strukturen innerhalb eines Workflows. Daraus resultiert eine rekursive Dimension (vgl. Kap. 3.3.1). Es wird damit eine beliebig tiefe Schachtelung des Prozesses in Form von *Sub-Prozessen* oder *Call Activities* ermöglicht (Aggregationsbeziehungen im Metamodell der BPMN). Jeder Knoten ist dabei eindeutig einem übergeordneten Knoten zugeordnet (*Parent-Beziehung*).

Dimension FlowNode-Sequenz Der Prozessablauf wird in Form von Sequenzflüssen im Workflowschema modelliert, kann jedoch je Prozessinstanz variieren. Der konkrete Ablauf einer Instanz wird über die Dimension *FlowNode-Sequenz* in das multidimensionale Datenmodell aufgenommen (Abb. 6.5b). Eine Knoteninstanz kann dabei sowohl mehrere Vorgänger (*Pre-Beziehung*) als auch Nachfolger (*Post-Beziehung*) haben. Zwischen diesen Knoteninstanzen besteht jedoch keine hierarchische, sondern eine sequentielle Beziehung. Innerhalb der Dimensionshierarchie wird deshalb in Anlehnung an Totok (2000, S. 124) keine Aggregationsbeziehung über mehrere Verdichtungsstufen, sondern eine Reihenfolge-Beziehung auf Ebene der *Node-Instanz* definiert. Das Metamodell des SDWM ist hinsichtlich dieser Beziehungsart zu erweitern. Eine Knoteninstanz (z.B. ein *Gateway*) kann (*0,1-Beziehung*) zudem mit einer Nachbedingung näher beschrieben werden. Entlang des Aggregationspfades der Dimension können Instanzen eines Knotens eindeutig einer Prozessinstanz zugeordnet und auf dieser Ebene verdichtet werden. Über die Hierarchiestufe *Gesamt* kann von instanzspezifischen hin zu instanzübergreifenden Prozessanalysen verdichtet werden.

Dimension FlowNode-Typ Die Dimension *FlowNode-Typ* beschreibt generisch die Typisierung von Knoten. Die Struktur der Dimension ist in Abb. 6.5d dargestellt und beinhaltet die Hierarchiestufen *Sub-Typ*, *Typ* und *Gesamt*. Typen auf Instanzebene der pDWH-Strukturen ergeben sich durch Auflösung der Vererbungsbeziehungen (*is-a-Semantik*) im Metamodell, ausgehend vom Metaobjekt *FlowNode*. Typausprägungen sind somit *Task*, *Sub-Process*, *Gateway* und *Event*. Auf der Hierarchiestufe *Sub-Typ* werden die Typen präzisiert. Diese sind im Metamodell nicht explizit dargestellt, können jedoch Abb. 6.3 (graphische Darstellung der Metaobjekte) entnommen werden: *Task* (*UserTask*, *ServiceTask*, *SendTask*, *ReceiveTask*); *Sub-Process* (*embedded Sub-Process*, *CallActivity*); *Gateway* (*Exclusive*, *Parallel*); *Event*

(*StartEvent*, *EndEvent*). Jeder Sub-Typ kann (*0,1-Beziehung*) durch ein beschreibendes Attribut weiter charakterisiert werden. Über das Attribut wird beispielsweise die weitere Unterscheidung von Ereignissen vorgenommen. Für das *StartEvent* werden die Ausprägungen *Message* und *Timer*, für das *EndEvent* *Message* und *Terminate* unterschieden.

Dimension Participant Die Interaktion zwischen Prozessen (Nachrichtenfluss, *MessageFlow* im Metamodell) wird in der Dimension *Participant* (Abb. 6.5e) beschrieben. Die Interaktion erfolgt mit einer Instanz eines *Participants* und ist über die ausgetauschte Nachricht in Form eines Attributs beschrieben. Die Aggregation entlang der Dimension erfolgt über die Hierarchiestufe *Typ* hin zu *Gesamt*. In einem BPMN-Workflow kann ein *Participant* als konkrete Entität oder abstrakte Partnerrolle modelliert werden (OMG 2011, S. 112). In Abhängigkeit davon ist eine eins-zu-eins-Aggregation (Entität) oder viele-zu-eins-Aggregation (Rolle) von Instanz zu Typ des *Participants* möglich.

Dimension SwimLane Die Dimension *SwimLane* (Abb. 6.5c) leitet sich aus dem Metaobjekt *LaneSet* ab. Sie beschreibt die organisatorische Strukturierung eines Teilnehmers durch *SwimLanes*. So wird die Zuordnung eines Knotens zu einer Lane ermöglicht. Die Dimension zeigt aufgrund einer möglichen mehrstufigen Strukturierung ebenfalls eine rekursive Struktur.

Dimension Zeit In Kap. 3.1 werden die stetige Erweiterung des Datenbestandes (*non-volatile*) eines DWH-Systems sowie der Zeitbezug der einzelnen Datensätze (*time-variant*) beschrieben. Dieser Zeitbezug spiegelt sich in der Dimension *Zeit* wieder. Die Dimension leitet sich zwar nicht aus dem Metamodell der BPMN ab, ist jedoch obligatorisch für DWH-Strukturen. In Abb. 6.5f ist eine beispielhafte Struktur dazu dargestellt. Gezeigt sind dabei alternative Aggregationspfade von *Tag* zu *Monat* und *Jahr* bzw. von *Tag* zu *Kalenderwoche* und *Jahr*. Aufbau und Granularität können dabei je nach Anwendungsdomäne variieren. Durch die Dimension *Zeit* werden Analysen der Prozessdaten im Zeitverlauf ermöglicht.

Das pDWH-Schema ist optional um weitere prozessspezifische Dimensionen erweiterbar. Diese begründen sich aus dem jeweiligen Anwendungskontext bzw. der Zielsetzung des pDWH-Systems und können nicht generisch festgelegt wer-

den. Denkbar wären beispielsweise die Berücksichtigung des Ausführungszustandes (z.B. *gestartet*, *wartend*, *beendet*) (z.B. Pau et al. 2007, Casati et al. 2007) oder der Versionshistorie von Aktivitäten und Prozessen.

6.4.2 Spezifikation von prozessorientierten Kennzahlen

Anhand des Metamodells der BPMN können zwar Dimensionen, jedoch keine Kennzahlen abgeleitet werden. Die Spezifikation generischer Kennzahlen für das strukturorientierte Szenario erfolgt auf Basis der verfolgten Zielsetzung der Prozessgestaltung. Zur Bewertung der Prozessgestaltung auf Basis eines pDWH-Systems werden die Ziele *kurze Durchlaufzeiten*, *niedrige Prozesskosten*, *hohe Prozessqualität* sowie *Flexibilität* herangezogen (vgl. Abschn. 6.2.2). Diese können auf Basis der modellierten Prozesse bewertet werden (Fischermanns 2013, S. 309). Aus diesen Zielen begründen sich die nachfolgenden Kennzahlen:

- **Zeit (t):** Die Kennzahl *Zeit* leitet sich aus dem Ziel möglichst kurzer Durchlaufzeiten ab. Die Zeit wird auf Granularität der Prozessknoten definiert. Insbesondere die Dauer von Aktivitäten wird als maßgebend für die Durchlaufzeit von (Sub-)Prozessen bewertet.
- **Kosten (c):** Zur Bewertung von Prozesskosten zählen u.a. auch Kosten für einzelne Prozessschritte (Knoten). Als Beispiel sind Kosten für die Ressourcennutzung zu nennen. Die Kennzahl *Kosten* sollte ebenfalls auf Granularität der Prozessknoten erfasst werden.
- **Anzahl (n):** Nach Fischermanns (2013, S. 310) ist die Prozessqualität über die *Beherrschbarkeit* und *Fehlerfreiheit* des Prozessablaufs definiert. Zur Bewertung der Beherrschbarkeit werden die Kenngrößen *Symbolhäufigkeit* (Fischermanns 2013, S. 313f.) und *Variantenzahl* (Fischermanns 2013, S. 315ff.) vorgeschlagen. Bei Overhage et al. (2012) werden zudem umfassend Qualitätsmerkmale, -metriken und -messverfahren für Geschäftsprozessmodelle vorgestellt. Alle Metriken werden, wie auch die syntaktische Fehlerfreiheit (Fischermanns 2013, S. 328ff.), während der Entwicklungszeit auf Grundlage des Prozessschemas ermittelt. Ein pDWH-System ist hierzu nicht notwendig. Zur Bewertung der Fehlerfreiheit kann das Verhalten von Prozessinstanzen jedoch ergänzend beitragen. Als Beispiel ist das tatsächliche Auftreten von Normal- und Ausnahmeverhalten zu nennen. Das

tatsächliche Verhalten von Prozessinstanzen ist Gegenstand des pDWH-Systems. Jeder durchlaufene Knoten stellt im multidimensionalen Datenmodell auf Instanzebene eine Kombination der zugehörigen Dimensionen dar. Als Kennzahl wird *Anzahl* vorgeschlagen. Diese kann als *Factless Fact* nach Kimball & Ross (2002, S. 246ff.) realisiert werden. Die Kennzahl *n* dient ebenso zur Bewertung der Flexibilität eines Prozesses. Bei Dumas et al. (2013, S. 215f.) wird die Flexibilität eines Prozesses während der Ausführung (run-time) und zur Gestaltungszeit (build-time), also zur Anpassung des Prozessschemas, unterschieden. Bei Wagner et al. (2011b) finden sich hierzu die Begriffe der *Struktur-* und *Verhaltensflexibilität*. V.a. die Flexibilität im Verhalten eines Geschäftsprozesses (z.B. Vergleich zwischen Ist- und Soll-Flexibilität) kann durch Aufzeichnung und Analyse der tatsächlichen Pfade durch den Prozess bewertet werden. Die Strukturflexibilität ist über Ablaufdaten in einem pDWH-System nicht hinreichend bewertbar.

Vergleichbare generische Kennzahlen werden auch bei Becker (2007, S. 269f.) vorgeschlagen. Diese Kennzahlen können weiter als *Prozessablaufkennzahlen* (auf Aktivitäts- oder Prozessebene) und *Prozessergebniskennzahlen* differenziert werden (Becker 2007, S. S. 269f.). Bei allen Kennzahlen handelt es sich um *Basiskennzahlen* der prozessorientierten, multidimensionalen Datenstruktur, die im jeweiligen Anwendungsszenario zur Ableitung spezifischer Kennzahlen genutzt werden können. Für das *verhaltensorientierte* pDWH-Szenario sind Basiskennzahlen aus der jeweiligen betrieblichen Domäne abzuleiten und zu ergänzen. Zu beachten ist, dass Kennzahlen mit heterogener Granularität vorliegen können. Manche Kennzahlen liegen für Knoten, andere hingegen nur für (Sub-)Prozesse vor (Casati et al. 2007).

6.4.3 Anwendung des multidimensionalen Prozessmodells

Zur Verdeutlichung des multidimensionalen pDWH-Schemas erfolgt die Anwendung auf das Fallbeispiel *Mobility Provision*, konkret auf den Ausschnitt des Vertriebsprozesses aus Abb. 6.4. Das resultierende Datenschema ist in Tabellenstrukturen, angelehnt an das Konzept des *Star-Schemas* (vgl. Kap. 3.3.3), in Abb. 6.6 dargestellt. Die zugehörige SQL-basierte Implementierung der Tabellen findet sich in Anhang A.1.

FlowNode-Schema			FlowNode-Sequenz				
NODE-ID	NODE-DESC	PARENT	N-INSTANZ	PRE	POST	P.COND.	P-INSTANZ
[...]	[...]	[...]	[...]	[...]	[...]	[...]	[...]
2	Senden Verfügbarkeitsprüfung	-	2-1	[...]	3-1	-	1
3	Empfang Verfügbarkeitsliste	-	3-1	2-1	4-1	-	1
4	Ergebnis Empf. Verfügbarkeitsliste	-	4-1	3-1	[...]	[...]	1
301	Ereignis Empfang Liste e-Car	3	301-1	-	302-1	-	1
302	Empfang Liste e-Car	3	302-1	301-1	303-1	-	1
303	e-Cars verfügbar?	3	303-1	302-1	304-1	-	1
304	Ergebnis e-Cars verfügbar?	3	304-1	303-1	305-1	true	1
305	Verfügbarkeit speichern	3	305-1	304-1	[...]	[...]	1
306	Prüfung Suchkriterien	3	[...]	[...]	[...]	[...]	[...]
[...]	[...]	[...]	[...]	[...]	[...]	[...]	[...]

FlowNode-Typ				Participant			
TYPE-ID	SUBTYP	CHARAKTER	TYP	P-ID	P-INSTANZ	P-MESSAGE	P-TYP
T1	start_event	message	event	P1	Flotte-AwS	Liste eCars	Flotte-AwS
T2	embedded_sub_process	-	sub_process	P2	e-Car Daten	Liste Verfügbarkeit	e-Car Daten
T3	exclusive_gateway	-	gateway	[...]	[...]	[...]	[...]
T4	receive_task	-	task				
T5	service_task	-	task				
T6	send_task	-	task				
[...]	[...]	[...]	[...]				

Fakten					
NODE-ID	N-INSTANZ	TYPE-ID	P-ID	timestamp	n
[...]	[...]	[...]	[...]	[...]	1
2	2-1	T2	-	2015-12-13T13:15:20	1
3	3-1	T2	-	2015-12-13T13:15:50	1
4	4-1	T3	-	2015-12-13T13:18:45	1
301	301-1	T1	P1	2015-12-13T13:15:50	1
302	302-1	T4	-	2015-12-13T13:15:52	1
303	303-1	T5	-	2015-12-13T13:16:40	1
304	304-1	T3	-	2015-12-13T13:17:50	1
305	305-1	T6	P2	2015-12-13T13:17:57	1
[...]	[...]	[...]	[...]	[...]	1

Abb. 6.6: Exemplarische Instanzen des pDWH-Schemas zum Vertriebsprozess

Die Tabelle *Fakten* stellt ihre Instanzen als Kombination aus den Dimensionen *FlowNode-Schema* (referenziert über die *Node-ID*), *FlowNode-Sequenz* (*N-Instanz*), *FlowNode-Typ* (*Type-ID*) und, sofern relevant, *Participant* (*P-ID*) dar. Zudem wird die Kennzahl *Anzahl* (*n*) abgebildet. Auf eine explizite Darstellung der Dimension *Zeit* wird verzichtet. Durch einen Zeitstempel (*timestamp*) kann der Bezug für Analysen im Zeitverlauf hergestellt werden.

Knoteninstanzen werden über eine Kombination aus *Node-ID* und einem globalen, prozessinstanzübergreifenden Zähler identifiziert (z.B. Instanz 301-1). Die Zuordnung einer Knoten- zu einer Prozessinstanz erfolgt in der Dimensionstabelle *FlowNode-Sequenz* durch das Attribut *P-Instanz*. Knoteninstanz 301-1 aus Prozessinstanz 1 wird beispielhaft aufgegriffen. Die Einbettung der Knoteninstanz in den Prozessablauf wird ebenfalls in Tabelle *FlowNode-Sequenz* deutlich. Als Startereignis ist für den Knoten 301-1 lediglich ein Nachfolger (302-1) hinterlegt. In der Tabelle *FlowNode-Typ* ist ersichtlich, dass es sich bei Instanz 301-1 um ein Ereignis, konkret ein *Start-Event* zum Nachrichtenempfang (Charakterisierung

Message), handelt. Aus der Tabelle *Participant* ist ersichtlich, dass es sich bei der Nachricht um eine Liste von e-Cars, übermittelt vom Teilnehmer *Flotte-AwS*, handelt. Die Knoteninstanz 301-1 ist durch das Schemaobjekt 301 mit dem Bezeichner *Ereignis Empfang Liste e-Car* definiert. Zudem ist die hierarchische Beziehung zu Knotentyp 3 (*Empfang Verfügbarkeitsliste*) in der Dimension *FlowNode-Schema* dargestellt. Knoten 3 stellt einen eingebetteten Sub-Prozess auf höchster Hierarchieebene des Workflows dar. Als Beispiel eines Gateways dient Knoteninstanz 304-1. Die eingetretene Nachbedingung ist in der Tabelle *FlowNode-Sequenz* mit *true* hinterlegt. Das bedeutet, dass e-Cars verfügbar sind und im nächsten Schritt (Knoteninstanz 305-1) gespeichert werden können.

Nachfolgend werden anhand des Beispiels weitere ausgewählte Modellsituationen erläutert:

Alternative und parallele Pfade Alternative und parallele Abläufe sind gängiger Bestandteil von Prozessmodellen. Alternative Abläufe entstehen durch *Exclusive-Gateways*. Die Abbildung in der multidimensionalen Datenstruktur kann anhand von Knoteninstanz 304-1 gezeigt werden. Im Schema werden zwar mehrere Nachfolger (305, 306) für Schemaobjekt 304 modelliert, auf Instanzebene wird jedoch genau ein Pfad/Nachfolger (hier: 305-1) gewählt. Durch ein paralleles *Gateway* werden parallele Pfade gestartet bzw. zusammengeführt. Hieraus begründet sich eine n:m-Beziehung eines Knotens zu Vorgängern und Nachfolgern. Bei relationaler Realisierung wie in Abb. 6.6 wird eine zusätzliche Tabelle notwendig, die die n:m-Beziehung abbildet und die Tabelle *FlowNode-Sequenz* in den Rollen *pre* und *post* referenziert.

Schleifen Die Modellierung von Schleifen im Workflow führt zum mehrfachen Durchlaufen eines Knotens, also mehreren Instanzen des Knotens. Der Sub-Prozess *Empfang Verfügbarkeitsliste* (Abb. 6.4) sowie die zugeordneten Elemente (z.B. Knoten 301) sind in einer Schleife eingebettet und werden möglicherweise pro Prozessinstanz mehrfach durchlaufen. Die Kombination aus *Node-ID* und Zähler (Syntax: *NodeID-Zähler*) als *N-Instanz* dienen der Identifikation. Je Prozessinstanz kann die Anzahl als auch die Reihenfolge (z.B. durch Höhe des Zählers oder dessen Zeitbezug) des Durchlaufs eines Schemaobjektes ermittelt werden.

6.4.4 Unterstützung der vollen BPMN-Konformität zur Prozessmodellierung

Im vorangehenden Abschn. 6.4.1 erfolgt die Ableitung des pDWH-Schemas auf Basis eines Ausschnitts der BPMN, der sich durch die *Descriptive Process Modeling Conformance* der BPMN-Spezifikation abgrenzt. Durch die volle *Process Modeling Conformance* wird die Palette von Modellelementen zur Spezifikation von Workflows erweitert. Nachfolgend wird erörtert, ob und wie sich diese Erweiterungen auf das vorgestellte pDWH-Schema auswirken.

Zum einen kommen zusätzliche Sub-Typen von *FlowNodes* durch die Erweiterung hinzu. Zur Abbildung dieser ist keine Anpassung des pDWH-Schemas notwendig. In der vorgestellten Datenstruktur können diese als Instanzen der Dimensionsstufe *Sub-Typ* der Dimension *FlowNode-Typ* verwaltet werden. Hinzu kommen die Sub-Typen für *Task* (*Manual*, *Business Rule*, *Script*), für *Event* (*intermediate*) sowie für *Gateway* (*Complex*, *Event-based*). Die detaillierte Charakterisierung der Sub-Typen erfolgt über das entsprechende Dimensionsattribut. Beispielsweise kann so die Differenzierung eines *event-based Gateways* in *parallel* und *exclusive* erfolgen. Für Ereignisse kann das Dimensionsattribut *Charakter* Aufschluss über die Art des Ereignisses geben (z.B. *Message Sub-Process Interrupting*).

Zum anderen kommen durch die Erweiterung sog. *Marker* hinzu, die die nähere Spezifikation der Ausführungssemantik von Aktivitäten ermöglichen. Einige Marker sind sowohl für *Tasks* als auch für *Sub-Processes* (*Loop*, *Multiple Instance*, *Compensation*), andere nur auf das Element *Sub-Process* (*Ad-hoc*) anwendbar. Die Marker werden in der Dimension *FlowNode-Typ* als Charakterisierung des *embedded Sub-Process* bzw. des jeweiligen Sub-Typs von *Task* geführt. Die Abbildung der Marker im pDWH-Schema wird nachfolgend beschrieben:

- **Loop:** Die Modellierung und Abbildung von Schleifen wird bereits in Abschn. 6.4.3 anhand des Fallbeispiels eingeführt. Alternativ zur dort beschriebenen expliziten Modellierung der Schleifenlogik kann eine Aktivität (*Sub-Process*, *Task*) als Schleife gekennzeichnet und mit einer Abbruchbedingung versehen werden (OMG 2011, S. S. 190f.). Die Abbildung von *Loop*-Aktivitäten im pDWH-Schema erfolgt analog zur explizit modellierten Schleifenlogik. Je Durchlauf einer Aktivität wird eine Aktivitätsinstanz (Syntax:

NodeID-Zähler) erzeugt und der *Zähler* inkrementiert. Im Falle von Sub-Prozessen mit *Loop*-Marker gilt dies für alle Kind-Elemente.

- **Multiple Instance:** Zur Laufzeit werden für eine Aktivität (*Sub-Process*, *Task*) mit *Multiple Instance*-Marker mehrere Instanzen gebildet. Diese werden je nach Spezifikation parallel oder sequentiell ausgeführt. Die Anzahl der Instanzen kann vorgegeben oder auf Basis zu verarbeitender Datensätze ermittelt werden (OMG 2011, S. 191f.). Pro Aktivitätsinstanz wird, vergleichbar mit der Abbildung von *Loop*-Aktivitäten, eine Knoteninstanz im pDWH-Schema gebildet. Die parallele bzw. sequentielle Ausführung der Aktivitätsinstanzen ist aus den Vorgänger-Nachfolger-Beziehungen ersichtlich. Auch hier ist eine Anpassung der Struktur des pDWH-Schemas nicht notwendig.
- **Compensation:** Zielsetzung von Kompensation ist es, bereits ausgeführte Prozessschritte rückgängig zu machen. Aktivitäten zur Kompensation sind nicht in den Sequenzfluss eines Prozesses eingebunden, sondern sind mit einer Aktivität des regulären Prozessablaufs assoziiert (OMG 2011, S. S. 303ff.). Zur Abbildung dieser Assoziation zwischen Knoten wird ein zusätzliches Attribut (*Assoziation*) für die Tabelle *Fakten* aus Abb. 6.6 des pDWH-Schemas vorgeschlagen. Ein *Compensation*-Sub-Prozess ist explizit als Teil eines übergeordneten Prozesses modelliert (OMG 2011, S. S. 303ff.). Diese Zuordnung wird über die Dimension *FlowNode-Schema* bereits abgedeckt.
- **Ad-hoc:** Ein *Ad-hoc* Sub-Prozess besteht aus Elementen, die keine oder nur teilweise eine Ablaufreihenfolge vorgegeben haben (OMG 2011, S. 180ff.). Der Ablauf wird während der Prozessausführung bestimmt. Im pDWH-Schema wird grundsätzlich der Ablauf einer Prozessinstanz zur Laufzeit in der Dimension *FlowNode-Sequence* verwaltet. Damit kann auch die *ad-hoc* gewählte Sequenz instanzspezifisch aufgezeichnet werden.

Als weitere Spezialisierung von Sub-Prozessen kommen mit der *Process Modeling Conformance* der *Event Sub-Process* sowie der *Transaction Sub-Process* hinzu. Wie auch die *Call Activity* stellen beide Sub-Typen eines *Sub-Process* in der Dimension *FlowNode-Typ* dar. Die jeweilige Abbildung auf die pDWH-Struktur wird nachfolgend beschrieben:

- **Event Sub-Process:** Der *Event Sub-Process* stellt eine Verallgemeinerung der *Compensation* dar. Der Prozess ist einem übergeordneten Prozess zugeord-

net (Dimension *FlowNode-Schema*), jedoch nicht Bestandteil im normalen Ablauf. Ausgelöst wird dieser, wie aus dem Bezeichner ableitbar, über Ereignisse im übergeordneten Prozess. Die Assoziation zwischen den Prozessen bzw. zwischen deren Ereignissen kann analog zur *Compensation* über das Attribut *Assoziation* der Tabelle *Fakten* realisiert werden.

- **Transaction Sub-Process:** Ein Sub-Prozess kann als Transaktion gekennzeichnet werden. Die Transaktion ist in einen Sequenzfluss eingebunden, wobei das Ergebnis der Transaktion den nachfolgenden Ablauf bestimmt. Bei erfolgreicher Ausführung wird mit dem die Transaktion umgebenden Sequenzfluss fortgesetzt. Bei Abbruch der Transaktion ist zum einen die Durchführung assoziierter Kompensationen (s.o.) für Aktivitäten innerhalb der Transaktion möglich. Zum anderen werden mit der Transaktion assoziierte Abbruch-Ereignisse (*throw-catch*, modelliert als *boundary event*) behandelt (OMG 2011, S. 178ff.). Über das bereits eingeführte Attribut *Assoziation* der Faktentabelle kann diese Assoziation der Ereignisse ebenfalls abgebildet werden.

Zusammenfassend wird festgehalten, dass das in den Abschn. 6.4.1 bis 6.4.2 spezifizierte pDWH-Schema mit minimalen Anpassungen auch die Elemente der vollen *Process Modeling Conformance* aufnehmen kann.

6.5 Bereitstellung prozessorientierter Daten durch die Komponentenarchitektur

Die in Abschn. 6.4 beschriebenen Datenstrukturen basieren auf der Aufzeichnung der Workflow-Ausführung. Aufzeichnung sowie Bereitstellung der multidimensionalen Prozessdaten sind Aufgaben der *Monitoring-Komponente* der Referenzarchitektur aus Kap. 5. Sowohl für das verhaltens- als auch das strukturorientierte pDWH-Szenario ist die Erweiterung des pDWH-Schemas um Informationen (Kennzahlen und Dimensionen) der betrieblichen Domäne notwendig. Als Grundlage zu deren Spezifikation dienen die *Datenkomponenten* bzw. die zugeordneten Datenschemata.

Zur Ableitung von Dimensionen wird die Arbeit von Böhnlein (2001, S. 353ff.) herangezogen. Sie beschreibt die Spezifikation multidimensionaler Datensche-

mata basierend auf objektorientierten Strukturen der Datenquelle. Für den Ableitungsprozess von Dimensionen bzw. Dimensionshierarchien ist Expertenwissen über die Anwendungsdomäne erforderlich. Eine algorithmische Beschreibung ist nicht möglich (Böhnlein 2001, S. 355). Dimensionen bzw. Dimensionshierarchien können sich aus abgegrenzten Objekttypen, deren Attributen oder den Beziehungen zwischen Objekttypen ableiten (Böhnlein 2001, S. 355). Nachfolgend ist ein Ausschnitt der Regeln nach Böhnlein (2001, S. 356) angegeben, die zur Identifikation von Dimensionen und Dimensionshierarchien beitragen können:

- (0,*)- bzw. (1,*)-Beziehungen zwischen Objekttypen können Hinweise auf die Dimensionshierarchie darstellen.
- *is-a*- oder *is-part-of*-Beziehungen zwischen Objekttypen können Hinweise auf Dimensionshierarchien darstellen.
- Auf Basis der Attribute der Objekttypen können Kandidaten für Dimensionen bzw. Dimensionshierarchien abgeleitet werden.

Gegenstand der Betrachtung ist jeweils der von einer Datenkomponente verwaltete Ausschnitt des operativen Datenschemas. Die Ableitung von Dimensionen wird nachfolgend anhand zweier Beispiele aus der Fallstudie *Mobility Provision* verdeutlicht. In Abb. 6.7 ist linksseitig das operative Datenschema der Komponente *Vertriebsdaten* (vgl. Komponentenarchitektur Abb. 5.3) als UML-Klassendiagramm dargestellt. Der Typ *Vertriebskanal* stellt den Supertyp für *CallCenter* und *e-Shop* dar. Beide stehen damit in einer *is-a*-Beziehung zu *Vertriebskanal*. *CallCenter* steht in einer *1*-*0..**-Beziehung zu *CC Mitarbeiter*.

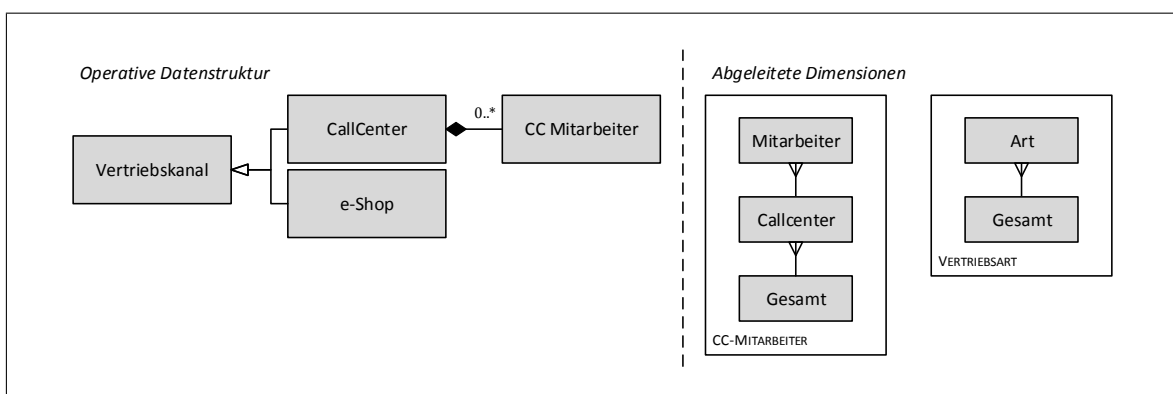


Abb. 6.7: Operative und multidimensionale Datenstrukturen der Komponente *Vertriebsdaten*

Anhand dieser Beziehung wird die Dimension *Vertriebsart* (Abb. 6.7 rechts) abgeleitet. Auf unterster Hierarchiestufe werden die Ausprägungen *CallCenter* und

e-Shop unterschieden. Die Dimension *CC-Mitarbeiter* (Abb. 6.7 rechts) begründet sich aus der Komposition (*is-part-of*-Beziehung) von *CallCenter* zu *CC-Mitarbeiter*. Mitarbeiter bilden die niedrigste Stufe der Dimensionshierarchie. Es wird angenommen, dass in der Fallstudie *Mobility Provision* mehrere Call Center beauftragt sind, zu denen Mitarbeiter eindeutig zugeordnet werden können. Daraus begründet sich die Hierarchiestufe *CallCenter*.

In Abb. 6.8 ist der Objekttyp *Kunde* mit seinen beschreibenden Attributen dargestellt. Die Verwaltung der operativen Daten ist der Komponente *Kundendaten* zugeordnet. Dimensionen bzw. Dimensionshierarchien begründen sich hier anhand kundenspezifischer Attribute. Durch die Attribute *Postleitzahl* und *Ort* wird die geographische Herkunft des Kunden festgelegt. In der Dimension *Geographie* wird die Aggregation hin zu *Bundesland* und *Land* spezifiziert. Hinsichtlich dieser Zuordnung ist die Komponente *Kundendaten* mit zusätzlichen Informationen zu konfigurieren. Das Attribut *Geschlecht* führt zur gleichnamigen Dimension, anhand der die Differenzierung nach *männlich* und *weiblich* ermöglicht wird. Aus dem Attribut *Kundengruppe* wird die Dimension *Kundenstruktur* abgeleitet. Ein *Kunde* wird eindeutig einer *Kundengruppe* zugeordnet. Zudem ist *Geburtsjahr* entscheidungsrelevant und ermöglicht Analysen von Alterstrukturen.

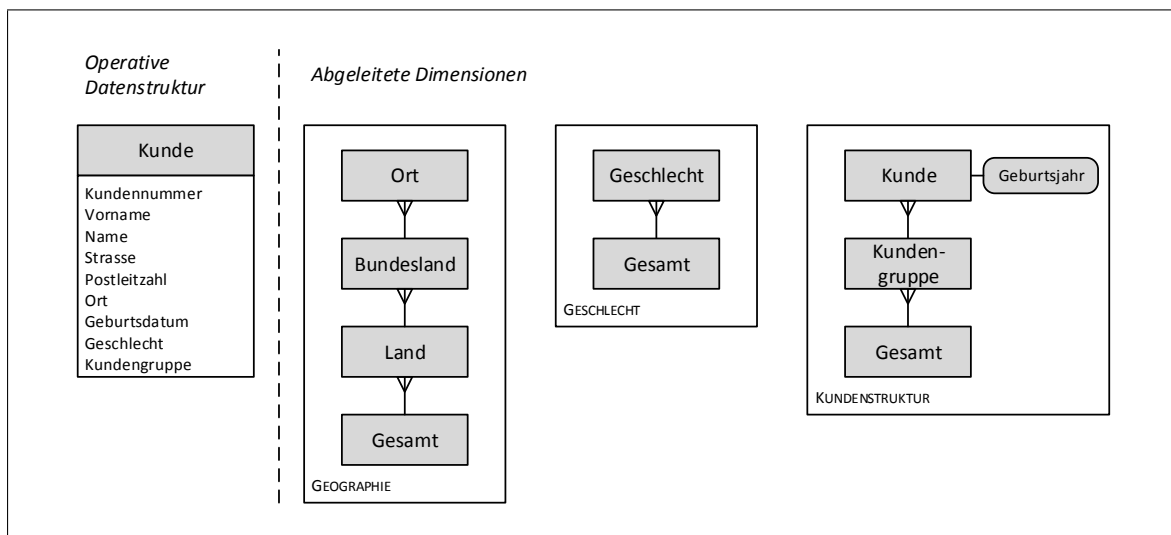


Abb. 6.8: Operative und multidimensionale Datenstrukturen der Komponente *Kundendaten*

Die Spezifikation von Kennzahlen der betrieblichen Domäne erfolgt ebenfalls nach Böhnlein (2001, S. 351ff.). Kennzahlen werden zur Leistungsbewertung eines Geschäftsprozesses spezifiziert. Zunächst werden diese identifiziert und im

nächsten Schritt Objekttypen zugeordnet, die die Transaktion (Leistungserstellung) innerhalb eines Geschäftsprozesses beschreiben (Böhnlein 2001, S. 351f.). Übertragen auf die Referenzarchitektur aus Kap. 5 werden die Kennzahlen den Datenkomponenten zugeordnet, die Transaktionsdaten verwalten. Anhand des Beispiels *Mobility Provision* werden die Komponenten *Buchungsdaten*, *Mietprozessdaten* und *Forderungsdaten* (Abb. 5.3) identifiziert. In Kap. 4.4 wird das Entscheidungsszenario zur *Angebotsweiterung* eingeführt. Die *Erfolgsquote* (Annahmehquote erweiterter Angebote) als Kennzahl je nach Phase des Leihprozesses und Kundenkategorie kann zur Entscheidung, ob ein erweitertes Angebot unterbreitet wird, herangezogen werden. Die notwendigen Basiskennzahlen werden der Komponente *Buchungsdaten* zugeordnet.

Für das Entscheidungsszenario *Prozessgestaltung Vertriebskanal* aus Kap. 4.4 können die generischen Kennzahlen *Kosten*, *Zeit* und *Anzahl* über die *Monitoring*-Komponente verwendet werden. Diese können entlang der vorgestellten Dimensionen aus Abb. 6.7 und Abb. 6.8 sowie des pDWH-Schemas aus Abschn. 6.4 verdichtet und analysiert werden. Beispielsweise können Unterschiede zwischen *Online*- und *CallCenter*-Vertrieb bzgl. Kosten und Zeit bis zum Abschluss der Aktivität *Versenden des individuellen Angebotes* aufgedeckt werden. Ebenso können Kundenpräferenzen für die Art des Vertriebskanals nach Kundenmerkmalen wie *Geschlecht* oder *Altersgruppe* aufgedeckt werden.

Die identifizierten multidimensionalen Strukturen (Kennzahlen und Dimensionen) dienen als Grundlage zur Parametrisierung der *Export*-Schnittstellen zum *OLAP-Export*, wie in Kap. 5.3.4 beschrieben. Die Bereitstellung der Daten erfolgt multidimensional, jedoch nicht integriert. Die Daten werden für die jeweiligen Data Marts und Entscheidungsszenarien (*Vertriebsgestaltung*, *Angebotsweiterung*) durch den Prozess der *Datenintegration* (vgl. 5.4.2) zusammengeführt und übergeben.

6.6 Zusammenfassende Einordnung

In Kap. 6 wird die Bereitstellung von entscheidungsrelevanten Prozessdaten beschrieben. Hierzu wird zunächst ein generisches, multidimensionales Schema für Prozessdaten erstellt und im Weiteren anhand der Referenzarchitektur aus Kap. 5 eingeordnet. Als Ausgangspunkt zur Spezifikation des pDWH-Schemas dient die BPMN als aktuell weit verbreitete und anerkannte Sprache zur Prozessmodellierung. Die Dimensionen des Datenmodells begründen sich dabei aus dem Metamodell. Vergleichbare Ansätze zur Spezifikation von multidimensionalen Prozessmodellen basieren z.T. auf proprietären oder domänenspezifischen Prozessdefinitionen. Diese sind bezogen auf die Modellierungsmächtigkeit der BPMN jedoch eingeschränkt. Das konzipierte Datenmodell wird nachfolgend anhand der Perspektiven der Prozessmodellierung, der pDWH-Szenarien sowie anhand der Referenzarchitektur zusammenfassend eingeordnet.

Einordnung anhand der Perspektiven der Prozessmodellierung In Abschn. 6.3 werden verwandte Forschungsarbeiten anhand der Unterstützung der Perspektiven der Prozessmodellierung nach Curtis et al. (1992) bewertet. Nachfolgend wird das vorgestellte pDWH-Schema ebenfalls anhand dieser bewertet. Zu berücksichtigen ist, dass Sprachen zur Prozessmodellierung in unterschiedlicher Ausprägung eine oder mehrere, i.d.R. jedoch nicht vollständig die Perspektiven der Prozessmodellierung unterstützen (Curtis et al. 1992). Bei Korherr (2007, S. 33ff.) wird die BPMN anhand der eingeführten Perspektiven eingeordnet. Die Bewertung des vorgestellten pDWH-Schemas erfolgt in Teilen auf Basis der Einordnung.

- Perspektive **Funktion**: Die Funktion ist durch die Dimension *FlowNode-Schema* mit hierarchischer Struktur des Prozesses und semantischen Bezeichnern v.a. der *Tasks* beschrieben. Eine Erweiterung der Ablaufinformationen des pDWH-Schemas um Dimensionen der betrieblichen Domäne ist vorgeschlagen. Somit werden auch fachliche Informationen der betrieblichen Domäne (*GP-Semantik*) im multidimensionalen Schema berücksichtigt. Eine Lücke verwandter Arbeiten ist die die Einordnung von Kennzahlen und Dimensionen in die Funktionen der betrieblichen Leistungserstellung bzw. deren Lenkung. Dies bildet u.a. den Untersuchungsgegenstand des nachfolgenden Kap. 7 im Rahmen der Konzeption einer Entwicklungsmethodik.

- Perspektive **Verhalten**: Die Verwaltung des Prozessverhaltens sowie dessen Analyse werden durch die Dimension *FlowNode-Sequenz* ermöglicht. Die Verwaltung der jeweils ausgeführten Sequenzen (bestimmt über Gateways) erfolgt auf Instanzebene. Die Art der ausgeführten Elemente des Sequenzflusses ist durch die Dimension *FlowNode-Typ* beschrieben.
- Perspektive **Organisation**: Organisatorische Verantwortlichkeiten innerhalb des Prozesses sind durch die Strukturierung eines *Pools* in *Lanes* in der BPMN sowie der Dimension *SwimLane* abgedeckt. Durch die Sub-Typen *UserTask* und *ServiceTask* (Dimension *FlowNode-Typ*) kann zwischen automatisierten und nicht-automatisierten Aufgabenträgern unterschieden werden. Über die Dimension *Participant* werden zudem die Einbettung und die Interaktion des Prozesses mit anderen Teilnehmern/Prozessen aus organisatorischer Perspektive beschrieben.
- Perspektive **Information**: Die durch Interaktion mit Partnerprozessen ausgetauschten Nachrichten (Dimension *Participant*) werden der Perspektive *Information* zugeordnet. Während des Prozessablaufs eingetretene Ereignisse stellen ebenfalls relevante Prozessinformationen dar. Erfasst werden diese über den Typ des Ereignisses (Dimension *FlowNode-Typ*) sowie die Semantik in Dimension *FlowNode-Schema*. In Abschn. 6.1.3 wird bereits auf den nicht operablen Datenbegriff der BPMN hingewiesen. Im vorgestellten Konzept finden die verarbeiteten Daten als Dimensionen der betrieblichen Domäne (Abschn. 6.5) Berücksichtigung.

Auch die von verwandten Arbeiten eingeführten Dimensionen finden sich im vorgestellten Modell wieder. Zusammen mit BPMN-spezifischen Dimensionen (z.B. *Participant*) werden diese anhand des Metamodells begründet. Zudem wird ein Vorschlag zur Integration von Dimensionen der betrieblichen Domäne unterbreitet. Wie gezeigt, ergibt sich damit eine umfassende Abdeckung der Perspektiven zur Prozessmodellierung. Die Bereitstellung der Daten erfolgt auf Granularität der Datenquelle. Kennzahlen werden auf der Granularität von Knoteninstanzen (sofern sinnvoll) bereitgestellt. Aufgrund des multidimensionalen Charakters des pDWH-Schemas ist die Möglichkeit zur Aggregation und Abstraktion gegeben.

Einordnung der pDWH-Szenarien Durch das vorgestellte pDWH-Schema werden sowohl das struktur- als auch das verhaltensorientierte pDWH-Szenario unterstützt. Auf Basis der Literatur wird die Anforderung aufgestellt (vgl. Abschn. 6.2.2), in beiden Szenarien den Ablauf von Prozessen analysierbar zu gestalten. Anhand der Hauptziele der Prozessgestaltung werden die Basiskennzahlen *Zeit*, *Kosten* und *Anzahl* für das strukturorientierte Szenario vorgeschlagen, anhand derer weitere Kennzahlen abgeleitet werden können. Dimensionen dazu leiten sich zum einen aus dem Metamodell der BPMN, zum anderen aus den Datenstrukturen operativer Quellsysteme ab. Für das verhaltensorientierte Szenario kann das pDWH-Schema um Kennzahlen und Dimensionen der betrieblichen Anwendungsdomäne erweitert werden, die die Ableitung von Maßnahmen zur aktiven Prozesssteuerung ermöglichen.

Einordnung anhand der Referenzarchitektur Die Bereitstellung von multidimensionalen Prozessdaten wird innerhalb der Referenzarchitektur der *Monitoring*-Komponente zugeordnet. Zur Bereitstellung ergänzender, betriebswirtschaftlicher Daten (Kennzahlen und Dimensionen) werden die operativen Datenkomponenten konfiguriert. Flexibilität zur Anpassung an sich veränderndes Markt- und Kundenverhalten mit sich verändernden Prozessen (Workflows) stellt eine Anforderung an die Referenzarchitektur für operativ-dispositive AwS dar. Hieraus resultiert auch die Anforderung der *flexiblen Anpassbarkeit* der Datenstrukturen des pDWH-Systems als Kern des dispositiven Teilsystems. Im generischen pDWH-Schema kann ohne Anpassung auf Schemaebene auf Veränderungen im Workflowschema reagiert werden. Neue oder veränderte Knoten des operativen Workflowschemas finden sich als neue Instanzen der Dimension *FlowNode-Schema*, Veränderungen im Ablauf der Prozessausführung in den Instanzen der Dimension *FlowNode-Sequenz* wieder. Die Dimension *FlowNode-Typ* führt Kontentypen des BPMN-Metamodells und ist unabhängig von konkreten Anwendungsszenarien. Das pDWH-Schema kann aufgrund seiner generischen Struktur somit als robust bzgl. angepasster Prozesse bezeichnet werden. Aufgrund dieser stabilen Strukturen bleibt die Rekonstruierbarkeit von Ergebnissen und Berichten auch nach Anpassung operativer Prozesse sichergestellt. Das pDWH-Schema stellt eine geeignete Grundlage zur Realisierung der dispositiven Datenstrukturen prozessorientierter operativ-dispositiver AwS dar.

7 Entwicklung geschäftsprozess-orientierter operativ-dispositiver Systeme

Kapitelzusammenfassung: In Kap. 5 wird eine Referenzarchitektur für operativ-dispositive Anwendungssysteme, in Kap. 6 das zugehörige Datenmodell vorgestellt. Nachfolgend wird eine Entwicklungsmethodik konzipiert, die die Anwendbarkeit dieser Artefakte in betrieblichen Szenarien als Zielsetzung verfolgt und damit auch zu deren Validierung beiträgt. Ausgehend von Anforderungen an eine geeignete Entwicklungsmethodik (Abschn. 7.1) sowie der Diskussion verwandter Arbeiten (Abschn. 7.2) wird die Grundkonzeption in Abschn. 7.3 vorgestellt. Die Methodik entkoppelt zeitlich die Entwicklung des operativen AWS (Abschn. 7.4) sowie des angebots- und des nachfrageorientierten DWH-Systems. Einen zentralen Beitrag der Entwicklung des angebotsorientierten DWH-Systems stellt die Einordnung von AT-spezifischen Prozessinformationen in den Geschäftsprozess-Kontext (Abschn. 7.5) dar. Bei der Konzeption bzgl. des nachfrageorientierten Systems wird zudem die Integration analytischer Informationen in operative Prozessabläufe thematisiert (Abschn. 7.6). Den Abschluss des Kapitels bilden Zusammenfassung und Diskussion der Ergebnisse in Abschn. 7.7. Grundlage zur Konzeption der Entwicklungsmethodik bildet das *Semantische Objektmodell*, zur Demonstration der Ergebnisse die Fallstudie *Mobility Provision*.

7.1 Anforderungen an die Entwicklung geschäftsprozessorientierter operativ-dispositiver Systeme

Bei Sommerville (2016, S. 18) werden zwei wesentliche Faktoren für das Scheitern von Software-Entwicklungsprojekten genannt: Einen Faktor stellt die *steigende Komplexität* von AWS, einen anderen das Fehlen adäquater Entwicklungskonzepte in System-Entwicklungsprojekten dar. Es besteht Bedarf, Entwickler durch Bereitstellung adäquater Konzepte bei ihrer komplexen Tätigkeit zu unterstützen und diese in der Handhabung der Konzepte zu schulen. Die Argumentation bzgl. der

Komplexität kann auch für geschäftsprozessorientierte operativ-dispositive AwS geführt werden. Wie in Kap. 1.1.2 mit Abgrenzung des Untersuchungsgegenstandes der Arbeit gezeigt wird, werden Teilsysteme zur operativen Durchführung von Geschäftsprozessen (*operatives AwS*), zur Bereitstellung entscheidungsrelevanter Daten (*pDWH-System*) sowie zur Entscheidungsunterstützung (*EU-System*) zu einem operativ-dispositiven AwS integriert. Die Entwicklung der einzelnen Systeme stellt für sich bereits eine komplexe Aufgabe dar. Hinzu kommen die Komplexität der Integration sowie der konsistenten Abstimmung dieser Teilsysteme. Die Zielsetzung des Kapitels ist die Unterstützung von Entwicklern während der Entwurfsphase geschäftsprozessorientierter operativ-dispositiver AwS durch eine modellbasierte Entwicklungsmethodik. Bestandteile einer solchen Methodik sind nach Bork & Sinz (2013) sowie Ferstl & Sinz (2013, S. 136f.) *Modellierungsansatz*, *Vorgehensmodell* und *Modellarchitektur*: Ein *Modellierungsansatz* ist durch *Metapher* und *Metamodell* gekennzeichnet. Die *Metapher* beschreibt die grundsätzliche Sichtweise, die der Modellerstellung zugrunde liegt. Im *Metamodell* sind darauf abgestimmt Modellbausteine und Beziehungen, Regeln zu deren Verknüpfung sowie deren Semantik festgelegt. Durch das *Architekturmodell* wird, zur Beherrschung der Komplexität, das Modellsystem in Ebenen und Sichten strukturiert. Das *Vorgehensmodell* beschreibt die schrittweise, an der Modellarchitektur ausgerichtete Modellbildung in ablauforientierter Form.

Als Zielsystem der nachfolgend konzipierten Methodik dient die komponentenorientierte Referenzarchitektur aus Kap. 5 sowie das pDWH-Schema aus Kap. 6 festgelegt. Aus dem Untersuchungsproblem (Kap. 1.2) der Arbeit sowie den Eigenschaften und Anforderungen von Architektur und Datenschema ergeben sich zwei zentrale Anforderungen an die Methodik, die im Nachfolgenden beschrieben und bei der Konzeption zu berücksichtigen sind:

Expliziter Geschäftsprozessbezug Wie in Kap. 3.5.2 identifiziert, können sowohl operative (*verhaltensorientierte*) als auch gestaltungsbezogene (*strukturorientierte*) Entscheidungsszenarien durch ein pDWH-System unterstützt werden. Beide Szenarien haben den Prozessbezug gemein. In Kap. 6 wird ein Datenschema für pDWH-Systeme vorgestellt, das diesen Prozessbezug explizit abbildet. Der Begriff *Prozess* wird dabei im Sinne von Workflow, also auf AT-Ebene, verwendet. Der Begriff *Geschäftsprozess* ist hingegen der Aufgabenebene zugeordnet und beschreibt

aus fachlicher Sicht die betriebliche Leistungserstellung sowie deren Lenkung (vgl. Kap. 2.2.1). Diese Perspektive wird durch das vorgestellte pDWH-Konzept (Architektur und Datenschema) jedoch noch nicht unterstützt. Dem Konzept des *Geschäftsprozesses* kommen im Weiteren somit zwei zentrale Bedeutungen zu. Zum einen bildet er den Ausgangspunkt zur Entwicklung betrieblicher AwS zur Durchführung automatisierbarer Aufgaben. Zum anderen spannt er auch den Kontext zur fachlichen Bewertung von Daten des pDWH-Systems auf. Diese enge Verknüpfung zwischen pDWH-System und GP-Modell zur Entwicklungs- und zur Laufzeit wird vergleichbar auch bei Becker & Chamoni (2008) thematisiert. Neben der Nutzung von GP-Modellen zur Ableitung operativer Systeme dienen diese auch zur Ableitung von DWH-Schemata sowie zur Visualisierung von Analyseergebnissen. Als Beispiel wird ein modellbasierter Soll-Ist-Vergleich angeregt (Becker & Chamoni 2008). Die zentrale Bedeutung des *Geschäftsprozesses* für die Ergebnisse der Arbeit drückt sich durch die Spezialisierung des Konzeptes der *Prozessorientierung* hin zur *Geschäftsprozessorientierung* (GP-Orientierung) aus. Sowohl während der Entwicklung der Systeme als auch bei der Bewertung der Ablaufdaten ist explizit der Bezug zur Aufgabenebene, also zum Geschäftsprozess, aufzuzeigen. Entscheidend ist die konsequente Nachvollziehbarkeit aller Entwicklungsschritte (*Traceability*). In der Konsequenz wird damit nicht nur die explizite Bewertung der betrieblichen Leistungserstellung, sondern auch der Zugang von (operativen) Nutzern mit betriebswirtschaftlichem Hintergrund zum DWH-System unterstützt (Anforderung A2, Kap.5.1).

Konsistenz der Teilsysteme Die Forderung der konsistent entwickelten Teilsysteme schließt an die vorangegangene Argumentation an. Wie dargestellt, bildet der *Geschäftsprozess* den zentralen Entwicklungsgegenstand. Dieser bildet die Grundlage zur Spezifikation des operativen AwS, zur Ableitung multidimensionaler Datenschemata sowie den Gegenstand der aktiven Steuerung im verhaltensorientierten pDWH-Szenario. Bzgl. der Qualität prozessbezogener Entscheidungen sowie der Interaktion zwischen den Teilsystemen ist es notwendig, dass diese konsistent zueinander entwickelt werden. Die Spezifikation des operativen AwS, des pDWH-Systems, des EU-Systems sowie deren Integration sind ausgerichtet an einem *zentralen Geschäftsprozessmodell* vorzunehmen.

7.2 Stand der Literatur: Entwicklung geschäftsprozessorientierter operativ-dispositiver Systeme

Zu dieser Arbeit verwandte Ansätze betreffen v.a. die Architektur operativ-dispositiver AwS (Kap. 5.2) sowie die Datenstrukturen von pDWH-Systemen (Kap. 6.3). Die methodisch geleitete Entwicklung unter Berücksichtigung der Konsistenz der Teilsysteme wird lediglich im Ansatz nach Becker (2007) bzw. Becker & Chamoni (2008) thematisiert. Es handelt sich dabei um ein umfassendes DWH-Konzept im Rahmen des Geschäftsprozess- und Workflow-Managements. Der Ansatz ist in die *Architektur integrierter Informationssysteme* (ARIS) nach Scheer (2001, 2002) eingebettet. Die Ableitung eines generischen pDWH-Schemas aus ARIS-Geschäftsprozessmodellen stellt einen Aspekt der Arbeit dar und wird bereits in Kap. 6.3 vorgestellt. Einen weiteren Aspekt bildet das Vorgehensmodell für die Einführung und den Betrieb eines *Process Warehouses* mit Bezug zum Geschäftsprozessmanagement (GPM). Das *Process Warehouse* wird als DWH-Anwendung interpretiert. Folglich wird auch ein etabliertes Vorgehen zur DWH-Entwicklung mit prozessorientierten Erweiterungen vorgeschlagen (Becker 2007, S. 286ff., Becker & Chamoni 2008). Die Ausgestaltung der einzelnen Phasen wird nicht vertieft beschrieben. Eine Differenzierung nach Abstraktions- bzw. Modellebenen (vergleichbar zu Aufgaben- und AT-Ebene) findet nicht statt. Innerhalb des GPM ist die Entwicklung des pDWH-Systems der Phase des *kontinuierlichen Prozessmanagements und -controllings* zugeordnet und damit der GP-Entwicklung nachgelagert. Zwischen den Vorgehensmodellen des GPM sowie der DWH-Entwicklung werden drei Abstimmungsbeziehungen definiert (Becker 2007, S. 289ff.): (i) Anforderungen des pDWH-Systems sind bereits während der GP-Entwicklung zu berücksichtigen. (ii) Ergebnisse des GPM-Projektes fließen ebenfalls in die Entwicklung des pDWH-Systems ein. (iii) Aus dem pDWH-System abgeleitete Analyseergebnisse sind im Rahmen des GPM für das Prozess-Controlling zu nutzen.

Weitere vergleichbare Ansätze sind nicht bekannt. Insofern können im Weiteren nur verwandte Arbeiten zur Entwicklung der Teilsysteme beleuchtet werden. Die Anforderung der *Konsistenz* ist damit nicht weiter relevant. In Abschn. 7.2.1 wird die Abgrenzung und Spezifikation von Komponenten auf Basis von Geschäftsprozessen betrachtet. Abschn. 7.2.2 hat die prozessorientierte Entwicklung von DWH-Systemen zum Gegenstand. Als relevant werden jeweils Arbeiten ein-

gestuft, die die nachfolgenden Kriterien aus Abschn. 7.1 erfüllen: Zum einen ist die methodisch geleitete Entwicklung anhand von Modellarchitektur oder Vorgehensmodell zu berücksichtigen. Zum anderen sollte explizit der GP-Bezug thematisiert sein.

7.2.1 Geschäftsprozessorientierte Entwicklung komponentenbasierter Anwendungssysteme

Der GP-Bezug bildet das wesentliche Kriterium zur Auswahl verwandter Arbeiten. Übertragen auf Ansätze zur Entwicklung komponentenorientierter AwS bedeutet dies, dass ein GP-Modell den Ausgangspunkt der methodisch geleiteten Entwicklung von Komponenten darstellt. Der notwendige erste Schritt zur fachlichen Abgrenzung von Komponenten wird im idealtypischen Entwicklungsprozess (vgl. Kap. 2.3.2) als *Component Identification* bezeichnet.

Basierend auf Modellarchitektur und Vorgehensmodell des *Semantischen Objektmodells* (vgl. Kap. 2.2) finden sich in der Literatur zwei Ansätze zur Identifikation von Komponenten. Ein erster Ansatz nach Friedrich et al. (2016) basiert auf der Methode von Birkmeier & Overhage (2012b) sowie den dazu gehörigen Vorarbeiten nach Albani et al. (2008) und Birkmeier & Overhage (2012a). Der Ansatz ermöglicht die systematische Ableitung von Fachkomponenten aus konzeptuellen Modellen. Die optimale Strukturierung der Komponenten beruht auf der Formalisierung des Prinzips *Minimierung der Kopplung bei Maximierung der Kohäsion*. Die Lösung des dadurch beschriebenen Optimierungsproblems erfolgt automatisiert und kann auf Basis von Simulationen bzgl. ihrer Belastbarkeit getestet werden. Gerade bei der Auswahl von konzeptuellen Modellen, von Modellelementen, von Beziehungen zwischen diesen sowie der Gewichtung von Beziehungen kann der Entwickler die Strukturierung beeinflussen und kontextspezifische Präferenzen berücksichtigen. Für diese mitunter komplexen Entscheidungen sind in der Methode ebenfalls unterstützende Konzepte vorgesehen. Bei Friedrich et al. (2016) wird die Anwendung dieser Methode auf SOM vorgestellt. Das KOS sowie das VOS als fachliche objektorientierte Anwendungsmodelle stellen den Input der Methode dar. Die Modelle der SOM-Methodik auf AT-Ebene werden in Abschn. 7.4.1 noch im Detail eingeführt. KOS und VOS enthalten in konsolidierter Form ausschließlich automatisierbare, also durch AwS unterstützte Objekte. Im KOS sind v.a. Generalisierungs- und Aggregationsbeziehungen zwischen diesen

zur Komponentenabgrenzung relevant. Innerhalb des VOS gilt dies für objektinterne Interaktionsbeziehungen (Friedrich et al. 2016). Bei Ferstl et al. (1997) wird ein weiterer Ansatz zur fachlichen Spezifikation von Komponenten (sog. *Application Objects*) basierend auf SOM vorgestellt. Hierbei dient jedoch die Geschäftsprozessspezifikation der Aufgabenebene als Ausgangspunkt. Die Abgrenzung der fachlichen Komponenten erfolgt anhand betrieblicher Objekte, die Spezifikation der Schnittstellen sowie des Interaktionsprotokolls zwischen diesen anhand der Transaktionen des IAS. Aus Innensicht sind die Komponenten durch den unterstützten Ausschnitt der AT-Spezifikation der SOM-Methodik (KOS, VOS) beschreibbar (Ferstl et al. 1997).

Die Methode nach Andresen (2003) beschreibt ebenfalls Modellarchitektur und Vorgehensmodell zur Komponentenabgrenzung. Diese sind entlang der Abstraktionsebenen der *Model Driven Architecture (MDA)*² spezifiziert. Die Ebenen werden dabei auch als *Teil-Architekturen* bezeichnet. Auf höchster Ebene der *Business Architektur* werden Anforderungen, Business Cases und Prozesse erfasst. Der Prozess-Begriff wird dabei im Sinne von Workflow verstanden. Die daraus abgeleitete *Referenz-Architektur* hat die fachliche Spezifikation der Komponenten zum Gegenstand (*Platform Independent Model* der MDA). Die Realisierung basierend auf einem konkreten Komponentenmodell (z.B. *Enterprise Java Beans*) erfolgt dann auf Ebene der *Anwendungs-Architektur (Platform Specific Model* der MDA). Mit der *System-Architektur* wird schließlich die Realisierung der Komponenten in einer systemspezifischen Infrastruktur beschrieben (*Enterprise Deployment Model* der MDA) (Andresen 2003, S. 70ff.). Zur Ausgestaltung der Modellsichten der einzelnen Abstraktionsebenen werden Vorgehensmodelle bereitgestellt (Andresen 2003, S. 81ff.).

Andere Ansätze thematisieren v.a. das *Vorgehen* zur Identifikation von Komponenten ausgehend von Geschäftsprozessen. Das Verständnis des Begriffs *Geschäftsprozess* (siehe Kap. 2.2.1) wird jedoch nicht näher erläutert. Der Begriff ist eher abstrakt im Sinne einer Sequenz von Aktivitäten zur Verarbeitung von Datenobjekten gehalten. Das Vorliegen von Modellen der fachlichen Domäne (u.a. von GP-Modellen) wird angenommen. Eine umfassende Spezifikation einer Entwicklungsmethodik ist i.d.R. nicht gegeben. Bei Levi & Arsanjani (2002) wird ein

²Spezifikationen zur MDA: <http://www.omg.org/mda/specs.htm> – Zugriff: 29.07.2017

Ansatz zur Identifikation und zur Spezifikation von Komponenten vorgestellt. Die Nachvollziehbarkeit des Komponentenentwurfs bzgl. der Geschäftsziele und -prozesse wird explizit als Mehrwert der Arbeit genannt. Geschäftsprozesse werden lediglich aus Außensicht in Form von Anwendungsfällen (*Use Cases*) verwendet und dienen der Strukturierung einer betrieblichen Domäne durch Komponenten. Zur strukturierten Ableitung und Spezifikation von Komponenten werden Handlungsempfehlungen gegeben. Meng et al. (2005) identifizieren Komponenten ausgehend von Modellen der fachlichen Domäne zu Funktions-, Daten- und Verhaltenssicht. Zentrale Artefakte stellen Aktivitäten und verarbeitete Datenobjekte dar. Methodisch basiert die Identifikation von Komponenten auf hierarchischen Cluster-Verfahren. Ähnlich ist auch die Arbeit von Cai et al. (2011) auf die genannten Sichten ausgerichtet. Aktivitäten und Datenobjekte stellen auch hier die zentralen Artefakte zur Abgrenzung von Komponenten dar. Zielsetzung der Arbeit ist die Ableitung von Daten- und Prozesskomponenten auf Basis von Konzepten der *Fuzzy Logic* und der *Formal Concept Analysis*.

Bei Birkmeier & Overhage (2012b) werden weitere Arbeiten zur Identifikation von Komponenten gegenübergestellt, die jedoch keinen expliziten Bezug zu Geschäftsprozessen aufweisen. Zahlreiche Arbeiten (z.B. Lee et al. 1999, Ganesan & Sengupta 2001, Jang et al. 2003) gehen dabei von objektorientierten Strukturbeschreibungen wie Klassendiagramm und Anwendungsfällen, andere Arbeiten (z.B. Kim & Chang 2004, Jain et al. 2001) zudem von Ablaufmodellen, beispielsweise in Form von Sequenzdiagrammen, aus. Das Vorliegen der Modelle als Ergebnis fachlicher Analysen wird jeweils vorausgesetzt. Die Arbeiten thematisieren die Identifikation von Komponenten durch Cluster-Verfahren oder Matrix-Analysen (Birkmeier & Overhage 2012b).

7.2.2 Entwicklung geschäftsprozessorientierter Data-Warehouse-Systeme

Verwandte Entwicklungsansätze für pDWH-Systeme (bzw. deren Datenschemata) basieren auf der Spezifikation der betrieblichen AwS, die für die Ausführung der Geschäftsprozesse verantwortlich sind. Nach Santos & Oliveira e Sá (2016) und Sturm (2012) lassen sich diese Entwicklungskonzepte als *struktur-* oder *prozessorientiert* klassifizieren:

Strukturorientierte Entwicklungsmethoden (z.B. nach Boehnlein & Ulbrich-vom Ende 1999 oder Golfarelli et al. 1998) beschreiben die Ableitung von DWH-Schemata aus den Datenmodellen der operativen AwS. Die Datenmodelle werden dabei als gegeben angenommen. Der Bezug zum Geschäftsprozess wird i.d.R. nicht thematisiert. Der Ansatz nach Böhnlein (2001, S. 303ff.) kann ebenfalls als *struktur-*, aber auch als *zielorientiert* charakterisiert werden. Die beschriebenen Hinweise zur Ableitung multidimensionaler Datenschemata basieren auf dem objektorientierten KOS zur Beschreibung operativer Datenstrukturen. Das Vorgehen ist explizit in die SOM-Methodik eingebettet und ermöglicht damit die Spezifikation von DWH-Strukturen mit explizitem Bezug zum operativen AwS sowie zur übergeordneten GP-Spezifikation. Das resultierende DWH-Schema ist auf die Bewertung von Geschäftsprozessen und ihrer Leistung ausgerichtet sowie unter Berücksichtigung der Unternehmensziele spezifiziert. DWH-Schemata nach Böhnlein (2001) ermöglichen explizit die Analyse des Prozessergebnisses, nicht jedoch des Prozessablaufs.

Die Gruppe der *prozessorientierten Entwicklungsansätze* bezieht auch Prozessmodelle in die Ableitung von DWH-Strukturen mit ein. Zentrale Motivation ist es, in Entscheidungen zur Prozessgestaltung auch Zugang zu Verhaltensaspekten eines Prozesses zu erhalten. Bei Santos & Oliveira e Sá (2016) werden Regeln zur automatisierten Ableitung multidimensionaler Datenstrukturen aus Prozessen vorgestellt. Als Prozesssprache wird die BPMN in der Version 2.0 betrachtet. Ausgehend von operativen Datenstrukturen, die automatisiert aus den Prozessmodellen abgeleitet sind, sowie *Participants* in BPMN-Modellen können wiederum automatisiert, DWH-Schemata abgeleitet werden. Die Entwicklung von DWH-Schemata nach Dori et al. (2008) basiert auf der *Object-Process-Methodology* nach Dori (2002). Die Methode stellt einen integrierten Ansatz zur Unterstützung der Analyse und Entwicklung von Informationssystemen über deren Lebenszyklus hinweg dar. Ausgehend von mit dieser Methode spezifizierten Prozessen, werden regelbasiert alternative DWH-Strukturen vorgeschlagen. Die Auswahl eines Schemas erfolgt kontextabhängig durch den Entwickler. In engen Bezug dazu stehen die Arbeiten von Sturm (2008, 2012) die ebenfalls die regelbasierte Ableitung eines *Snowflake*-Schemas aus Modellen des operativen Systems beschrieben. Als Ausgangspunkt der Ableitung dienen dabei jedoch BPMN-Prozessdiagramm, UML-Klassendiagramm sowie eine Spezifikation der Organisationsstrukturen auf AT-

Ebene. Die resultierenden DWH-Strukturen sind auf die Analysephase im GP-Lebenszyklus ausgerichtet. Das bereitgestellte Regelwerk kann auf Prozessbausteine unterschiedlicher Abstraktionsstufen (z.B. Gesamtprozess, Sub-Prozesse oder Aktivitäten) angewendet werden.

Eine weitere Gruppe von Arbeiten, die auch häufig als prozessorientiert charakterisiert werden, stellen Zielsetzungen, die durch das DWH-System zu bewerten sind, in den Fokus. Der explizite Bezug zum Geschäftsprozessmodell ist in diesen Ansätzen jedoch nicht gegeben. In der vorliegenden Arbeit werden diese Ansätze als *zielorientiert* charakterisiert. Bei Niedrite et al. (2007) wird eine *zielorientierte* Methode zur Spezifikation von DWH-Strukturen zur Prozessanalyse (Workflows) vorgestellt. Auf Basis von Interviews mit Prozessbeteiligten werden mit dem Prozess assoziierte Zielsetzungen und *Indikatoren* (konkrete Fragestellungen) zu deren Bewertung identifiziert. Gleichzeitig werden die zentralen Prozessentitäten (z.B. Ressourcen, Produkte, Aktivitäten, Sub-Prozesse) mit ihren Beziehungen in einem Strukturmodell erfasst. Auf Basis dieses Modells sowie der erfassten Indikatoren wird eine multidimensionale Datenstruktur zur Analyse der Prozessziele abgeleitet. Auch bei List & Machaczek (2004) wird *zielorientiert* ein multidimensionales Prozessdatenschema spezifiziert. Ausgehend von Unternehmenszielen werden GP-spezifische Ziele abgeleitet und Fragestellungen zu deren Bewertung formuliert. Auf Basis der Fragestellung erfolgt die Spezifikation des DWH-Schemas. Generelle Zielsetzung der Arbeit ist die Erweiterung von *Corporate Performance Measurement Systemen* um eine Prozess-Perspektive.

7.3 Entwicklungsmethodik für geschäftsprozessorientierte operativ-dispositive Systeme

Gegenstand des Abschnittes ist die ganzheitliche Konzeption der Entwicklungsmethodik für GP-orientierte operativ-dispositive AwS. In Abschn. 7.3.1 werden dazu grundlegende Überlegungen erläutert. Anschließend werden in Abschn. 7.3.2 die Methodik zur Entwicklung des *operativen Systems* sowie in Abschn. 7.3.3 deren Erweiterung zur Entwicklung des *pDWH-Systems* eingeführt. Der Zusammenhang wird jeweils anhand von Vorgehensmodell und Modellarchitektur vorgestellt. Details der einzelnen Teilsysteme werden in den folgenden Abschnitten betrachtet (Abschn. 7.4 für das *operative*, Abschn. 7.5 u. 7.6 für das *pDWH-System*).

7.3.1 Grundlegende Konzeption der Entwicklungsmethodik

Bei der Entwicklung GP-orientierter operativ-dispositiver AwS sind die drei in Kap. 1.1.2 eingeführten Teilsysteme *operatives System*, *pDWH-System* sowie *EU-System* zu berücksichtigen. Aufgrund der Abhängigkeiten der Teilsysteme kann die Entwicklung grundsätzlich sequentiell durchgeführt werden. Die abgeschlossene Entwicklung des operativen Systems zur Automatisierung von Geschäftsprozessen stellt die Voraussetzung für die Gestaltung des pDWH-Systems dar. Aus der GP- sowie der zugehörigen AwS-Spezifikation werden Datenstrukturen zur GP-Analyse abgeleitet und für Nutzer des pDWH-Systems bereitgestellt. Das EU-System kann wiederum als Nutzer des pDWH-Systems interpretiert werden. Der automatisierte Geschäftsprozess selbst kann, neben seiner Rolle als Gegenstand eines pDWH-Systems, gleichzeitig auch Nutzer des EU-Systems sein. Als Beispiel ist hier die Steuerung des Prozessablaufs auf Basis komplexer (automatisiert getroffener) Entscheidungen des EU-Systems zu nennen. Unter Berücksichtigung der aufgezeigten Abhängigkeiten kann die Entwicklung der Teilsysteme zeitlich entkoppelt sequentiell erfolgen. Anhand der zugrunde liegenden Markt-Metapher von *Angebot und Nachfrage* (vgl. Kap. 1.3) werden auch die Teilsysteme der *flexiblen Data-Mart-Architektur* zeitlich entkoppelt (vgl. Kap. 5.3.2). Die Entwicklung des *angebotsorientierten* pDWH-Teilsystems ist an die Entwicklung des operativen AwS gebunden und deckt die Bereitstellung umfassender Prozessdaten durch *erweiterte Datenquellen* ab. Das *nachfrageorientierte* pDWH-Teilsystem wird auf Grundlage dieses Informationsangebots realisiert. Dies kann mitunter auch zur Laufzeit des Gesamtsystems erfolgen. Praktische Erfahrungen (z.B. bei Benker & Jürck 2012) zeigen, dass Veränderungen der Anforderungen an DWH-Systeme während der Laufzeit die Regel darstellen. Im Rahmen der Anforderungen an die Referenzarchitektur in Kap. 5.1 ist dies über die Anforderung der flexiblen Anpassbarkeit berücksichtigt. Aus den Abhängigkeiten der einzelnen Teilsysteme ergeben sich die nachfolgend zusammengefassten Entwicklungsschritte. Diese sind idealtypisch in der angegebenen Reihenfolge durchzuführen, was Rücksprünge oder Rückkopplungen keineswegs ausschließt:

1. Entwicklung des operativen Anwendungssystems
2. Entwicklung des geschäftsprozessorientierten DWH-Systems
 - a) Entwicklung des angebotsorientierten DWH-Teilsystems
 - b) Entwicklung des nachfrageorientierten DWH-Teilsystems
3. Entwicklung des Entscheidungsunterstützungssystems

Die Entwicklung der einzelnen Teilsysteme für sich stellt bereits eine komplexe Aufgabe dar. Diese Komplexität sowie die der Integration gilt es in der Entwicklungsmethodik zu beherrschen. Hierzu werden je Teilsystem Modellarchitektur (basierend auf dem Konzept des generischen Architekturrahmens, vgl. Kap. 2.1.3) sowie Vorgehensmodell vorgestellt. Zur Berücksichtigung der sequentiellen Abhängigkeiten werden die Architekturen zueinander in Beziehung gesetzt. Die Methodik zur Entwicklung des *angebotsorientierten* sowie des *nachfrageorientierten* pDWH-Systems stellen sukzessive Erweiterungen der Methodik zur Entwicklung des operativen AwS dar. Das EU-System wird in dieser Arbeit lediglich aus Außensicht betrachtet. Die Spezifikation der Innensicht ist nicht Gegenstand des Untersuchungsproblems. Auf die Spezifikation von Modellarchitektur und Vorgehensmodell wird deshalb verzichtet. Das EU-System wird im Kontext der Spezifikation nachfrageorientierter Data Marts in Abschn. 7.6 als Nutzer interpretiert und als solcher in die Gesamtmethodik eingeordnet.

Die Entwicklungsmethodik ist auf die Spezifikation einer fachlichen Komponentenarchitektur für geschäftsprozessorientierte operativ-dispositive Systeme ausgerichtet. Das Zielsystem ist dabei durch die Referenzarchitektur aus Kap. 5 sowie das pDWH-Schema aus Kap. 6 festgelegt. Die zu entwickelnden Artefakte begründen sich aus dem Zielsystem und dienen der Spezifikation sowie der Parametrisierung der Architekturbestandteile.

7.3.2 Methodik zur Entwicklung des operativen Teilsystems

Zur Entwicklung des *operativen Systems* wird die etablierte SOM-Methodik, wie in Kap. 2.2 vorgestellt, aufgegriffen. Die Spezifikation betrieblicher Informationssysteme auf Basis der SOM-Methodik erfolgt modellbasiert auf Aufgaben- und AT-Ebene. Für die Modellerstellung innerhalb von sowie zwischen Modellebenen sind regelbasierte Modelloperationen vorgegeben, wenngleich dem Modellierer

Freiheiten bei deren Anwendung zugestanden werden. Die SOM-Methodik ermöglicht basierend auf Modellarchitektur und Vorgehensmodell die nachvollziehbare und dokumentierbare Entwicklung betrieblicher AwS ausgehend von einer GP-Spezifikation. Aufgrund dessen ist die SOM-Methodik auch als erweiterbare Grundlage zur ganzheitlichen Entwicklung GP-orientierter operativ-dispositiver AwS geeignet.

Modellarchitektur Die Modellarchitektur ist in Abb. 7.1 dargestellt und orientiert sich mit dem *Geschäftsprozessmodell* auf Aufgabenebene, dem *fachlichen Anwendungsmodell* auf AT-Ebene sowie der modellbasierten Spezifikation der Modelle an der Modellarchitektur der SOM-Methodik, wie in Kap. 2.2.2 eingeführt. Zur Entwicklung komponentenorientierter AwS wird die AT-Ebene um das *komponentenorientierte Anwendungsmodell*, das modellbasiert aus dem fachlichen Anwendungsmodell abgeleitet wird, erweitert. Auf dieser Ebene wird die komponentenorientierte Perspektive auf die Automatisierung von Geschäftsprozessen eingenommen. Das verwendete Komponentenmodell abstrahiert von technischen Details konkreter Plattformen. Damit wird, wie auch bei Andresen (2003, S. 70ff.) ausgeführt, die Realisierung des AwS auf heterogenen Komponenten-Plattformen ermöglicht.

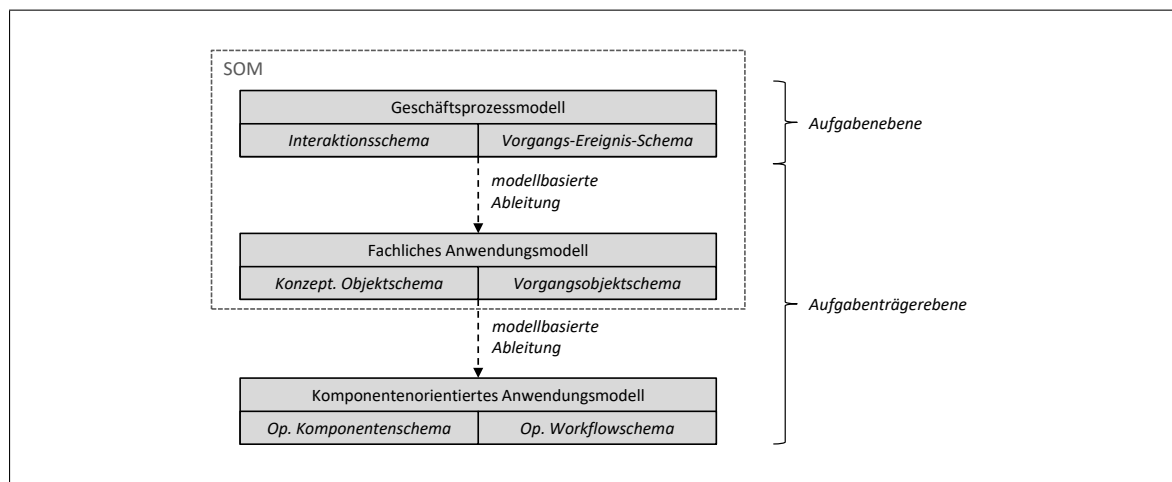


Abb. 7.1: Modellarchitektur zur Entwicklung des operativen Systems

Vorgehen Das Vorgehen zur Spezifikation von Geschäftsprozess- und Anwendungsmodell erfolgt analog zur SOM-Methodik, jeweils mit Struktur- und Verhaltenssicht, *top-down* (vgl. Kap. 2.2.3). Im fachlichen Anwendungsmodell erfolgt die Spezifikation der Verhaltenssicht (Vorgangsobjektschema) auf Basis der Arbeit

von Pütz & Sinz (2010). Damit wird die durchgängige Anwendung der BPMN zur Verhaltensmodellierung auf AT-Ebene ermöglicht. Im komponentenorientierten Anwendungsmodell erfolgt die Spezifikation des AwS ebenfalls aus Struktur- und Verhaltenssicht. Der Referenzarchitektur aus Kap. 5 folgend, sind Komponenten zur operativen Datenverwaltung sowie zur Vorgangssteuerung zu bilden und zu parametrisieren. Aus strukturorientierter Sicht (beschrieben durch das *operative Komponentenschema*) sind Komponenten abzugrenzen und deren Kommunikationsbeziehungen zu spezifizieren. Die verhaltensorientierte Sicht (*operatives Workflowschema*) spezifiziert den Ablauf zur Realisierung höherer GP-Funktionalität. Die Spezifikation der Verhaltenssicht erfolgt ebenfalls auf Basis der BPMN und stellt eine Verfeinerung des fachlichen Anwendungsmodells unter Berücksichtigung der operativen Komponenten dar.

7.3.3 Erweiterung der Methodik zur Entwicklung des prozessorientierten Data-Warehouse-Systems

Die Entwicklung des GP-orientierten DWH-Systems erfolgt zeitlich entkoppelt mit *angebots-* und *nachfrageorientierten* Teilsystem. In diesen beiden Ausbaustufen wird das *komponentenorientierte Anwendungsmodell* zur Beschreibung der Datenbereitstellung von der Datenquelle hin zum Nutzer sukzessive erweitert. Hinzukommende Schemata werden unter dem Konzept des *komponentenorientierten DWH-Modells* zusammengefasst. Die detaillierte Spezifikation der Teilsysteme wird in Abschn. 7.5 erläutert.

Entwicklung des angebotsorientierten DWH-Teilsystems

Die Entwicklung des *angebotsorientierten DWH-Systems* erweitert Vorgehensmodell und Modellarchitektur zur Entwicklung des operativen Teilsystems. Zielsetzung stellt die Spezifikation der angebotsorientierten Datenbereitstellung der Referenzarchitektur aus Kap. 5 dar. Operative Datenkomponenten sind mit multidimensionalen Informationen über die von ihnen verwalteten operativen Datenstrukturen zu parametrisieren.

Modellarchitektur Die Modellarchitektur zur *angebotsorientierten DWH-Entwicklung* ist in Abb. 7.2 rechtsseitig als eigenständige Architektur dargestellt. Zur Architektur des operativen Systems sind Ableitungsbeziehungen spezifiziert. Das

globale DWH-Modell stellt den zentralen Entwicklungsgegenstand dar. Innerhalb dieses Modells wird sowohl eine AT-spezifische (*AT-DWH-Schema*) als auch eine übergeordnete aufgabenspezifische (*GP-DWH-Schema*) Perspektive auf den zu überwachenden Geschäftsprozess ermöglicht. Diese, auf verschiedenen Modell-ebenen angeordneten Perspektiven spiegeln sich in der vertikalen Anordnung der Schemata wieder. Die Erweiterung um die GP-spezifische Perspektive führt zur Spezialisierung eines *prozessorientierten* hin zu einem *geschäftsprozessorientierten DWH-System* (gpDWH-System). In Kap. 5.5 wird bereits die Schemaarchitektur des pDWH-Schemas eingeführt. Anhand dieser wird u.a. die Aufteilung (*Partitionierung*) sowie die Zuordnung (*Allokation*) des *globalen DWH-Modells* beschrieben. Diese Bestandteile werden in die Modellarchitektur übertragen und im *Verteilungsmodell* (aufgrund von Abhängigkeiten ebenfalls vertikal) angeordnet. Durch das *Verteilungsmodell* wird die Brücke von der multidimensionalen Gesamtsicht hin zur Parametrisierung der operativen Datenkomponenten gebildet.

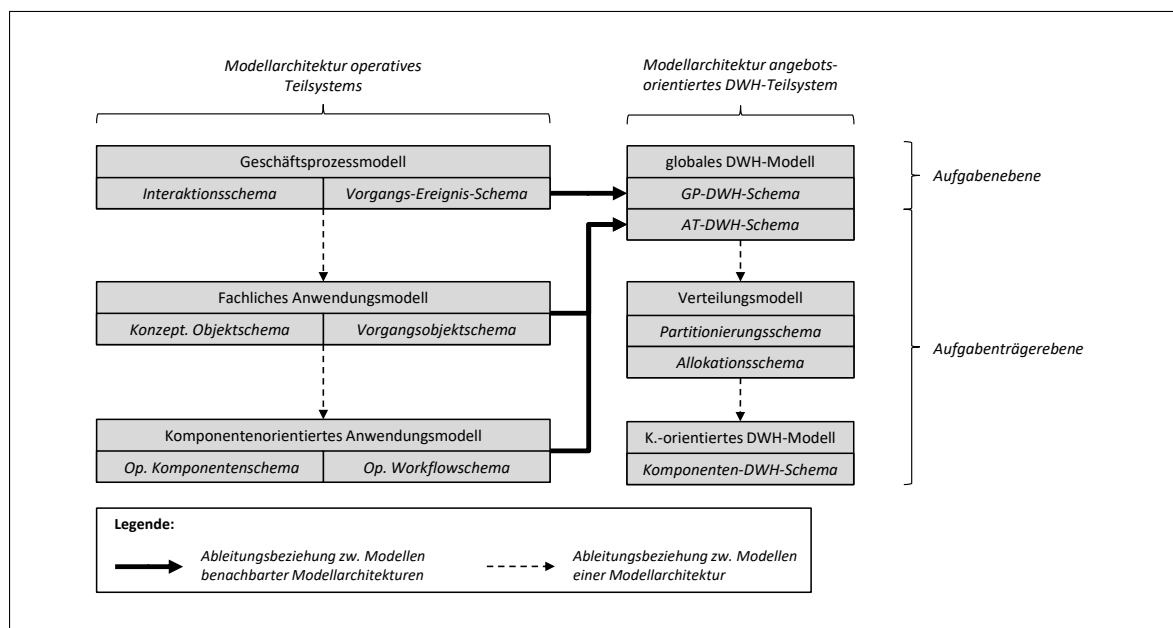


Abb. 7.2: Modellarchitektur zur Entwicklung der angebotsorientierten Datenbereitstellung

Vorgehen Die Spezifikation des *AT-DWH-Schemas* stellt den Einstieg zur Entwicklung des *angebotsorientierten DWH-Teilsystems* dar. Basierend auf den Anwendungsmodellen der AT-Ebene (Workflowschema und operative Datenstrukturen) wird eine multidimensionale Datenstruktur zur Workflow-Analyse abgeleitet. Hierzu findet das unter Kap. 6 beschriebene pDWH-Schema Anwendung. Zur

Einordnung der Ergebnisse in den GP-Kontext wird, basierend auf dem *AT-DWH-Schema*, das *GP-DWH-Schema* auf Aufgabenebene abgeleitet. Dieses stellt ausgehend vom Geschäftsprozessmodell eine fachliche Sicht auf die multidimensionalen Workflow-Daten dar. Zusammen bilden AT- und GP-DWH-Schema das *globale DWH-Modell* mit Unterstützung der Aufgaben- und der AT-spezifischen Perspektive. Die Verteilung des multidimensionalen Datenschemas zur Parametrisierung der operativen Datenkomponenten erfolgt auf Basis der von den Komponenten unterstützten Datenstrukturen. Auf dieser Grundlage wird das *AT-DWH-Schema* partitioniert und die Allokation zu den Komponenten festgelegt. Die von den Komponenten zu unterstützenden multidimensionalen Teilstrukturen sind im *Komponenten-DWH-Schema* beschrieben.

Entwicklung des nachfrageorientierten DWH-Teilsystems

Das *nachfrageorientierte* DWH-Teilsystem realisiert die Informationsanforderungen verschiedener Nutzergruppen. Die Erweiterung des DWH-Systems kann sowohl während der Entwicklungszeit als auch während der Laufzeit des Gesamtsystems stattfinden. Zielsetzung ist die Entwicklung von *Data-Mart-Komponenten* einschl. Datenschema sowie von Komponenten zur *Datenintegration*.

Modellarchitektur Die Modellarchitektur zur Entwicklung des nachfrageorientierten DWH-Teilsystems ist in Abb. 7.3 dargestellt. Dabei ist auch die Beziehung zur angebotsorientierten Modellarchitektur gezeigt. Ausgangspunkt bildet das *Data-Mart-Modell* (beschrieben durch das *Data-Mart-Schema*), das je nach Informationsbedarf der Zielgruppe sowohl multidimensionale Strukturen zur Aufgaben- als auch zur AT-Ebene enthalten kann. Das *Komponenten-DM-Schema* beschreibt die Abbildung der Strukturen in *Data-Mart-Komponenten*. Das *Integrationsschema* beschreibt die Parametrisierung von Komponenten zur Datenintegration.

Vorgehen Auf Basis zielgruppenspezifischer Informationsanforderungen (Anforderungsanalyse nicht Gegenstand der Arbeit) wird auf Basis des Informationsangebotes (*globales DWH-Modell*) das multidimensionale *Data-Mart-Schema* abgeleitet. Dieses enthält anhand der Nutzeranforderungen gebildete und bzgl. Granularität sowie Perspektive aufbereitete Daten. Ausgehend davon erfolgt die Bildung von Data-Mart- und Integrationskomponenten der Referenzarchitektur aus Kap. 5. Data-Mart-Komponenten werden mit dem *Komponenten-DM-Schema*,

Komponenten zur Datenintegration mit dem *Integrationsschema* parametrisiert. Dabei ist die Verteilung des *globalen DWH-Modells* anhand des *Verteilungsmodells* zu berücksichtigen.

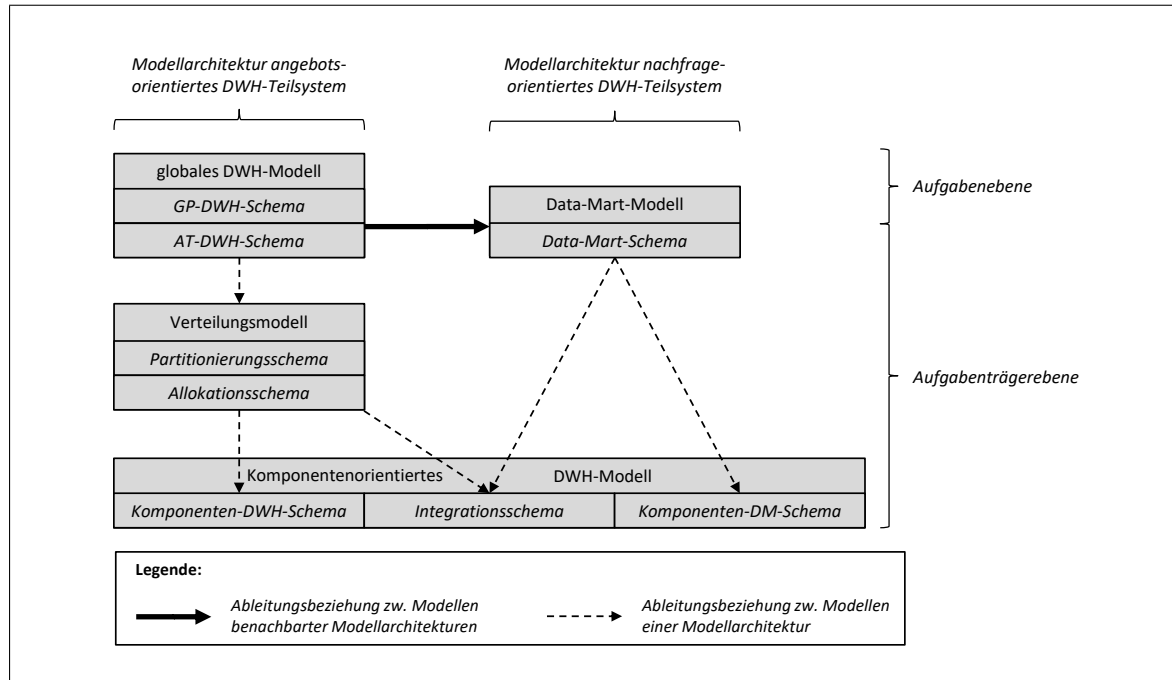


Abb. 7.3: Modellarchitektur zur Entwicklung der nachfrageorientierten Datenbereitstellung

7.4 Entwicklung des operativen Anwendungssystems

Die Entwicklung des operativen Teilsystems, basierend auf der Methodik aus Abschn. 7.3.2, erfolgt *top-down*. Ausgehend vom Geschäftsprozessmodell (vgl. Kap. 4) wird im ersten Schritt zunächst ein fachliches Anwendungsmodell (Abschn. 7.4.1) abgeleitet. Im zweiten Schritt wird dieses in eine komponentenorientierte Anwendungsspezifikation überführt (Abschn. 7.4.2).

7.4.1 Spezifikation des fachlichen Anwendungsmodells

Die Spezifikation des fachlichen Anwendungsmodells erfolgt gemäß SOM-Methodik. Das Metamodell der AT-Ebene (Abb. 7.4) folgt dem objektorientierten Paradigma (Ferstl & Sinz 1990). Zentraler Gegenstand ist der *Objekttyp*, der durch *Operatoren* und *Attribute* näher beschrieben werden kann. Objekttypen können miteinander in Beziehungen stehen. Als Beziehungstypen werden *is_a* (Vererbungsbeziehung zur Generalisierung bzw. Spezialisierung von Objekttypen), *interacts_with*

(Assoziationsbeziehung zur Spezifikation der Interaktionskanäle zwischen Objekttypen) sowie *is_part_of* (Aggregationsbeziehung zur Modellierung von Teil-Ganzes-Beziehungen zwischen Objekttypen) unterschieden (Ferstl & Sinz 1990). Auch auf AT-Ebene erfolgt die Modellierung aus struktur- (KOS) und verhaltensorientierter (VOS) Sicht. Die Differenzierung im Metamodell erfolgt über spezialisierte Objekttypen. Innerhalb des KOS werden nachfolgende *konzeptuelle Objekttypen* (KOT) unterschieden (Ferstl & Sinz 2013, S. 226): *objektspezifischer Objekttyp* (OOT), *leistungsspezifischer Objekttyp* (LOT), *transaktionsspezifischer Objekttyp* (TOT); das VOS wird anhand des aufgabenspezifischen *Vorgangsobjekttyp* (VOT) beschrieben (Ferstl & Sinz 1990). Nicht dargestellt sind Objekttypen des *Interface-Objektschemas* zur Spezifikation von Schnittstellen mit personellen oder maschinellen Nutzern sowie Objekttypen, die explizit die Persistenz von KOTs bestimmen. Diese sind für die weitere Arbeit nicht von Relevanz.

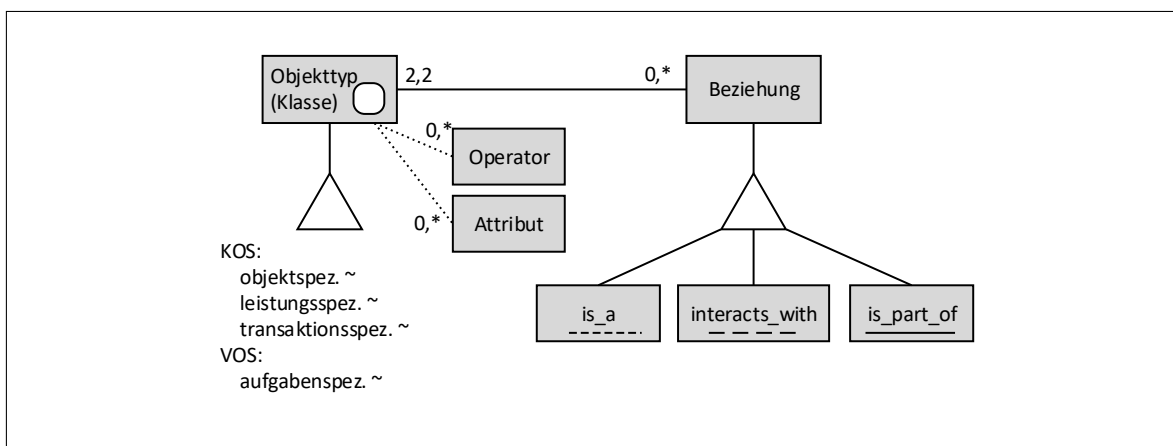


Abb. 7.4: Metamodell für SOM-Anwendungsmodelle (Ferstl & Sinz 1993, Ferstl & Sinz 2013, S. 233)

Ableitung und Konsolidierung des Konzeptuellen Objektschemas

Das *konzeptuelle Objektschema* (KOS) stellt die strukturorientierte Sicht auf das fachliche Anwendungsmodell (AT-Ebene) dar. Das initiale Modell leitet sich gemäß SOM-Methodik modellbasiert aus der Aufgabenebene, der GP-Spezifikation der SOM-Methodik ab. Die Transformationsregeln sind in Tab. 7.1 dargestellt. Die Anordnung von Objekttypen im resultierenden KOS erfolgt anhand von Existenzabhängigkeiten, wie initial bei Sinz (1988) für den Modellierungsansatz *Strukturiertes Entity-Relationship-Modell* (SERM) vorgestellt (Ferstl & Sinz 1990). *Diskurswelt-* und *Umweltobjekte* werden nach Ferstl & Sinz (2013, S. 225) als existenzunabhängige *objektspezifische* (OOT), die Leistungsspezifikation des Geschäftspro-

zesses als *leistungsspezifischer Objekttyp* (LOT) linksseitig angeordnet. *Transaktionsspezifische Objekttypen* (TOT) leiten sich aus Transaktionen des Geschäftsprozessmodells ab und sind existenzabhängig von KOTs der Diskurs- und Umweltobjekte und ggf. der Leistungsspezifikation. TOTs werden über *interacts_with*-Beziehungen mit den entsprechenden OOTs bzw. LOTs verknüpft. Zur Aufdeckung von Existenzabhängigkeiten zwischen TOTs werden Sequenzen von Transaktionen des VES analysiert (Ferstl & Sinz 2013, S. 225). Innerhalb eines KOS nimmt der Grad der Existenzabhängigkeit von links nach rechts zu.

SOM-Geschäftsprozessmodell	Konzeptuelles Objektschema
Diskurswelt- und Umweltobjekt	existenzunabh. objektspezifischer Objekttyp (OOT)
Leistungsspezifikation (siehe Leistungssicht)	existenzunabh. leistungsspezifischer Objekttyp (LOT)
Transaktion	existenzabh. transaktionsspezifischer Objekttyp (TOT) verknüpft durch <i>interacts_with</i> -Beziehung

Tab. 7.1: Transformationsregeln zur Ableitung des konzeptuellen Objektschemas aus SOM-Geschäftsprozessmodellen (Ferstl & Sinz 2013, S. 226)

Auf das initial abgeleitete KOS werden die nachfolgenden Konsolidierungsschritte nach Ferstl & Sinz (2013, S. 226) zur weiteren Spezifikation des objektorientierten Anwendungsmodells angewendet:

1. KOTs, die ausschließlich mit nicht automatisierten Aufgaben oder Transaktionen assoziiert sind, werden entfernt.
2. Die *interacts_with*-Beziehungen zwischen KOTs sind durch Kardinalitäten in min-max-Notation ((0,1), (0,*), (1,1), (1,*)) näher zu spezifizieren.
3. KOTs werden Attribute zugeordnet. Komplexe Objekte können anhand von *is_part_of*-Beziehungen aufgespalten (normalisiert) werden. Über den Beziehungstyp *is_a* können Generalisierungen spezifiziert werden.
4. KOTs sind Nachrichtendefinitionen und Operatoren zuzuordnen. Dies können einfache Zugriffsoperatoren sein oder Operatoren, die sich im Zuge der Spezifikation der Verhaltenssicht ergeben.
5. Zur Vermeidung von Funktions- und Datenredundanz können KOTs zusammengefasst werden. Hinweise liefern überlappende Attribute und Operatoren von KOTs.

Anhand des Geschäftsprozessmodells der Fallstudie *Mobility Provision* (Kap. 4.3) wird die Anwendung der Transformations- und Konsolidierungsschritte demonstriert. Das resultierende KOS ist in Abb. 7.5 dargestellt (Notation UML-Klassendiagramm), Transformation und Konsolidierung in Tab. 7.2 zusammengefasst. Jeder KOT wird dabei anhand der durchgeführten Transformations- und Konsolidierungsschritte begründet. Das Umweltobjekt *Kunde* führt so beispielsweise zu einem OOT *Kunde*. Die Leistungsspezifikation *Mobilität* der Leistungssicht des Geschäftsprozessmodells beschreibt Miet- und Abrechnungsmodelle und wird durch Konsolidierung zu einem LOT *Mietmodell*. Exemplarisch für die Transformation und Konsolidierung von Transaktionen kann der TOT *ind. Angebot* betrachtet werden. Die TOTs der Transaktionen hin zum *ind. Angebot* werden aufgrund überlappender Attribute zu einem TOT *ind. Angebot* konsolidiert. Durch die Abgrenzung des AwS zur Automatisierung des Vertriebsprozesses (Aufgaben der betrieblichen Objekte *Auftragsverwaltung* und *Vertriebskanal*) finden KOTs zu den betrieblichen Objekten *Flottenlenkung*, *Flottenbereitstellung* und *Finanzwesen* sowie zu Transaktionen zwischen diesen in der Konsolidierung keine Berücksichtigung.

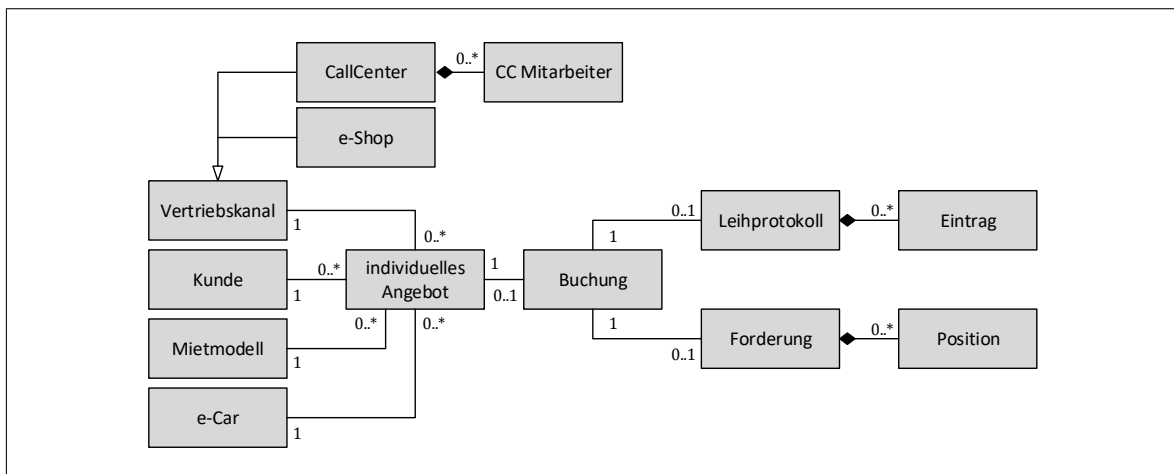


Abb. 7.5: Konsolidiertes konzeptuelles Objektschema zur Fallstudie *Mobility Provision*

KOT (Abb. 7.5)	Transformation [IAS zu KOS]	Konsolidierung
Vertriebskanal	Diskursweltobj. <i>Vertriebskanal</i> [OOT]	
e-Shop		Spezialisierung des OOTs <i>Vertriebskanal (is_a-Bez.)</i>
CallCenter		Spezialisierung des OOTs <i>Vertriebskanal (is_a-Bez.)</i>
CC-Mitarbeiter		Normalisierung des Typs <i>CallCenter (is_part_of-Bez.)</i>
Kunde	Umweltobj. <i>Kunde</i> [OOT]	
Mietmodell	Leistungsspez. <i>D: Mobilität</i> [LOT]	Attribute und Operatoren des Typs <i>Mietmodell</i> für AwS relevant
e-Car	Diskursweltobj. <i>Flottenlenkung</i> [OOT] Diskursweltobj. <i>F.-bereitstellung</i> [OOT] Transaktion <i>D: Verf.-Liste</i> [TOT]	Attribute und Operatoren des Typs <i>e-Car</i> für AwS relevant; Zusammenfassung mit TOT aufgrund überlappender Attribute
ind. Angebot	Transaktion <i>V: Anfrage</i> [TOT] Transaktion <i>V: Verf.-prüfung</i> [TOT] Transaktion <i>V: Auftrag Vorres.</i> [TOT] Transaktion <i>V: Best. Vorres.</i> [TOT] Transaktion <i>V: ind. Angebot</i> [TOT]	Zusammenfassung der TOTs aufgrund überlappender Attribute in TOT <i>ind. Angebot</i>
Buchung	Transaktion <i>V: B.-Auftrag</i> [TOT] Transaktion <i>V: B. Dokument.</i> [TOT] Transaktion <i>V: A. Reservierung</i> [TOT] Transaktion <i>D: B. Reservierung</i> [TOT] Transaktion <i>V: Buchung</i> [TOT] Transaktion <i>V: B.-Bestätigung</i> [TOT]	Zusammenfassung der TOTs aufgrund überlappender Attribute in TOT <i>Buchung</i>
Leihprotokoll	Transaktion <i>V: Anf. e-Car</i> [TOT] Transaktion <i>V: Rückgabe e-Car</i> [TOT] Transaktion <i>D: F.-Protokoll</i> [TOT]	Dokumentation der Ausleihe im TOT <i>Leihprotokoll</i> ; Zusammenf. aufgr. überlappender Attribute
Eintrag		Normalisierung des TOTs <i>Leihprotokoll (is_part_of-Bez.)</i>
Forderung	Transaktion <i>V: Forderung</i> [TOT] Transaktion <i>D: Zahlungsmeld.</i> [TOT]	Zusammenfassung im TOT <i>Forderung</i>
Position		Normalisierung des TOTs <i>Forderung (is_part_of-Bez.)</i>

Tab. 7.2: Transformations- und Konsolidierungsschritte der Fallstudie *Mobility Provision*

Ableitung des BPMN-Workflowschemas

In der SOM-Methodik wird die Verhaltenssicht auf das fachliche Anwendungsmodell ebenfalls basierend auf dem objektorientierten Paradigma im *Vorgangsobjektschema* (VOS) beschrieben. Über *Vorgangsobjekttypen* (VOT) wird die Steuerung der KOTs zur Realisierung betrieblicher Aufgaben spezifiziert (Ferstl & Sinz 1991). Bei Pütz & Sinz (2010) werden die Modellierung sowie die modellbasierte Ableitung der Verhaltenssicht auf Basis der BPMN als Alternative vorgestellt. Dabei bildet nicht mehr der objektorientierte VOT mit Struktur und Verhalten sondern das ablauforientierte Konzept der *Aktivität* den zentralen Baustein. Die nach Pütz & Sinz (2010) ableitbare initiale BPMN-basierte Verhaltensbeschreibung des fachlichen Anwendungsmodells ist eine geeignete Grundlage zur anschließenden Spezifikation des Verhaltens komponentenorientierter Systeme und wird für die weitere Entwicklung übernommen. Dieses initiale BPMN-Schema dient als Ausgangspunkt für die hierarchische Verfeinerung hin zu ausführbaren Workflows. Damit bleibt zum einen die Durchgängigkeit der Modellierungssprache gewahrt, zum anderen kann das in Kap. 6 vorgestellte pDWH-Schema für BPMN-Workflows angewendet werden. Die Grundlagen der BPMN sind in Kap. 6.1 eingeführt. Die Regeln zur Ableitung eines initialen *BPMN-Workflowschemas* aus dem *Vorgangs-Ereignis-Schema* (VES) nach Pütz & Sinz (2010) sind in Tab. 7.3 zusammengefasst und werden nachfolgend anhand der Fallstudie *Mobility Provision* erläutert.

SOM-Geschäftsprozessmodell	BPMN-Workflowschema
betriebliches Objekt	<i>Pool</i>
betriebliche Transaktion	Nachrichtenfluss (<i>MessageFlow</i>) zwischen Pools
betriebliche Aufgabe	Aktivität (<i>Activity</i>) und ggf. Ereignis (<i>Event</i>)
objektinternes Ereignis	Sequenzfluss (<i>SequenceFlow</i>) innerh. eines Pools
Pre/Post-Conditions von Aufgaben	<i>Gateway</i> und Sequenzfluss

Tab. 7.3: Transformationsregeln zur Ableitung von BPMN-Workflowschemata aus SOM-Geschäftsprozessmodellen (Pütz & Sinz 2010)

Ausgangspunkt der Ableitung in der Fallstudie bildet das VES aus Kap. 4.3.2. Das resultierende Workflowschema ist in Abb. 7.6 dargestellt. Jedes betriebliche Objekt wird als *Participant* interpretiert und im Workflowschema als *Pool* abgebildet. Im Kontext der Fallstudie *Mobility Provision* sind die Aufgaben der betrieblichen Objekte *Vertriebskanal* und *Auftragsverwaltung* zu automatisieren (vgl. Kap.

4.4). Aufgrund der Abgrenzung des AwS werden für diese Objekte *public processes* abgeleitet. Die *Pools* der verbleibenden Objekte werden lediglich als *Black Box* dargestellt.

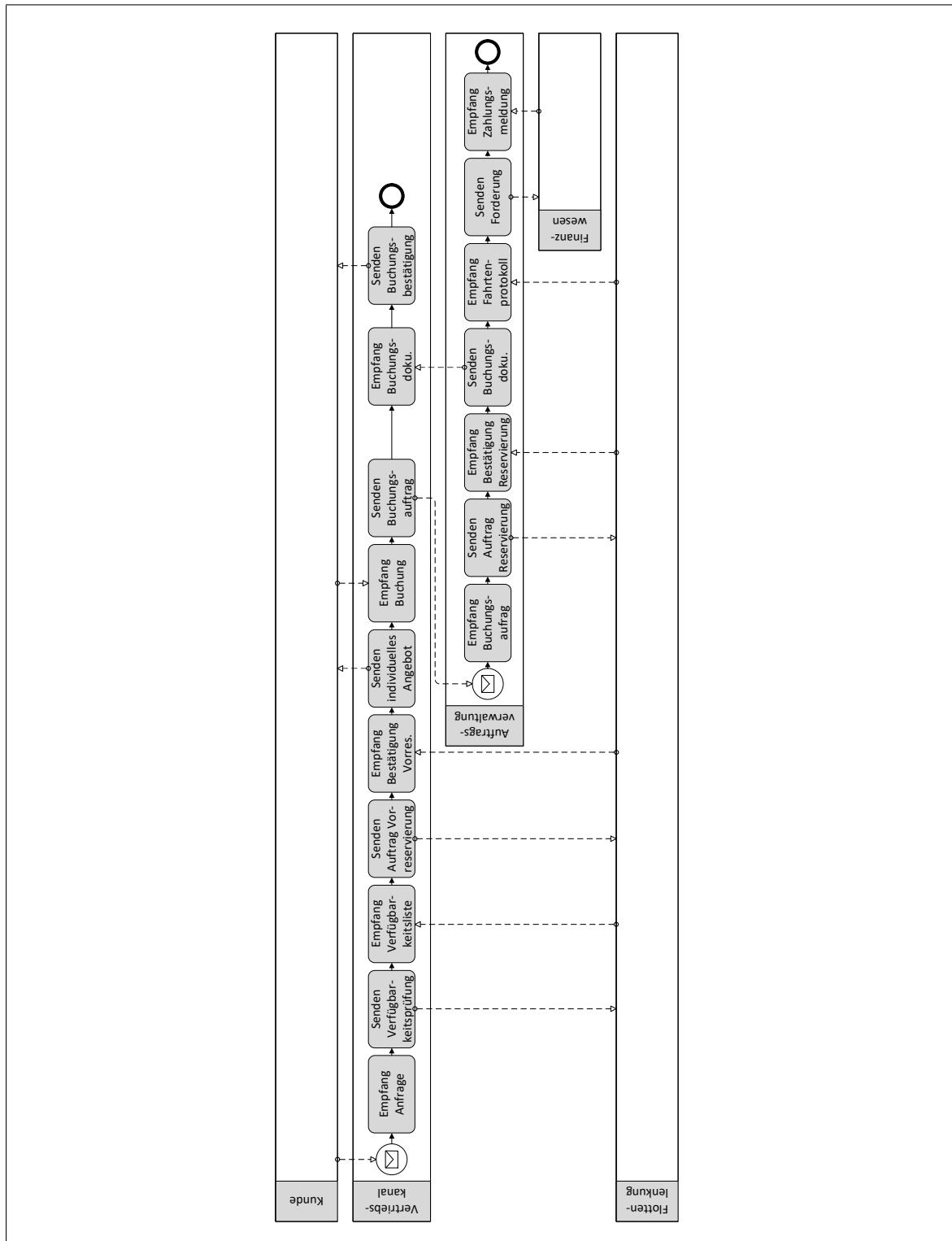


Abb. 7.6: Initiales BPMN-Kollaborationsdiagramm zum Vertriebsprozess

Betriebliche Aufgaben werden als *Aktivität* und ggf. *Ereignis* abgebildet. Als Bei-

spiel dient die Aktivität *Empfang Anfrage*, die zusammen mit einem Start-Ereignis aus der korrespondierenden Aufgabe des VES abgeleitet wird. Betriebliche Transaktionen bilden Nachrichtenflüsse zwischen den Pools bzw. zwischen deren Aktivitäten und Ereignissen. Objektinterne Ereignisse des VES führen zu Sequenzflüssen (z.B. Sequenzfluss des *public process* Vertriebskanal). Sofern betriebliche Aufgaben mit Vor- bzw. Nachbedingungen (*Pre-* bzw. *Post-Conditions*) versehen sind, werden diese in Gateways des abgeleiteten BPMN-Prozess transformiert (Pütz & Sinz 2010).

Die modellbasierte Ableitung von BPMN-Workflows nach Pütz & Sinz (2010) liefert einen *public process*, dessen Spezifikation vor dem Hintergrund der Automatisierung weiter zu verfeinern ist. Bei Ferstl & Sinz (2013, S. 231) werden fünf Schritte zur Konsolidierung des VOS angegeben. Diese sind jedoch auf dessen objektorientierte Eigenschaften (Nachrichten, Attribute, Operatoren) ausgerichtet und nicht direkt auf den BPMN-Workflow übertragbar. Ziel ist es, die Automatisierung von Geschäftsprozessen basierend auf dem komponentenorientierten Paradigma zu realisieren. Aus diesem Grund werden die Konsolidierungsschritte auf der Ebene des *komponentenorientierten Anwendungsmodells* angepasst durchgeführt.

7.4.2 Spezifikation des komponentenorientierten Anwendungsmodells

Nach Birkmeier & Overhage (2009) sind in einer belastbaren Komponentenspezifikation die drei nachfolgenden Modellsichten zu berücksichtigen: (1) In der *Datensicht* werden zu verarbeitende Datenstrukturen spezifiziert. (2) Die *Funktions-sicht* beschreibt das Systemverhalten basierend auf den Datenstrukturen sowie Zusammenhänge zwischen Funktionen. (3) Aus *Prozesssicht* wird die Kombination von Funktionen in Workflows zur Realisierung betrieblicher Aufgaben betrachtet (Birkmeier & Overhage 2009). Diese drei Sichten entsprechen den Sichten (Daten-, Funktions-, Interaktions- und Vorgangssicht) nach Sinz (1999) und Ferstl & Sinz (2013, S. 142f.) (vgl. Kap. 2.1.3), wenn man berücksichtigt, dass in der *Funktionssicht* nach Birkmeier & Overhage (2009) *Funktions-* und *Interaktions-sicht* nach Ferstl & Sinz zusammengefasst sind. Wie auf Ebene des GP-Modells sowie des fachlichen Anwendungsmodells erfolgt auch die komponentenorientierte Systemspezifikation unter Berücksichtigung von Struktur und Verhalten.

Das *operative Komponentenschema* unterstützt *Daten-, Funktions- und Interaktions-sicht*, das *operative Workflowschema* die *Vorgangssicht*. Die Ableitung von Komponenten sowie die Spezifikation von Workflows bilden nicht den Kern des Untersuchungsproblems dieser Arbeit. Auch aus Gründen der Nachvollziehbarkeit wird nachfolgend eine pragmatische Lösung bzgl. der Fallstudie gewählt.

Spezifikation des operativen Komponentenschemas

Zur Spezifikation des komponentenorientierten Anwendungsmodells wird zunächst das *operative Komponentenschema* aus dem fachlichen Anwendungsmodell abgeleitet. Zielsetzung ist die Abgrenzung operativer Komponenten (Datenverwaltung und Vorgangsteuerung) der Referenzarchitektur aus Kap. 5. In Abschn. 7.2.1 werden zwei Ansätze (Ferstl et al. 1997, Friedrich et al. 2016) zur Identifikation von Komponenten auf Basis der SOM-Methodik vorgestellt. Der passende Ansatz bestimmt sich anhand von Anforderungen des konkreten Entwicklungsprojektes (z.B. aufgrund zulässiger Datenredundanz in Komponenten oder der Granularität der Komponenten). Entscheidend für die Anwendbarkeit eines Ansatzes im Rahmen operativ-dispositiver Systeme ist die Nachvollziehbarkeit der Ableitung aus dem Geschäftsprozessmodell. Nachfolgend findet eine pragmatische Alternative zu den genannten Ansätzen Anwendung. Der Ansatz basiert auf der Analyse der objektorientierten Strukturen des fachlichen Anwendungsmodells sowie der Analyse der Transaktionszerlegung auf Aufgabenebene.

Nach Birkmeier & Overhage (2012b) lässt sich die Kohäsion innerhalb einer Komponente maximieren, wenn diese zur Laufzeit gemeinsam genutzte Daten verwaltet. Das *Aggregate Data Model* nach Sadalage & Fowler (2012, S. 14ff.) zur denormalisierten Datenstrukturierung folgt einem ähnlichen Konzept. Der Begriff des *Aggregats* ist dabei dem *Domain Driven Design* nach Evans (2004, S. 126) entnommen. Es wird gefordert, Daten, die in Schreib- oder Leseoperationen gemeinsam behandelt werden, zusammenhängend als Aggregat zu persistieren. Bei Sadalage & Fowler (2012, S. 14ff.) werden dazu objektorientierte Strukturen analysiert. Diese Grundüberlegungen werden auf die Abgrenzung von Komponenten übertragen und erweitert. Zur Identifikation der Kohäsion von Datenobjekttypen werden objektorientierte Strukturmodelle (z.B. UML-Klassendiagramm oder KOS), im Speziellen deren Beziehungstypen auf Hinweise untersucht. Bei Friedrich et al. (2016) werden ebenfalls die Beziehungen innerhalb des KOS betrachtet.

Generalisierungs- und Aggregationsbeziehungen werden bzgl. der Komponentenabgrenzung stärker gewichtet als Assoziationsbeziehungen. Ähnlich wird bei Jain et al. (2001) bzgl. der Generalisierungsbeziehungen argumentiert. Klassen einer Vererbungshierarchie sollten innerhalb einer Komponente zusammengefasst werden. Die Verteilung auf mehrere Komponenten führt zu einer höheren Kopplung zwischen diesen. In Tab. 7.4 sind die objektorientierten Beziehungstypen aufgelistet und mit Hinweisen zur Abgrenzung von Komponenten versehen. Das Ergebnis ist jedoch auch vom jeweiligen Anwendungskontext abhängig, so dass der Entwickler in seiner Entscheidung lediglich unterstützt wird, ihm diese jedoch nicht vollkommen abgenommen werden kann.

Beziehungstyp	Hinweise zur Komponentenabgrenzung
Aggregation u. Komposition (<i>is_part_of</i>)	Objekttypen mit einer Teil-Ganzes-Beziehung zeigen eine hohe Kohäsion und sind in einer Komponente gemeinsam zu verwalten.
Vererbung (<i>is_a</i>)	Objekttypen in einer Generalisierungsbeziehung sollten aufgrund dieser in einer Komponente gemeinsam verwaltet werden.
Assoziation (<i>interacts_with</i>)	Für Assoziationen ist die gemeinsame Verwaltung kontextabh. zu diskutieren. Existenzabhängigkeiten oder die Richtung der Beziehung können Hinweise liefern.

Tab. 7.4: Kriterien zur Komponentenabgrenzung in objektorientierten Strukturen

Aufgrund seiner Einbettung in die SOM-Methodik können anhand des KOS zusätzliche Hinweise zur Identifikation von Komponenten abgeleitet werden. Durch die Visualisierung von Existenzabhängigkeiten kann innerhalb des KOS zwischen Stamm- und Transaktionsdaten unterschieden werden. Existenzunabhängige Objekttypen (LOT, OOT) stehen als Stammdaten mehreren Geschäftsprozessen zur Verfügung und werden als eigenständige, wiederverwendbare Komponenten verstanden. In Abb. 7.7 ist die Kartierung von Komponenten innerhalb des KOS der Fallstudie *Mobility Provision* dargestellt. Als wiederverwendbare *Datenkomponenten* zur Verwaltung von Stammdaten ergeben sich Komponenten für *e-Car-Daten*, das *Mietmodell*, für *Kundendaten* sowie für *Vertriebsdaten*. Letztere umfasst aufgrund von Generalisierungs- und Aggregationsbeziehungen Informationen zu den verschiedenen Typen von Vertriebskanälen (*Callcenter*, *e-Shop*) sowie von *Callcenter-Mitarbeitern*.

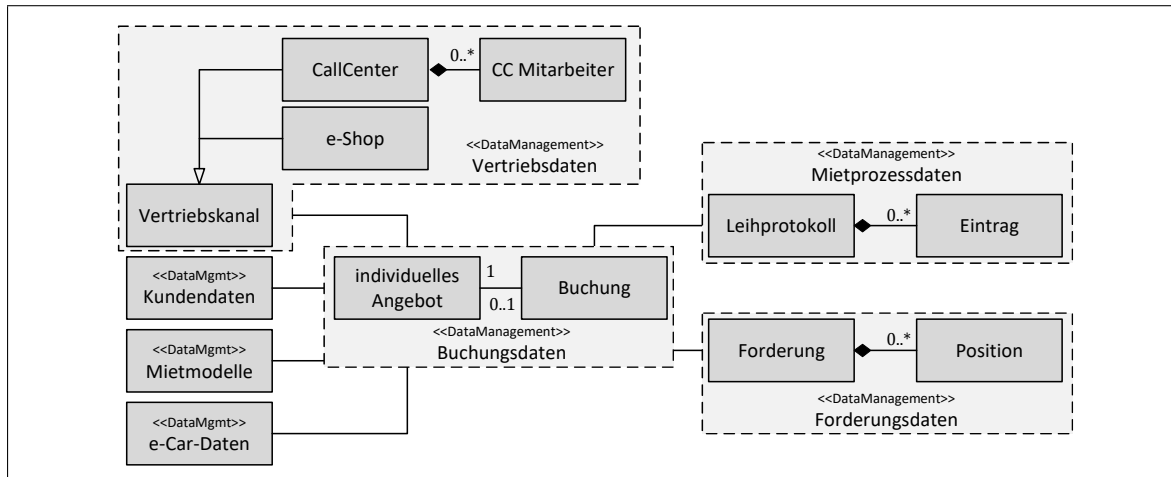


Abb. 7.7: Komponentenabgrenzung im KOS der Fallstudie *Mobility Provision*

Existenzabhängige TOTs stellen i.d.R. GP-spezifische Objekttypen dar. Zur Identifikation von Komponenten innerhalb der TOT-Strukturen wird zusätzlich der Zerlegungsbaum der GP-Transaktionen analysiert. Für die Fallstudie *Mobility Provision* wird die Transaktionszerlegung aus Kap. 4.3.1 herangezogen. Das Ergebnis ist in Abb. 7.8 zusammengefasst und wird nachfolgend erläutert. Durch die Anwendung von Regel (7) der Transaktionszerlegung (Abb. 4.2 in Kap. 4.1) auf Durchführungstransaktionen wird die Homogenisierung zusammengesetzter Leistungen erreicht. Die Übertragung resultierender Teilleistungen erfolgt anhand sequentieller bzw. paralleler Durchführungstransaktionen (Ferstl & Sinz 2013, S. 211). Eine daraus resultierende homogenisierte Leistung sowie zugehörige Lenkungsnachrichten besitzen eine starke Kohäsion und können Hinweise zur Komponentenabgrenzung geben. Durch Zerlegung der Transaktion *Nutzung_Mobilität* in *Mobilitätsleistung* und *Mobilitätsabrechnung* entstehen zwei homogene Teilleistungen. Eine Kartierung innerhalb des KOS (Abb. 7.7) ergibt zwei Komponenten zur Verwaltung der *Mietprozessdaten* sowie der *Forderungsdaten*. Beide Komponenten haben einen abgrenzbaren fachlichen Fokus und hohe Kohäsion bzgl. der verwalteten Daten (Aggregationsbeziehung). Die Objekttypen *Leihprotokoll* und *Eintrag* werden zur Dokumentation von der Anforderung eines *e-Cars* bis hin zur Rückgabe stets gemeinsam verwendet. Eine vergleichbare Argumentation kann auch auf die *Forderung* geführt werden. Regel (7) findet auch bei der Vereinbarung *Buchung der Mobilität* Anwendung. Da die Buchung jedoch auf einem individuellen Angebot beruht, werden beide Datenobjekte in der Komponente *Buchungsdaten* zusammengefasst.

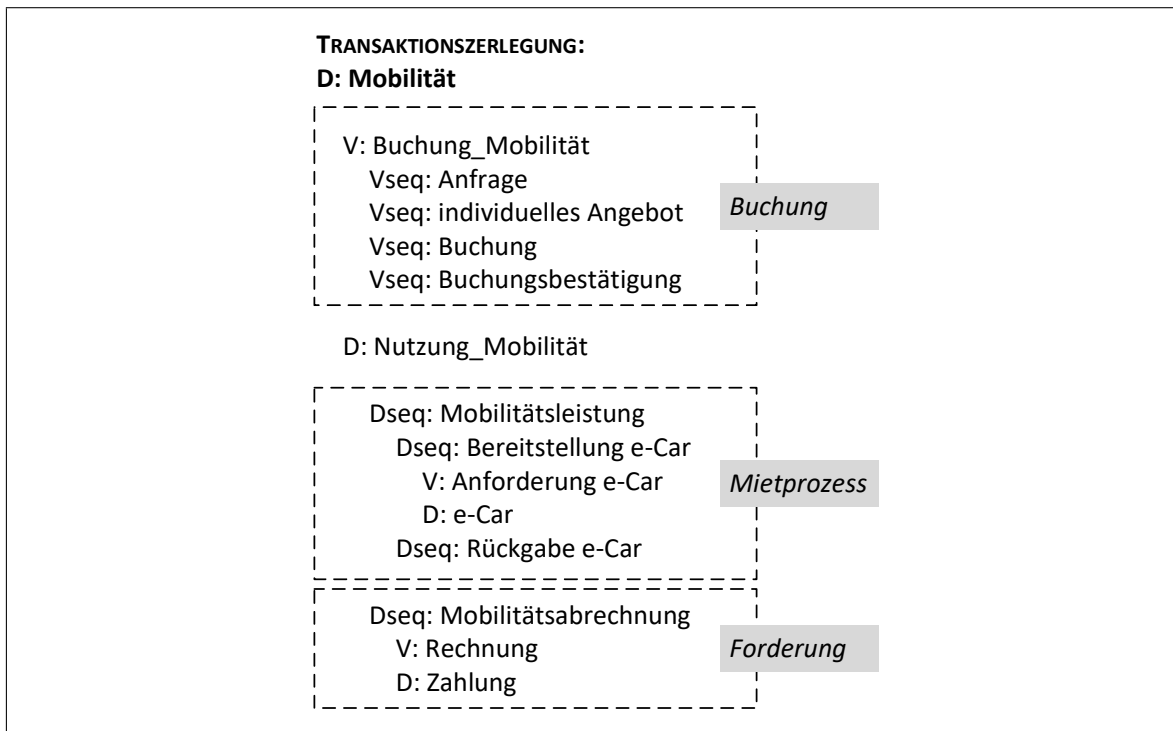


Abb. 7.8: Komponentenabgrenzung im Zerlegungsbaum der Fallstudie *Mobility Provision*

Abb. 7.7 zeigt die *Datensicht* auf die Komponentenspezifikation. Dabei sind die Datenstrukturen im Gesamtzusammenhang sowie deren Verteilung auf *Datenkomponenten* dargestellt. Durch Parametrisierung einer *Datenkomponente* mit einem Ausschnitt des Datenschemas erhält diese einen fachlich abgrenzbaren Fokus und kann als *Fachkomponente* (vgl. Kap. 2.3.1) verstanden werden. Die *Funktionssicht* auf die Komponente ist im Wesentlichen durch die Operatoren ihrer Schnittstelle beschrieben. Diese ergeben sich zum einen aus den Operatoren der verwalteten KOTs (siehe Abschn. 7.4.1 - Konsolidierung des KOS). Zum anderen können sich weitere Operatoren aus der Nutzung der Komponente in übergeordneten Workflows (Verhaltenssicht, nachfolgender Abschnitt) ergeben. In der *Interaktionssicht* werden die Kommunikationsbeziehungen zwischen den Komponenten zur Datenverwaltung sowie zur Vorgangsteuerung beschrieben.

Spezifikation des operativen Workflowschemas

Neben dem operativen Komponentenschema ist das darauf abgestimmte Workflowschema als Teil des komponentenorientierten Anwendungsmodells zu spezifizieren. Hierzu wird das in Abschn. 7.4.1 abgeleitete initiale BPMN-Workflowschema aufgegriffen. Der dargestellte *public process* wird um die Innensicht hin

zum *private process* erweitert. Nachfolgend werden die zugehörigen Entwurfsschritte vorgestellt und auf das Fallbeispiel *Mobility Provision*, im Speziellen auf das betriebliche Objekt *Vertriebskanal* angewendet:

(1) Spezifikation der Automatisierungsgrade: Für die Aktivitäten des initialen BPMN-Workflowschemas werden Automatisierungsgrade festgelegt. Zu entscheiden ist, ob die Aktivitäten durch das AwS voll- oder teilautomatisiert zu unterstützen sind oder ob diese nicht-automatisiert durch personelle Aufgabenträger ausgeführt werden. Erstere können als *Sub-Process* oder *CallActivity* zur weiteren Spezifikation der Innensicht gekennzeichnet werden. Nicht-automatisierbare Aktivitäten werden als *Manual Task* deklariert. Im Gegensatz zu den Konsolidierungsschritten des VOS bei Ferstl & Sinz (2013, S. 231) verbleiben nicht-automatisierbare Aktivitäten im Workflowschema. So können diese zumindest aus Außensicht in die Prozessanalyse einbezogen werden. Für das Fallbeispiel *Mobility Provision* wird für die Aufgaben der betrieblichen Objekte *Vertriebskanal* und *Auftragsverwaltung* volle Automatisierbarkeit angestrebt (vgl. Kap. 4.4). Die Aktivitäten des betrieblichen Objektes (*Pool*) *Vertriebskanal* werden in Abb. 7.9 durchweg als *Sub-Prozess* gekennzeichnet und sind auch aus Innensicht nachfolgend zu spezifizieren.

(2) Prozessauswahl: Die Spezifikation von Workflows zur Ablaufsteuerung erfolgt separat für die einzelnen betrieblichen Objekte. Jeder für das AwS relevante Prozess wird isoliert betrachtet und ist als solcher Gegenstand der Spezifikation des *private process* in nachfolgenden Entwicklungsschritten. Die übrigen Teilnehmer (betriebliche Objekte) werden, sofern diese mit dem betrachteten Prozess Nachrichten austauschen, als *Black Box* in die Spezifikation übernommen. Betriebliche Objekte ohne Nachrichtenbeziehungen fallen aus der Spezifikation heraus. Für das komponentenorientierte AwS der Fallstudie *Mobility Provision* sind die betrieblichen Objekte *Vertriebskanal* und *Auftragsverwaltung* relevant. Die Spezifikation des *private process* erfolgt nachfolgend beispielhaft für den Ablauf des Teilnehmers *Vertriebskanal*. Der zugehörige Pool mit Innensicht auf den Prozess (*public process*) wird zur Spezifikation übernommen (Abb. 7.9). Pools mit Nachrichtenflüssen von bzw. zum *Vertriebskanal* (*Kunde*, *Auftragsverwaltung*, *Flotte*) werden als *Black Box* beschrieben.

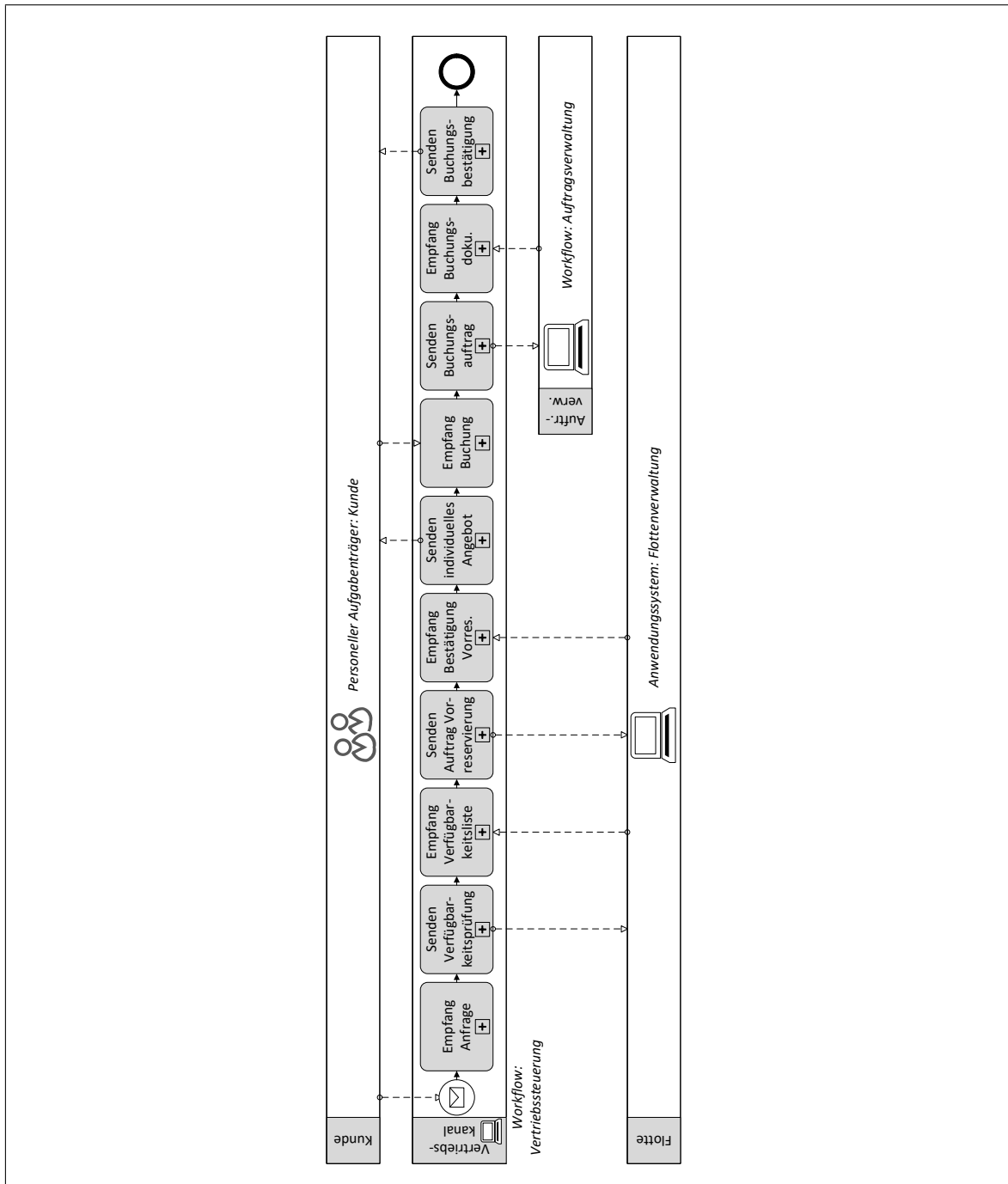


Abb. 7.9: Konsolidiertes BPMN-Kollaborationsdiagramm des Vertriebskanals

(3) Aufgabenträgerzuordnung: Basierend auf der Ableitung nach Pütz & Sinz (2010) werden *Participants* als betriebliche Objekte verstanden, deren Verhalten durch die zugehörigen Prozesse beschrieben wird. Zur weiteren Verfeinerung der Workflows hin zur Automatisierung ist es notwendig, in Abstimmung mit der Modellebene, *Participants* als Aufgabenträger zu verstehen. Als Aufgabenträger können benachbarte AWS, Komponenten des zu spezifizierenden AWS oder personelle Aufgabenträger, wie Arbeiter oder Kunden, verstanden werden. Die Prozes-

se der betrieblichen Objekte werden Aufgabenträgern zugeordnet und ausgerichtet auf diese weiter verfeinert. Die Veränderung der Sichtweise ist konform zum Verständnis der BPMN-Spezifikation. Eigenschaft eines *Participants* ist die Verantwortlichkeit zur Ausführung eines Prozesses bzw. Prozessabschnittes (OMG 2011, S. 114). In Abb. 7.9 ist ebenfalls die Aufgabenträgerzuordnung der Fallstudie dargestellt. Der *Kunde* stellt als Nutzer des Systems einen personellen Aufgabenträger dar. Die Verwaltung der e-Car-Flotte wird von einem benachbarten AWS realisiert, das nicht Gegenstand der Entwicklungsaufgabe ist. Für den separat zu spezifizierenden Workflow zur Durchführung der Auftragsverwaltung ist eine eigenständige Komponente zur Vorgangssteuerung vorgesehen.

(4) Spezifikation des Lösungsverfahrens für automatisierbare Aktivitäten: Für die zu automatisierenden Aktivitäten des Workflows ist die Innensicht (Lösungsverfahren) zu spezifizieren. Hierzu wird ein hierarchisches Vorgehen in Anlehnung an Silver (2011, S. 67) vorgeschlagen. Durch das hierarchische Vorgehen wird zum einen die Betrachtung des Prozesses auf unterschiedlichen Abstraktionsebenen (z.B. abstrahierte Gesamtsicht auf den Prozess vs. Detailbetrachtung einzelner Aktivitäten) (Silver 2011, S. 37f.), aber auch die Nachvollziehbarkeit der Entwicklung ermöglicht. Hierzu stellt die BPMN zwei grundsätzliche Alternativen bereit: Die zu automatisierende Aktivität wird als *Sub-Process* interpretiert und die Spezifikation des übergeordneten Workflows um die Innensicht erweitert. Alternativ kann eine Aktivität auch als *CallActivity* interpretiert werden. Die Spezifikation des Lösungsverfahrens der *CallActivity* stellt einen eigenständigen, wiederverwendbaren Prozess dar, der aus dem übergeordneten Workflow heraus aufgerufen wird (Silver 2011, S. 38f.). In beiden Fällen ergeben sich während der Spezifikation der Innensicht neue Nachrichtenflüsse mit *Datenkomponenten*, die ebenfalls als *Black-Box-Participant* in das Schema aufgenommen werden. In Abb. 7.10 ist exemplarisch die hierarchisch verfeinerte Spezifikation des Sub-Prozesses *Empfang Verfügbarkeitsliste* dargestellt, die bereits im Kontext des pDWH-Schemas (Abb. 6.4) vorgestellt wird. Durch Spezifikation der Innensicht des Sub-Prozesses ergibt sich der Nachrichtenaustausch mit der Komponente zur Verwaltung der *e-Car-Daten*, die als Teilnehmer in das Kollaborationsdiagramm aufgenommen wird.

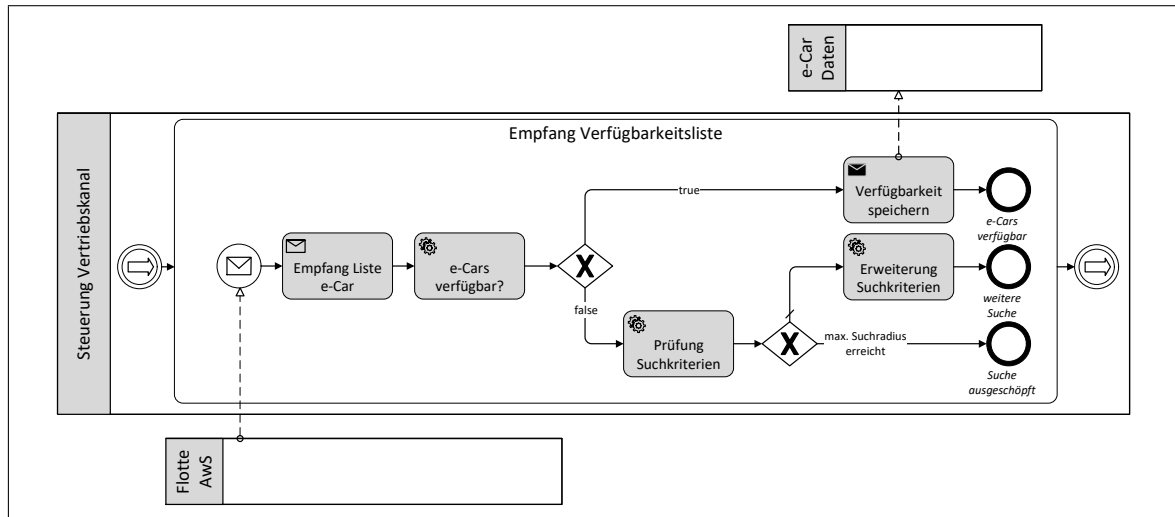


Abb. 7.10: Innensicht des Sub-Prozesses *Empfang Verfügbarkeitsliste*

(5) **Abgrenzung von Komponenten zur Vorgangssteuerung:** Zur Ausführung der *private processes* sind Komponenten zur Vorgangssteuerung zu parametrisieren. Die Verteilung eines Workflows auf Komponenten ist nicht Kern der Arbeit und wird nicht vertieft betrachtet. Zur horizontalen Abgrenzung von Vorgangskomponenten kann beispielsweise auf die Arbeit von Krücke & Sinz (2011) verwiesen werden. Die vorgeschlagene Abgrenzung basiert auf der Anwendung von *Message Exchange Patterns* (W3C 2007). Alternativ können die verschiedenen Hierarchiestufen der Workflow-Spezifikation (v.a. *CallActivities*) Hinweise zur Bildung von Komponenten liefern. Aus Gründen der Nachvollziehbarkeit wird im Rahmen des Fallbeispiels auf die Aufteilung in mehrere Vorgangskomponenten verzichtet.

7.5 Entwicklung des geschäftsprozessorientierten Data-Warehouse-Systems

Ausgehend von der Spezifikation des operativen Systems erfolgt die Entwicklung des pDWH-Systems, im Speziellen des *angebotsorientierten* Teilsystems. Hierzu wird in Abschn. 7.5.1 die Ableitung des AT-spezifischen DWH-Modells sowie in Abschn. 7.5.2 die Einordnung in den GP-Kontext beschrieben.

7.5.1 Spezifikation des aufgabenträgerspezifischen Data-Warehouse-Schemas

Zur Spezifikation des aufgabenträgerspezifischen DWH-Schemas (*AT-DWH-Schema*) finden die Artefakte aus Kap.6, das generische pDWH-Schema sowie sei-

ne domänenspezifische Erweiterung, in der Entwicklungsmethodik Anwendung. Zielsetzung des AT-DWH-Schemas ist es, die operative Systemspezifikation bzw. deren Instanzen prozessorientierten Analysen zugänglich zu machen. Anhand des generischen pDWH-Schemas wird die hierarchisch entwickelte BPMN-Struktur vom initialen *public process* auf Ebene des fachlichen Anwendungsmodells bis hin zum *private process* (op. Workflowschema) auf Ebene des komponentenorientierten Anwendungsmodells abgebildet. Die Dimension *FlowNode-Schema* mit ihrer rekursiven Dimensionshierarchie ermöglicht die Anwendung multidimensionaler Operatoren entlang dieser Strukturen. In Kap. 6.5 wird zudem die Identifikation domänenspezifischer, fachlicher Dimensionen nach Böhnlein (2001, S. 353ff.) auf Basis objektorientierter Datenstrukturen beschrieben. Das konsolidierte KOS des fachlichen und des komponentenorientierten Anwendungsmodells stellen dazu eine geeignete Grundlage dar. Das AT-DWH-Schema wird somit auf Basis des pDWH-Schemas mit domänenspezifischen Erweiterungen gebildet und aus den Spezifikationen des fachlichem bzw. komponentenorientiertem Anwendungsmodell abgeleitet.

Kennzahlen und Dimensionen des AT-DWH-Schemas dienen zudem der Parametrisierung operativer Datenkomponenten zur angebotsorientierten Datenbereitstellung. Anhand der operativen Strukturen einer Datenkomponente erfolgt die Zuordnung von Dimensionen. Eine Dimension *Geographie* mit den Hierarchiestufen *Ort* und *Region* wird der Komponente zugeordnet, die für die Verwaltung der zugehörigen operativen Kundendaten verantwortlich ist. Die Zuordnung von Basiskennzahlen mit Berechnungsvorschriften erfolgt analog. Zur Bereitstellung und zur Aufbereitung der Ablaufdaten eines Workflows werden *Monitoring-Komponenten* parametrisiert. Die Auf- und Verteilung des AT-DWH-Schemas erfolgt, wie unter Abschn. 5.5 bzw. Abschn.7.3.3 beschrieben, anhand von *Partitionierungs-* und *Allokationsschema*.

7.5.2 Spezifikation des geschäftsprozessorientierten Data-Warehouse-Schemas

Die Einordnung des AT-DWH-Schemas in den GP-Kontext erfolgt zweistufig. In einem ersten Schritt erfolgt die isolierte Betrachtung von Aufgaben sowie die Zuordnung zur AT-Ebene. Im zweiten Schritt werden Aufgaben im Kontext des gesamten Geschäftsprozesses betrachtet und pDWH-basierte Ergebnisse eingeord-

net. Das GP-orientierte DWH-Schema (*GP-DWH-Schema*) stellt dabei eine nicht-materialisierte Interpretation des AT-DWH-Schemas dar. Mit diesem Schritt erfolgt die eigentliche Erweiterung eines pDWH- zu einem gpDWH-System.

Aufgabenorientierte Sicht auf das prozessorientierte DWH-Schema

pDWH-Systeme ermöglichen die Analyse des Verhaltens von Prozessen. Sowohl das struktur- als auch das verhaltensorientierte pDWH-Szenario basieren auf dieser ablauforientierten Betrachtung. Aus diesem Grund stellt auch die verhaltensorientierte Sicht des SOM-Geschäftsprozessmodells, das VES, das zentrale Artefakt zur Einordnung des AT-DWH-Schemas in den Kontext von Geschäftsprozessen dar. Die Zuordnung des AT-DWH-Schemas erfolgt über Vorgänge als zentrale Bausteine des VES. Vorgänge beschreiben die Ausführung betrieblicher Aufgaben (vgl. Kap. 2.1.2). Vorgangstypen, gekoppelt über Ereignisbeziehungen, beschreiben das Verhalten eines Geschäftsprozesses als Vorgangnetz (vgl. Kap. 4.1). Betriebliche Aufgaben werden dabei aus Außensicht betrachtet, die Innensicht mit Lösungsverfahren sowie AT-Zuordnung sind der AT-Ebene zugeordnet. Zur Interpretation des AT-DWH-Schemas werden zunächst einzelne Aufgaben des VES isoliert betrachtet und über deren Innensicht die Verknüpfung zum AT-DWH-Schema aufgezeigt. Hierzu werden die Aufgaben des GP-Modells der höchsten Zerlegungsstufe gewählt, die auch die Grundlage zur Ableitung des initialen fachlichen Anwendungsmodells (KOS, VOS) bei der Entwicklung des operativen Systems darstellt.

In Tab. 7.5 sind die Merkmale der Aufgabenaußensicht anhand der Aufgabe *>D: Verfügbarkeitsliste* (Fallbeispiel *Mobility Provision*) dargestellt. Sachziel der Aufgabe stellt die Überprüfung dar, ob die erhaltene Liste verfügbare e-Cars für einen Kunden enthält. Die Aufgabe des betrieblichen Objektes *Vertriebskanal* ist dabei entsprechend der übergeordneten Formalziele (vgl. Kap. 4.4) zeit- und kosteneffizient durchzuführen. Auslösendes Vorereignis stellt die Übertragung der Liste mit e-Cars aus dem betrieblichen Objekt *Flotte* dar, Nachereignis das Vorliegen einer geprüften Liste mit e-Cars. Das Aufgabenobjekt bilden Attribute zu *Kunde*, zu *Vertriebskanal*, zu *e-Car*, zur *Kundenanfrage* sowie zum *Prüfergebnis*. Die eingeführte Aufgabe dient nachfolgend als Grundlage zur Demonstration der GP-orientierten Interpretation des multidimensionalen AT-DWH-Schemas.

Merkmale Aufgabenaußensicht	Ausprägungen: >D: Verfügbarkeitsliste
Sachziel	Feststellung Verfügbarkeit e-Cars
Formalziel	Zeit- und Kosteneffizienz
Vorereignis	Übertragung Verfügbarkeitsliste
Nachereignis	Liste verfügbarer e-Cars ist eingegangen u. geprüft
Aufgabenobjekt	Kunde, Anfrage, e-Cars, Vertrieb, Prüfergebnis

Tab. 7.5: Merkmale der betrieblichen Aufgabe >D: Verfügbarkeitsliste

Schema sowie Instanzen von Workflows zur GP-Durchführung auf AT-Ebene sowie dabei verarbeitete Datenbestände werden im *AT-DWH-Schema* aufbereitet und zugänglich für multidimensionale Operatoren. Zur fachlichen Analyse werden diese Informationen basierend auf dem Konzept der betrieblichen Aufgabe in den GP-Kontext integriert. Hierzu wird kein neues DWH-Schema abgeleitet, vielmehr wird das *AT-DWH-Schema* aus GP-Sicht interpretiert. Angelehnt an das Konzept der Sichten (*Views*) relationaler Datenbanksysteme (z.B. Kemper & Eickler 2013, S. 139ff., Ferstl & Sinz 2013, S. 399ff.) werden Sichten auf die DWH-Strukturen der AT-Ebene als virtuelles DWH-Schema spezifiziert. Eine Sicht stellt dabei nur einen Ausschnitt des gesamten AT-DWH-Schemas dar. Reichweite sowie inhaltliche Aufbereitung der Sicht ergeben sich anhand der betrachteten betrieblichen Aufgabe. Sichten können entweder angebotsorientiert in Form von Metadaten für potenzielle Nutzer bereitgestellt oder aufgrund konkreter Nachfragen (nachfrageorientiert) erstellt werden.

Die Spezifikation der Sichten basiert im Wesentlichen auf OLAP-Operatoren wie *Drill-Down*, *Roll-Up* oder *Selektion* (vgl. Kap. 3.3.2). Zur Analyse des Lösungsverfahrens ist ein *Drill-Down* in die multidimensionale Datenstruktur der AT-Ebene vorzunehmen. Entscheidende Bedeutung kommt dabei der Realisierung von Aufgaben bzw. der Abbildung dieser auf Aktivitäten der BPMN-Workflows zu. Den Ableitungsregeln aus Abschn. 7.4.1 folgend besteht eine 1-zu-1-Beziehung und stellt den Standardfall dar. Konsolidierungsschritte auf AT-Ebene können jedoch auch zur Aufspaltung von Aktivitäten führen. Dies ist bei der Zuordnung von Aktivitäten zum Lösungsverfahren einer Aufgabe zu berücksichtigen. Anhand der dokumentierten Entwicklungsschritte ist eine entsprechende Zuordnung möglich. Der Grad der Automatisierung von Aktivitäten stellt einen limitierenden Faktor v.a. für *Drill-Down*-Operationen dar. Wie in Abschnitt 7.4.1 beschrieben, wer-

den automatisierbare Aktivitäten als Sub-Prozess verstanden und hierarchisch verfeinert, während manuelle Aktivitäten lediglich aus Außensicht betrachtet werden. Entsprechend ist die Analyse der Innensicht nur für automatisierte Aktivitäten möglich.

Die Gestaltung der Sichten zur Bewertung betrieblicher Aufgaben ist in hohem Maße durch die konkrete Aufgabe bestimmt. Nachfolgend können anhand der Merkmale der betrieblichen Aufgabe jedoch Hinweise zur Gestaltung der Sichten abgeleitet werden:

- **Ziele (Sach- u. Formalziele):** Eine zentrale Fragestellung der Aufgabenanalyse ist, ob mit Durchführung einer Aufgabe das Sachziel unter Einhaltung der Formalziele erreicht wird. Die Erreichung der Sachziele bzw. der Zielerreichungsgrad der Formalziele einer Aufgabe beeinflusst sowohl die Leistungserstellung als auch die Erreichung der Formalziele des übergeordneten Geschäftsprozesses. Zur Bewertung der Zielerreichung von Aufgaben können in Anlehnung an Böhnlein (2001, S. 351ff.) Kennzahlen aus assoziierten Zielen abgeleitet werden. Die Kennzahlen stellen i.d.R. *abgeleitete Kennzahlen* (vgl. Kap. 3.3.1) dar, die sich aus Kennzahlen des AT-DWH-Schemas berechnen. Sofern zusätzliche Kennzahlen benötigt werden, ist das AT-DWH-Schema zu erweitern.
- **Aufgabenobjekt:** Das Aufgabenobjekt kann als Ausschnitt des operativen Datenschemas verstanden werden. In Kap. 6.5 ist die Ableitung fachlicher Dimensionen aus objektorientierten Strukturen nach Böhnlein (2001, S. 353ff.) beschrieben. Anhand des Aufgabenobjektes bzw. des dadurch bestimmten Ausschnitts des operativen Datenschemas können fachliche Dimensionen im AT-DWH-Schema identifiziert werden.
- **Vor- u. Nachereignisse:** Durch auslösende Vor- und resultierende Nachereignisse ist eine betriebliche Aufgabe aus ablauforientierter Sicht abgegrenzt. Durch diese Grenzen bzw. deren Abbildung auf AT-Ebene kann die Selektion von Instanzen innerhalb des AT-DWH-Schemas unterstützt werden. Aktivitäten können so beispielsweise dem Lösungsverfahren einer Aufgabe zugeordnet oder von diesem abgegrenzt werden.

Die Anwendung des Konzeptes *aufgabenorientierter Sichten* wird anhand der in Tab. 7.5 vorgestellten Aufgabe >D: *Verfügbarkeitsliste* demonstriert. Zunächst ist

die Abbildung der Aufgabe aus dem VES auf die AT-Ebene zu identifizieren. Im konkreten Fall liegt eine 1-zu-1-Abbildung auf die Aktivität *Empfang Verfügbarkeitsliste* des BPMN-Schemas aus Abb. 7.9 (NodeID = 3 im Star-Schema in Abb. 6.6) vor. Das Lösungsverfahren der Aufgabe kann über die Innensicht des Sub-Prozesses *Empfang Verfügbarkeitsliste* analysiert werden.

Abb. 7.11 zeigt ein Beispiel zur Analyse der Formalziele. Die Bewertung des Formalziels der *Zeiteffizienz* (siehe Tab. 7.5) erfolgt durch die Kennzahl *durchschnittliche Laufzeit*. Diese kann aufgrund der generischen Kennzahlen Anzahl der Ausführungen (n) und Laufzeit (t) des pDWH-Schemas der AT-Ebene berechnet werden. Die Kennzahl soll für Knoten der höchsten Hierarchiestufe (im Falle rekursiver Strukturen) berechnet werden. Zur Einschränkung auf den zur Aufgabe >D: *Verfügbarkeitsliste* gehörenden Sub-Prozess wird ein Filter definiert. Dieser selektiert Aktivitäten, die dem Sub-Prozess (*parent = 3*) untergeordnet sind. Für weitergehende Analysen erscheinen die Dimensionen *FlowNode-Sequenz* sowie *FlowNode-Typ* des pDWH-Schemas relevant. Anhand des Aufgabenobjektes kann die fachliche Dimension *Vertriebsart* (vgl. Abb. 6.7) zugeordnet werden. Damit wird ein Vergleich der Laufzeit getrennt nach Typ des Vertriebskanals (Online-Shop vs. Call Center) ermöglicht. Gestalterische Maßnahmen können gezielt auf einen Vertriebstyp ausgerichtet werden. Durch Berücksichtigung der Dimension *Zeit* wird die Entwicklung der Laufzeiten im historischen Verlauf betrachtet. Die Bewertung des Formalzieles *Kosteneffizienz* kann mit angepasster Kennzahl analog realisiert werden. Die SQL-basierte Implementierung der Sicht ist in Anhang A.2 (Code A.4) angegeben. Sie basiert auf der relationalen Interpretation des pDWH-Schemas aus Kap. 6.4.3.

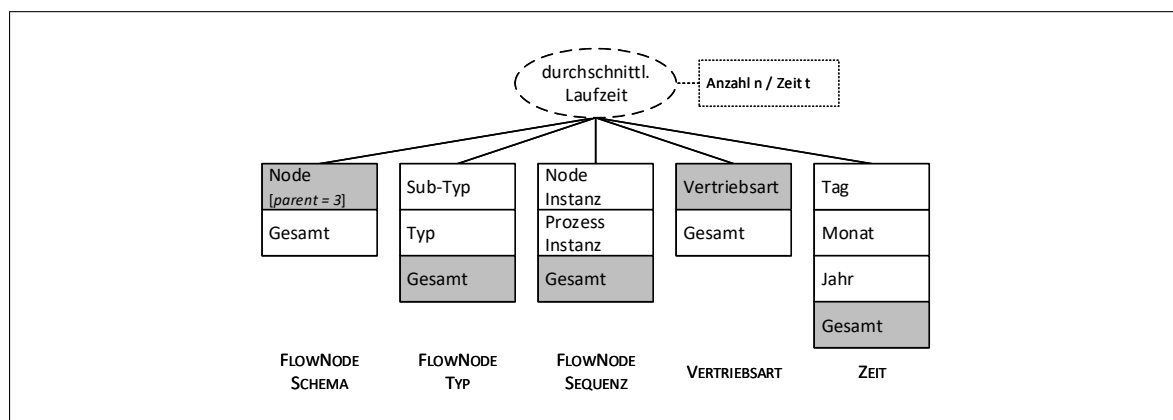


Abb. 7.11: Aufgabenorientierte Sicht – Durchschnittliche Laufzeit

Ein Beispiel zur Analyse der Sachzielerreichung ist in Abb. 7.12, die zugehörige SQL-basierte Implementierung in Anhang A.2 (Code A.5) dargestellt. Wie in Abb. 7.10 dargestellt, kann die Aktivität *Empfang Verfügbarkeitsliste* mit dem Ereignis *e-Cars für den Kunden verfügbar* erfolgreich terminieren. Sofern keine Fahrzeuge verfügbar sind, kann die Suche erweitert werden oder als endgültig gescheitert (*Suche ausgeschöpft*) abschließen. Mit beiden zuletzt genannten Endereignissen wird das Sachziel der erfolgreichen Feststellung verfügbarer e-Cars verfehlt. Die Bewertung der Sachzielerreichung erfolgt über den *Anteil der Nachereignisse* im Verhältnis zur *Gesamtzahl der Ausführungen* des Sub-Prozesses. Die hieraus berechnete Kennzahl *Anteil Nachereignisse* basiert auf der generischen Kennzahl n des AT-DWH-Schemas. Auch sie wird auf Knotenebene (*FlowNode-Schema* mit Filter *NodeID = 3*) gebildet. Für das Ergebnis sind ausschließlich Endereignisse (*FlowNode-Typ*, Filter *subtype = end_event*) relevant. Zur Feststellung lokaler oder regionaler Abweichungen bei der Verfügbarkeit von Fahrzeugen wird auf Basis des Aufgabenobjektes die Dimension *Geographie* in die multidimensionale Sicht aufgenommen und auf Hierarchiestufe *Ort* betrachtet. Analysen über die Dimension *Zeit* ermöglichen hier Einblicke über die Entwicklung der Verfügbarkeit im Zeitverlauf.

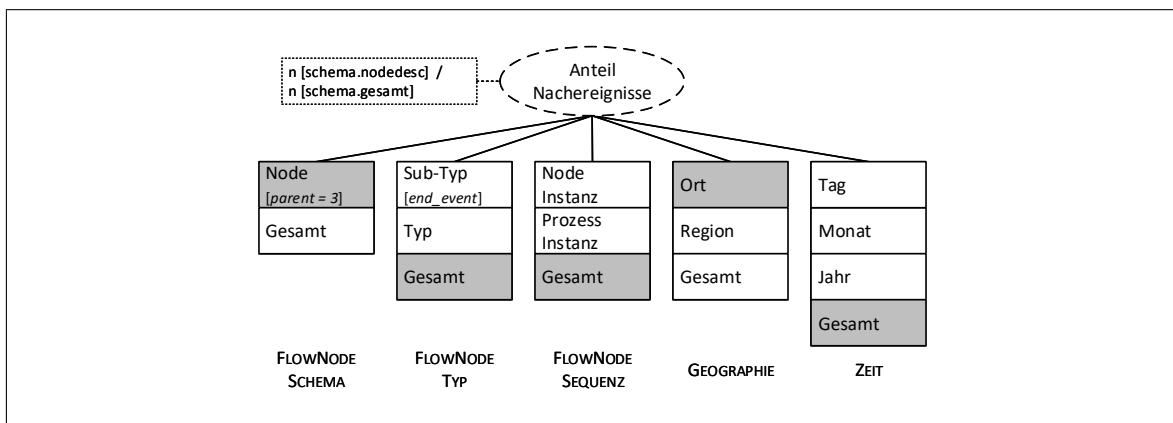


Abb. 7.12: Aufgabenorientierte Sicht – Anteil der Nachereignisse

Geschäftsprozessspezifische Einordnung des DWH-Schemas

Im Weiteren erfolgt die Einordnung *aufgabenorientierter Sichten* in den GP-Kontext. Zunächst erfolgt dies anhand der Integration der betrieblichen Aufgabe in die verhaltensorientierte Sicht, dem VES. Anschließend wird der Bezug zur Leistungserstellung eines Geschäftsprozesses sowie deren Lenkung anhand des strukturorientierten IAS hergestellt.

Verhaltensorientierte Einordnung Wie in Kap. 6.4 ausgeführt, umfasst ein pDWH-Würfel einen Prozess bzw. Teilnehmer eines BPMN-Kollaborationsdiagramms. Übrige Teilnehmer werden aus Außensicht (*Black Box*) betrachtet. Übertragen auf die vorgestellte Entwicklungsmethodik und mit Bezug zu den Ableitungsregeln für BPMN-Prozesse aus dem VES nach Pütz & Sinz (2010) (Abschn. 7.4.1) bedeutet das, dass das Vorgangnetz eines betrieblichen Objektes Gegenstand des GP-orientierten DWH-Schemas wird. Vorgänge dieses Objekts werden im VES über Ereignisse (Vor- bzw. Nachereignisse) verknüpft und beschreiben das GP-Verhalten in Form eines Vorgangnetzes (vgl. Kap. 4.1). Auf Grundlage dieses Verständnisses ergeben sich die nachfolgenden Alternativen zur ablauforientierten Betrachtung einer aufgabenorientierten Sicht im GP-Kontext:

- **Analysen bezogen auf einen Prozessabschnitt:** Ein Prozessabschnitt wird als Teil des VES-Vorgangnetzes eines betrieblichen Objektes zwischen zwei Ereignissen verstanden. Dieser Abschnitt wird Gegenstand der Analyse, wobei der gewählte Pfad (z.B. bei alternativen Pfaden) ohne Relevanz ist.
- **Analysen bezogen auf einen Prozesspfad:** Pfadbezogene Analysen hingegen fokussieren auf einen konkreten Pfad innerhalb eines Prozesses bzw. eines Prozessabschnittes.

In beiden Fällen hängt die Berücksichtigung der aufgabenbezogenen Sichten in übergeordneten GPKennzahlen von der Positionierung der Aufgabe innerhalb des Pfades bzw. Abschnittes ab (z.B. eingeschränkte Summierbarkeit).

Abb. 7.13 zeigt die Einordnung der betrieblichen Aufgabe >D: *Verfügbarkeitsliste* in das VES der Fallstudie *Mobility Provision*. Gegenstand des gpDWH-Schemas ist das Vorgangnetz des betrieblichen Objektes *Vertriebskanal*. Als Beispiel für eine abschnittsbezogene Analyse ist der Prozessabschnitt von der Anfrage verfügbarer e-Cars (Aufgabe V: *Verfügbarkeitsprüfung*>) bis zum Ergebnis bzgl. der Verfügbarkeit (Nachereignis der Aufgabe >D: *Verfügbarkeitsliste*) eingezeichnet. Ziel ist die Ermittlung der durchschnittlichen Zeit ($avg(t)$) und Kosten ($avg(c)$), die, aufgeschlüsselt nach Vertriebsart, pro Instanz für diesen Prozessabschnitt entstehen. Die zugehörige multidimensionale Sicht zur Aufgabe >D: *Verfügbarkeitsliste* ist in Abb. 7.11 exemplarisch eingeführt. Bei einmaliger Durchführung der Aufgaben pro GP-Instanz sowie gleicher Granularität der Kennzahlen (hier bzgl. der Dimension *Vertriebsart*) ergibt sich die durchschnittliche Laufzeit des Prozessabschnitt-

tes durch die Summe der Teilergebnisse. Eine mehrfache Ausführung der zugehörigen Lösungsverfahren erfordert zusätzlich die Berücksichtigung der durchschnittlichen Anzahl der Aufgabendurchführungen pro Prozessinstanz. Wie in Kap. Abb. 6.4 dargestellt, resultiert die mehrfache Ausführung aus einer Schleife, die sich erst auf AT-Ebene im Rahmen der Konsolidierung ergibt.

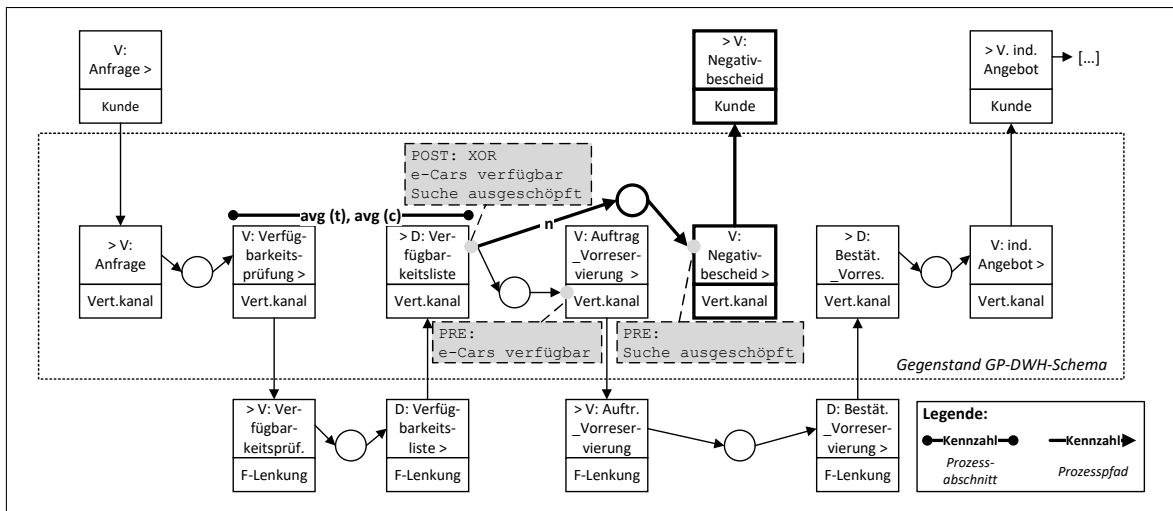


Abb. 7.13: Verhaltensorientierte Einordnung aufgabenorientierter Sichten

Eine pfadbezogene Analyse ist ebenfalls in Abb. 7.13 dargestellt und ermöglicht die Analyse von Normal- und Ausnahmeverhalten des betrieblichen Objektes *Vertriebskanal*. Im Vergleich zu Normalverhalten werden Sach- und/oder Formalziele bei Ausnahmeverhalten nur eingeschränkt erreicht (Ferstl & Sinz 2013, S. 112f.). Die Deklaration von Normal- und Ausnahmeverhalten ist kontextabhängig und erfordert Kenntnisse der Anwendungsdomäne. Der normale Ablauf des dargestellten Ausschnitts der Fallstudie *Mobility Provision* endet nach erfolgreicher Prüfung auf verfügbare e-Cars in einem individuellen Angebot gegenüber dem Kunden. Ausnahmeverhalten greift, sollte kein e-Car für den Kunden zur Verfügung stehen. Zwar wird das Sachziel nicht erreicht, dennoch sollte der Kunde zeitnah per Negativbescheid informiert werden. Der Ablauf von Normal- und Ausnahmeverhalten ist in Abb. 7.13 über Pre- und Post-Conditions beschrieben. Anhand der Analyse der Nachereignisse der Aufgabe >D: Verfügbarkeitsliste (Kennzahl Anteil Nachereignisse) (vgl. aufgabenorientierte DWH-Sicht in Abb. 7.12) kann der relative Anteil von GP-Instanzen mit Ausnahmeverhalten bestimmt werden. Aufgrund des Nachereignisses *Suche ausgeschöpft* wird die Ausführung des Ausnahmeverhaltens (*Versand Negativbescheid*) ausgelöst.

Strukturorientierte Einordnung Im Weiteren wird die verhaltensorientierte Einordnung der aufgabenorientierten DWH-Sichten um Informationen strukturorientierter Sichten (*Lenkungs-* und *Leistungssicht*) auf einen Geschäftsprozess (vgl. Kap. 4.1) ergänzt. Die Grundlage bilden das IAS sowie zugehörige Zerlegungsprotokolle:

- **Bezug zur Leistungssicht:** Die aufgabenorientierte DWH-Sichten werden explizit, ggf. mehrstufig, mit der Leistung eines Geschäftsprozesses in Beziehung gesetzt. Damit kann wesentlich zur Bewertung der betrieblichen Leistungserstellung beigetragen werden.
- **Bezug zur Lenkungsicht:** Die aufgabenorientierten DWH-Sichten werden in den Kontext der Lenkung der Leistungserstellung eingeordnet. Dies ermöglicht die Bewertung der GP-Koordination über das *Verhandlungs-* oder das *Regelungsprinzip* bzw. der damit beschriebenen Phasen.

Mit Lenkungs- und Leistungssicht wird die strukturorientierte Einordnung von abschnitts- bzw. pfadbezogenen Analysen (Verhaltenssicht) ermöglicht. In Gegenrichtung wird aber auch die Identifikation relevanter Prozessabschnitte bzw. -pfade durch die Struktursicht unterstützt.

Abb. 7.14 zeigt den für das betriebliche Objekt *Vertriebskanal* relevanten Ausschnitt des IAS, ergänzt um Informationen der Objekt- bzw. Transaktionszerlegung. Nachfolgend relevante Modellbestandteile sind explizit hervorgehoben. Als Beispiel zur Analyse von Prozessabschnitten wird mit der verhaltensorientierten Einordnung (vorheriger Abschn.) die Analyse der durchschnittlichen Zeit und Kosten, die von der Anforderung bis hin zum Empfang einer Liste mit verfügbaren e-Cars entstehen, beschrieben. Anhand der strukturorientierten Betrachtung auf Basis von Abb. 7.14 wird deutlich, dass es sich dabei um einen Teil der Leistung des betrieblichen Objektes *Flottenlenkung* (*D: Flottenverwaltung*) handelt. Es werden also Kosten und Zeit der Teilleistung *D: Verfügbarkeit* zwischen Vertriebskanal und Flottenlenkung bewertet. Die Koordination erfolgt dabei nicht-hierarchisch nach dem Verhandlungsprinzip. In der Gegenrichtung ermöglicht das IAS mit Zerlegungsinformationen auch die Identifikation zu analysierender Prozessabschnitte. Die Gesamtleistung von *Mobility Provision* besteht in der Bereitstellung von Mobilität (*D: Mobilität*) für Kunden. Das betriebliche Objekt *Vertriebskanal*, so ist in Abb. 7.14 ersichtlich, ist für die Vereinbarung der Leistung

7.6.1 Spezifikation von Data Marts für das strukturorientierte Szenario

Durch das strukturorientierte pDWH-Szenario wird die innovativere und effizientere Gestaltung von Prozessen auf Basis eines DWH-Systems angestrebt. Bei Weske (2012, S. 16) werden *Stakeholder* der Domäne *Geschäftsprozessmanagement* charakterisiert, die potenzielle Nutzer von gpDWH-Systemen darstellen:

- **Chief Process Officer** (Weske 2012, S. 16): Das Aufgabenfeld kann mit der übergeordneten Standardisierung und Harmonisierung von Geschäftsprozessen beschrieben werden. Ferner fällt die Evolution von Geschäftsprozessen vor dem Hintergrund eines sich verändernden Unternehmensumfeldes (z.B. Marktverhältnisse) in den Verantwortungsbereich des *Chief Process Officers*.
- **Process Owner** (Weske 2012, S. 16): Dem *Process Owner* ist die Verantwortung für die korrekte und effiziente Durchführung eines Geschäftsprozesstyps übertragen. Ineffizienzen sind in Abstimmung mit Prozessbeteiligten sowie *Process Designern* zu beseitigen.
- **Process Designer** (Weske 2012, S. 16): *Process Designer* sind für die Modellierung der Geschäftsprozesse in Abstimmung mit dem Geschäftsbereich sowie anderen Stakeholdern verantwortlich. Die Prozessverbesserung in Abstimmung mit dem *Process Owner* ist bereits thematisiert.

Anhand der vorgestellten Stakeholder lassen sich zwei exemplarische Zielsetzungen für das strukturorientierte Szenario des Fallbeispiels *Mobility Provision* ableiten. Zum einen kann der *Chief Process Officer* in seiner Tätigkeit zur Evolution von Geschäftsprozessen aufgrund von Marktveränderungen unterstützt werden. Zur Stärkung der Wettbewerbsposition stellt die Neugestaltung der Vertriebskanäle ausgerichtet auf Kundengruppen eine Möglichkeit dar. Auf Basis des gpDWH-Systems können Kundengruppen sowie deren Verhalten während der Vereinbarungsphase analysiert und entsprechende Maßnahmen abgeleitet werden. Die Steigerung der Prozesseffizienz stellt eine weitere mögliche Zielsetzung zur Nutzung des gpDWH-Systems im Rahmen der Fallstudie dar. Analyse und daraus resultierende Entwurfsentscheidungen bzgl. der Prozesseffizienz erfolgen durch Kooperation von *Process Owner* und *Process Designer*. Die Analyse von Abschnitten eines GP mit der Möglichkeit zur Fokussierung auf einzelne Aufgaben und deren

Lösungsverfahren kann die Identifikation von ineffizienten Prozessbestandteilen unterstützen. Für beide Zielsetzungen bzw. Nutzergruppen kann ein Informationsbedarf abgeleitet werden, der über nachfrageorientierte Data Marts zu decken ist.

Mit der Identifikation von *Data Marts* sind zwei Aspekte der Komponentenarchitektur aus Kap. 5.3.3 festzulegen:

1. **Spezifikation der Data-Mart-Komponenten** (vgl. Kap. 5.4.3): Das *Data-Mart-Schema* wird anhand des nutzerspezifischen Informationsbedarfs aus dem angebotsorientierten, globalen DWH-Modell abgeleitet. Je nach Anforderung wird das *Data-Mart-Schema* aus dem AT-DWH-Schema oder auf Basis GP-orientierter Sichten (GP-DWH-Schema) gebildet. Bei der Abbildung auf die Data-Mart-Komponenten sind u.a. eine geeignete Technologie zur Datenverwaltung sowie eine Historisierungsstrategie festzulegen (*Komponenten-DM-Schema*).
2. **Spezifikation der Komponenten zur Datenintegration** (vgl. Kap. 5.4.2): Zur Versorgung der Data-Mart-Komponenten ist die Integration des *Data-Mart-Schemas* über das *Integrationsschema* festzulegen. Als Metadaten werden das zu integrierende Zielschema mit Aggregationsvorschriften, die Ziel-Data-Marts sowie die Quell-Komponenten benötigt. Letztere können anhand des *Verteilungsmodells* bestimmt werden. Zudem sind Ereignisse als Auslöser der Datenbereitstellung zu spezifizieren (vgl. Kap. 5.3.3).

Data-Mart-Strukturen zur Deckung des Informationsbedarfs des *Chief Process Officers* können anhand einer GP-orientierten Sicht (vgl. Abschn. 7.5.2) abgeleitet werden. Ausgehend von der strukturorientierten Betrachtung kann innerhalb der Verhaltenssicht die *Vereinbarungsphase* als Prozessabschnitt abgegrenzt werden. Kennzahlen (*erfolgreiche Abschlüsse, durchschnittliche Bearbeitungsdauer*) sind auf Granularität von *Altersgruppe* sowie *Art des Vertriebskanals* für diesen Prozessabschnitt vorzuhalten. GP-orientierte Sichten sowie dazu gehörige Berechnungsvorschriften sind als Parameter an die Komponenten zur Datenintegration zu übergeben. Dimensionen und Kennzahlen der fachlichen Domäne werden für den vorliegenden Anwendungsfall im Wesentlichen durch die op. Datenkomponenten *Kunden-, Buchungs- sowie Vertriebsdaten* bereitgestellt. Die Aktualität des zu

analysierenden Datenbestandes wird als unkritisch betrachtet, so dass nächtliche Integrationsprozesse ausreichend sind. Zur Analyse der Prozesseffizienz ist es notwendig, Daten auf Granularität der Knoten des Lösungsverfahrens bereitzustellen. Mit GP-orientierten Sichten (z.B. zur Vereinbarungsphase) als Einstieg wird es durch multidimensionale Operatoren (z.B. Drill-Down) ermöglicht, Analysen von GP-Abschnitten über einzelne Aufgaben bis hin zu deren Lösungsverfahren auf AT-Ebene zu verfeinern. Ineffiziente Prozessabschnitte können sukzessive eingegrenzt werden. Ähnlich wie die Dimension *Vertriebsart* (vgl. Beispiel aus Abb. 7.11) können Dimensionen der betrieblichen Domäne ergänzende Informationen zur Aufdeckung von Ineffizienzen liefern. *Process Owner* und *Process Designer* werden im strukturorientierten Szenario umfassend bei Analysen auf Aufgaben- und AT-Ebene unterstützt.

In der Anwendung der Referenzarchitektur in Kap. 5.3.4 wird eine Data-Mart-Komponente *Vertriebsgestaltung* thematisiert. Diese kann aufgrund vergleichbarer Inhalte zur Deckung der Nachfrage beider *Stakeholder* parametrisiert werden. Auch die Bildung jeweils einer Komponente ist aufgrund heterogener Granularitätsanforderungen vorstellbar.

7.6.2 Spezifikation von Data Marts für das verhaltensorientierte Szenario

Das verhaltensorientierte pDWH-Szenario ist, wie in Kap. 3.5.2 beschrieben, auf die aktive Steuerung von Prozessinstanzen ausgerichtet. Auf Architekturebene erfordert dies eine engere Integration der zugehörigen Systeme (*operatives, pDWH- u. EU-System*) mit bidirektionalem Datenfluss zwischen operativen und dispositiven Teilsystemen (vgl. Anforderung A4, Kap. 5.1). Die Identifikation sowie die Spezifikation von Data Marts für das verhaltensorientierte Szenario kann nur unter Berücksichtigung der Einbettung in das Gesamtsystem erfolgen. Diesbezüglich werden nachfolgend zunächst Spezialisierungen des verhaltensorientierten Szenarios identifiziert. Auf Basis einer Detaillierung der Fallstudie *Mobility Provision (Angebotserweiterung)* wird die Spezifikation verhaltensorientierter Data Marts erläutert.

Spezialisierungen des verhaltensorientierten DWH-Szenarios

Das verhaltensorientierte Szenario kann phasenorientiert über den *OpDW-Cycle* nach Russom (2011) beschrieben werden, dessen Phasen in Kap. 3.5.3 eingeführt werden. In Abb. 7.15 ist der *OpDW-Cycle* mit der Referenzarchitektur aus Kap. 5.3.3 überlagert. Dabei wird deutlich, dass das operative Teilsystem sowohl dessen Quelle als auch Ziel darstellt. *Recognize* beschreibt das Erkennen von Ereignissen im operativen Ablauf. Durch das *pDWH-System* werden relevante Informationen zur Ereignisanalyse aufbereitet (*capture*) und für das *EU-System* zur Analyse (*analyze*) bereitgestellt. Das Ergebnis (teil-)automatisiert getroffener Entscheidung wird schließlich als Reaktion (*react*) an die Ereignisquelle, das op. Teilsystem zurückgegeben.

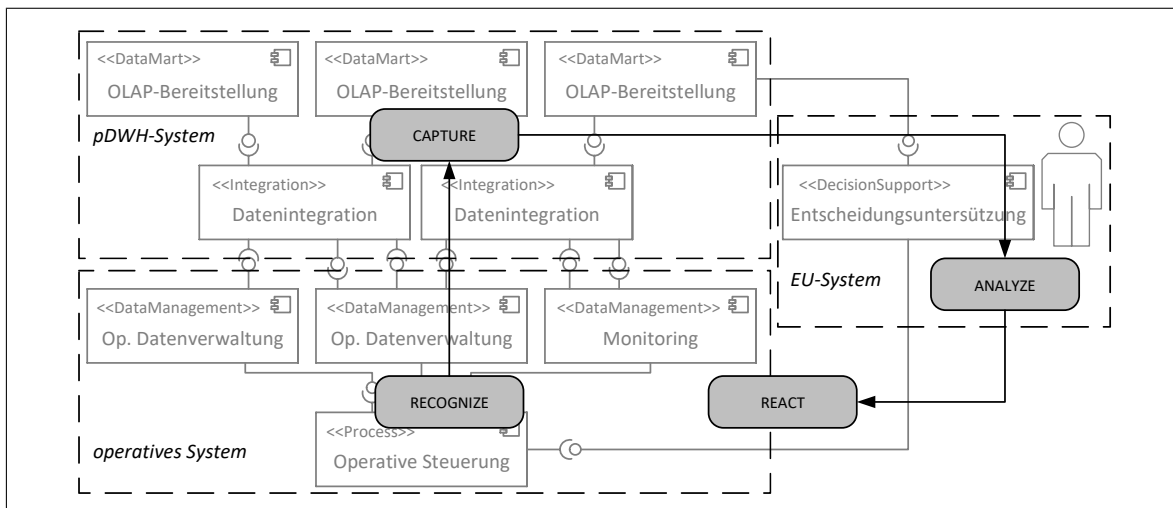


Abb. 7.15: Überlagerung von OpDW-Kreislauf und Referenzarchitektur

Identifikation und Spezifikation von Data Marts sind wesentlich von der Einbettung in diesen Zyklus beeinflusst. Anhand des Instanzbezuges von Quelle (*recognize*) und Ziel (*react*) können vier Spezialisierungen des verhaltensorientierten Szenarios abgegrenzt werden, über die nachfolgend heterogene Entwicklungsaspekte von Data Marts sowie EU-Komponenten aufgezeigt werden. Tab. 7.6 stellt die Phasen *recognize* und *react* anhand des Instanzbezugs einander gegenüber. Mit dem Merkmal *instanzspezifisch* wird die Berücksichtigung von Ereignissen bzw. Reaktionen (z.B. direkte Steuerung) im Ablauf einer Prozessinstanz beschrieben. Mit *instanzübergreifend* wird der Fokus auf die Prozessumgebung (z.B. Ressourcenausstattung) sowie die indirekte Steuerung von Prozessinstanzen gelegt.

		recognize	
		instanzspezifisch	instanzübergreifend
react	instanzspezifisch	(I) Entscheidungsunterstützung als Prozessbaustein (direkt, <i>embedded BI</i>)	(II) Entscheidungsunterstützung zur Prozesssteuerung (direkt)
	instanzübergreifend	(III) Entscheidungsunterstützung zur Gestaltung der Prozessumgebung (indirekt)	(IV) Entscheidungsunterstützung zur Gestaltung der Prozessumgebung (indirekt)

Tab. 7.6: Abgrenzung der Data-Mart-Nutzung im verhaltensorientierten Szenario

Anhand der Quadranten in Tab. 7.6 werden drei Spezialisierungen zur Integration von DWH- und EU-Komponenten mit dem operativen AwS abgeleitet:

- **Entscheidungsunterstützung als Prozessbaustein** (Quadrant I): Die Nutzung von entscheidungsunterstützenden Funktionen wird nach dem *request-response*-Paradigma explizit im Prozessschema berücksichtigt. Das Ergebnis beeinflusst maßgeblich den Ablauf einer Prozessinstanz. In der Literatur (z.B. bei Schwingel 2010) wird hier auch der Begriff *embedded BI* verwendet.
- **Entscheidungsunterstützung zur Prozesssteuerung** (Quadrant II): Auch hier wird die direkte Steuerung einer/mehrerer Prozessinstanzen betrachtet. Der Auslöser bzw. das Ereignis, aufgrund dessen eine Reaktion notwendig wird, findet sich jedoch nicht im Prozessschema. Mögliche Auslöser liegen in der Prozessumgebung. Dies können beispielsweise instanzübergreifende (komplexe) Ereignisse oder Ergebnisse von (ggf. Data-Mart-basierten) Prozessanalysen sein.
- **Entscheidungsunterstützung zur Gestaltung der Prozessumgebung** (Quadrant III u. IV): Die indirekte Steuerung von Prozessen erfolgt durch Anpassungen der Prozessumgebung. Es sind keine Zugriffspunkte zur direkten Steuerung im Prozessschema vorgesehen. Der Auslöser kann entweder

in der Prozessumgebung liegen (Quadrant IV) oder aktiv aus einer Prozessinstanz heraus erfolgen (Quadrant III). Die Lokalisierung des Auslösers ist für die Spezifikation von Data Marts nicht ausschlaggebend.

Nachfolgend vorgestellte Lösungen für die abgegrenzten Spezialisierungen sind als Muster (*Patterns*) zu verstehen. Komplexe Szenarien, beispielsweise mit Reaktionen sowohl bzgl. der Instanzen als auch der Prozessumgebung, können anhand dieser Muster realisiert werden. Nachfolgend werden im Wesentlichen die Identifikation von Data Marts, die Abgrenzung des Informationsbedarfs sowie die Rückkopplung auf das Prozessschema für die jeweilige Spezialisierung thematisiert. Die Entwicklung der Data-Mart-Komponenten sowie der Komponenten zur Datenintegration erfolgt analog zu den in Abschn. 7.6.1 (strukturorientiertes Szenario) angegebenen Schritten. Die Erläuterungen werden durch eine weitere Detaillierung der Fallstudie *Mobility Provision* (nachfolgender Abschnitt) unterstützt.

Detaillierung der Fallstudie *Mobility Provision*: Angebotserweiterung

In Kap. 4.4 werden neben der operativen Durchführung des Vertriebsprozesses sowohl die kontinuierliche Analyse und Verbesserung des Vertriebskanals als auch die Angebotserweiterung zur Erhöhung der Kundenbindung während der Leihphase sowie zur optimalen Gestaltung der Flottenauslastung als unterstützende Szenarien genannt. In Abb. 7.16 wird eine Erweiterung des VES aus Kap. 4.3.2 zur Angebotserweiterung dargestellt. Während der Ausleihphase eines e-Cars soll, in Abhängigkeit von Flottenauslastung oder auch des Fahrverhaltens des Kunden, dieser mit individualisierten Angebotserweiterungen zu einem ressourceneffizienten Verhalten animiert werden. Das betriebliche Objekt *Auftragsverwaltung* wird hierzu weiter in die Objekte *Kommunikationsdienst (K-Dienst)* sowie *Verwaltungsdienst (V-Dienst)* zerlegt (vollständiges Zerlegungsprotokoll siehe Anhang A.3). Aus verhaltensorientierter Sicht erfolgen in regelmäßigen Abständen (Umwelt ereignis *Zeitintervall*) Meldungen über Fahrverhalten u. -daten des e-Cars seitens der *Flottenlenkung* an den *Verwaltungsdienst*. Dort erfolgt eine Entscheidung, ob der Kunde (z.B. per Kurznachrichte oder App) lediglich über seinen Leihstatus informiert oder ihm ein individuelles Angebot unterbreitet wird. Der Auslöser zum Senden eines Angebotes kann auch aus dem Prozessumfeld heraus, modelliert als Umwelt ereignis (Markierung R), erfolgen. Die Interaktion mit dem Kunden ist jeweils dem *Kommunikationsdienst* zugeordnet.

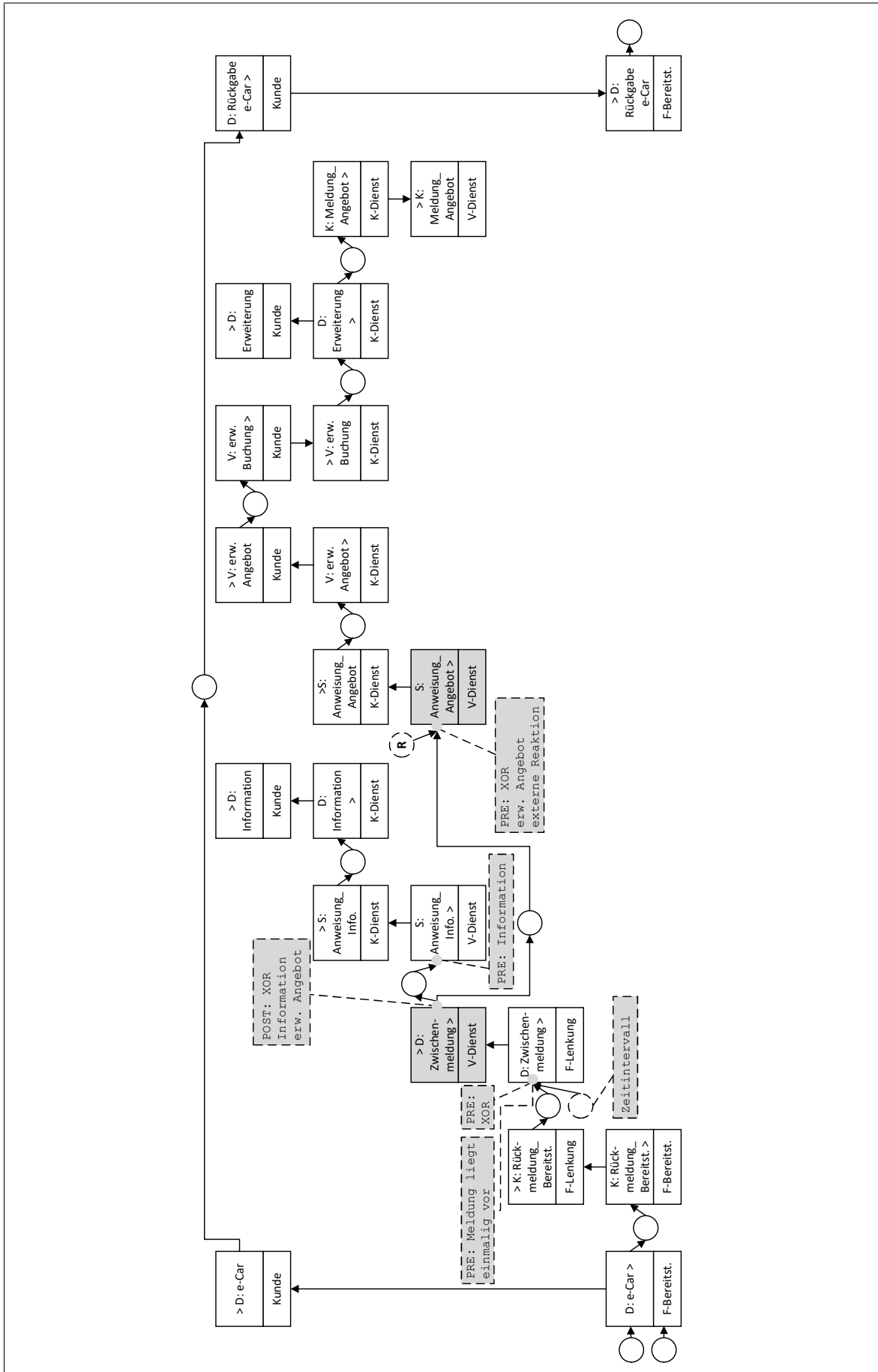


Abb. 7.16: Ausschnitt des VES zur Angebotsverweiterung

Entscheidungsunterstützung als Prozessbaustein

Entscheidungsunterstützung als Prozessbaustein bedeutet, dass der Prozess selbst Nutzer des EU-Systems und damit transitiv des pDWH-Systems wird. Die genannten Systeme stellen spezifische Aufgabenträger dar, deren Nutzung auf AT-Ebene im Rahmen der Spezifikation operativer Workflows festgelegt wird. Das Geschäftsprozessmodell der Aufgabenebene selbst fokussiert auf die Aufgabenaußensicht. Sach- und Formalziele der Aufgabe können diese lediglich als Entscheidungsaufgabe charakterisieren und Hinweise auf eine mögliche Nutzung dispositiver Komponenten im Rahmen des Lösungsverfahrens auf AT-Ebene liefern.

Zur Erläuterung wird die betriebliche Aufgabe >D: *Zwischenmeldung* aus dem VES (Abb. 7.16) aufgegriffen. Das Sachziel kann mit Verwaltung der Zwischenmeldung sowie Entscheidung zur Angebotserweiterung beschrieben werden. Als ausgewähltes Formalziel der Entscheidungsfindung wird, in Anlehnung an das Szenario *Angebotserweiterung* (vgl. Kap. 4.4), die Erhöhung der Kundenbindung übernommen. Mit Übergang auf die AT-Ebene wird festgelegt, die Entscheidung durch eine Komponente des EU-Systems auf Basis eines historisierten Datenbestandes durchzuführen. Abb. 7.17 zeigt den BPMN-Workflow (erweiterter Sub-Prozess) startend mit Empfang und Speicherung (Datenkomponente *Mietdaten*) der Zwischenmeldung. Eine Entscheidung bzgl. der Angebotserweiterung soll nur in späten Phasen des Mietprozesses stattfinden (Aktivität *Prüfung Angebotsentscheidung*). Die Entscheidung selbst wird von der Komponente *DS Angebot* getroffen und über die Aktivitäten *Anfrage* bzw. *Empfang Angebotsentscheidung* in den Workflow eingebunden. Als Parameter dienen aktuellen Kundendaten, das Fahrverhalten sowie die Position des e-Cars. Zur Beantwortung der Frage, ob es sinnvoll ist, dem Kunden ein Angebot zur Erweiterung seiner Buchung zu unterbreiten, werden diese Parameter anhand des historisierten Datenbestandes des Data Marts *DM Angebot* gespiegelt. Die Data-Mart-Nutzung selbst bleibt aus Sicht des Workflows transparent.

EU-Komponenten stellen im betrachteten Szenario den Nutzer eines Data Marts dar. Der Informationsbedarf ist auf Basis ihres Lösungsverfahrens ableitbar. Im Beispiel soll die Entscheidung auf Basis des Verhaltens von Vergleichsgruppen erfolgen. Dimensionen zur Analyse von Alterstrukturen, zum Geschlecht oder

zum Mietverhalten (Phase des Leihprozesses, Fahrverhalten) von Kunden werden genutzt. Anhand von Kennzahlen wie *Annahmequote* der Vergleichsgruppe soll eingeschätzt werden, ob ein Kunde zur Annahme eines Angebotes tendiert. Das Überfrachten von Kunden mit erweiterten Angeboten soll vermieden werden.

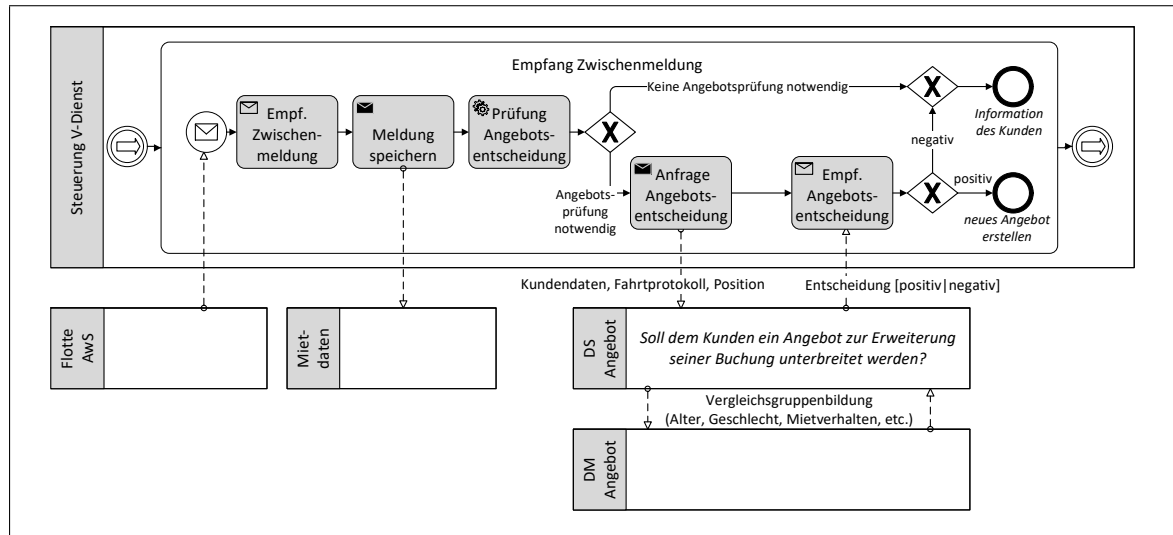


Abb. 7.17: Beispiel zur Entscheidungsunterstützung als Prozessbaustein

Die *Data-Mart*-Spezifikation erfolgt nach Abgrenzung des Informationsbedarfs analog zu Abschn. 7.6.1. Dimensionen für *Altersstruktur* und *Geschlecht* werden von der Datenkomponente *Kundendaten*, für das *Mietverhalten* von der Datenkomponente *Mietprozessdaten* bereitgestellt. Die Basiskennzahlen zur Annahmequote werden durch die Komponente *Buchungsdaten* angeboten. Die Aktualisierung des nachfrageorientierten Datenbestandes wird nicht als zeitkritisch bewertet und erfolgt mehrmals täglich anhand definierter Zeitintervalle.

Entscheidungsunterstützung zur Prozesssteuerung

Das Szenario *Entscheidungsunterstützung zur Prozesssteuerung* beschreibt das direkte, steuernde Eingreifen in eine Prozessinstanz. Der Auslöser des *OpDW-Cycles* liegt in der Prozessumgebung. Zum steuernden Eingreifen sind entsprechende Zugriffspunkte im Prozessschema zu hinterlegen. Konkret werden diese im GP-Schema (VES) deutlich. Objektexterne Ereignisse werden über das Konzept des *Umwelt ereignisses* modelliert. In Abb. 7.16 ist ein Umwelt ereignis (Markierung R für externe Reaktion) als Vorereignis der Aufgabe *S: Anweisung Angebot* angegeben. Anhand dieses Ereignisses kann die Erstellung sowie der Versand eines erweiterten Angebotes an den Kunden aus der Prozessumgebung heraus angestoßen werden.

Nutzergruppen, deren Zielsetzung und Informationsbedarf für ein Data Mart ergeben sich nicht aus dem Prozessschema, sondern aus übergeordneten GP-Zielen. In Kap. 4.3.2 wird die flexible Gestaltung der Flottenauslastung beschrieben. Bei zu hoher oder zu niedriger Auslastung sollen Kunden durch alternative Angebote zu entsprechendem Ausleih- bzw. Rückgabeverhalten animiert werden. Die Kennzahl *Anzahl erfolgreich durchgeführter Reservierungen von e-Cars* entlang der Dimensionen *Standortgeographie* und *Reservierungszeitraum* kann beispielsweise mit vergleichbaren Zeiträumen (z.B. Ferienzeit) vergangener Perioden abgeglichen werden. Anhand der Aufgabe *V: Buchungsauftrag* des *Vertriebskanals* kann eine geeignete GP-orientierte DWH-Sicht für diesen Fall abgeleitet werden. Als Reaktion auf eine mögliche Unterversorgung mit e-Cars an bestimmten Stationen (z.B. Flughafen) kann der Kunde durch Anpassung seines Angebotes zur dortigen Rückgabe animiert werden. Für die personell getroffene Entscheidung, beispielsweise durch einen Disponenten, ist ein aktueller Datenbestand notwendig. Mit Abschluss einer Buchung (Vertriebskanal *>D: Buchungsdokumentation*, vgl. Kap. 4.3) ist der Datenbestand des zugehörigen Data Marts zu aktualisieren.

Entscheidungsunterstützung zu Gestaltung der Prozessumgebung

Das Szenario zur *Gestaltung der Prozessumgebung* beschreibt Maßnahmen, die sich auf die Prozessumgebung (z.B. die Ressourcenausstattung) und damit nur indirekt auf den Ablauf von Prozessinstanzen auswirken. In den Prozessschemata sind weder auf Aufgaben- noch auf AT-Ebene Zugriffspunkte für Reaktionen zu berücksichtigen. Auslöser des *OpDW-Cycles* können im Workflowschema (Auffunktivitäten), in der Ausführungsumgebung (Ereignisüberwachung) oder in der Prozessumgebung liegen (vgl. Szenario zur Prozesssteuerung).

Zur Erläuterung kann das Beispiel des vorherigen Abschnittes (flexible Gestaltung der Flottenauslastung) sowie das dort identifizierte Data Mart (im Anwendungsbeispiel zur Referenzarchitektur in Kap. 5.3.4 als *Auslastungsanalyse* bezeichnet) aufgegriffen werden. Neben direkter Einflussnahme auf laufende Leihprozesse durch angepasste Angebote ist auch das Starten eines Serviceprozesses als Reaktion auf Anfragespitzen möglich. Ziel des Serviceprozesses ist es, die Fahrzeugausstattung der Standorte anhand von Erfahrungswerten und Prognosen anzupassen. Der Hauptprozess *Mobility Provision* wird davon nur indirekt beeinflusst. Der Auslöser kann, wie bereits im vorherigen Abschnitt vorgestellt, ein Ereignis in

der Prozessumgebung sein. Alternativ kann auch das Ereignis *Suche ausgeschöpft* der Aktivität *Empfang Verfügbarkeitsliste* (vgl. Abb. 7.9) als Auslöser dienen. Dieses signalisiert eine Negativantwort an den Kunden aufgrund nicht ausreichend verfügbarer Fahrzeuge.

7.7 Zusammenfassung und Diskussion der Ergebnisse

Abschließend werden die Ergebnisse zur Entwicklungsmethodik zusammengefasst (Abschn. 7.7.1) sowie bzgl. generischer und spezifischer Anforderungen diskutiert (Abschn. 7.7.2).

7.7.1 Zusammenfassung der Entwicklungsmethodik

Gegenstand des Kapitels stellen die Anwendung und die Erweiterung der SOM-Methodik zur Entwicklung operativ-dispositiver Systeme dar. Das Zielsystem ist dabei durch die Referenzarchitektur (Kap. 5) sowie das pDWH-Schema (Kap. 6) beschrieben. Durch die Methodik soll die Anwendbarkeit dieser Artefakte in betrieblichen Szenarien ermöglicht und gefördert werden. Zur Unterstützung der Entwickler in ihrer Tätigkeit werden für Teilsysteme der Referenzarchitektur (operatives AwS, angebots- und nachfrageorientiertes DWH-System) jeweils eigenständige Methodiken mit Modellarchitektur und Vorgehensmodell vorgestellt. Die Entwicklung der einzelnen Teilsysteme kann zeitlich entkoppelt erfolgen. Durch das Geschäftsprozessmodell als gemeinsamen Bezugspunkt sowie die Berücksichtigung der Abhängigkeiten zwischen den Teil-Methodiken wird die **Konsistenz der Teilsysteme** sichergestellt.

Zur Entwicklung des operativen AwS wird die SOM-Methodik um die Ebene des komponentenorientierten Anwendungsmodells erweitert. Die Abgrenzung von Komponenten selbst ist nicht Gegenstand der Arbeit. Zur vollständigen Beschreibung der Fallstudienanwendung sowie der Entwicklungsmethodik ist der Aspekt dennoch zu beleuchten. Für die Anwendung auf die Fallstudie *Mobility Provision* wird ein pragmatischer Ansatz gewählt. Die Gesamtmethodik ist jedoch nicht auf dieses Vorgehen festgelegt. In der Literatur werden zahlreiche Ansätze aufbereitet. Explizit sind hier die Ansätze nach Ferstl et al. (1997) und nach Friedrich et al. (2016), die beide eine Erweiterung der SOM-Methodik darstellen, zu nennen. Zentrale Anforderung sowohl an die Entwicklung des operativen Systems

und damit auch an die Methode zur Abgrenzung von Komponenten sowie an die Entwicklung der DWH-Teilsysteme ist der durchgehende, **explizite Geschäftsprozessbezug** (*Traceability*). Dieser ermöglicht neben der Analyse des Prozessablaufs auf AT-Ebene auch die Bewertung der betrieblichen Leistungserstellung auf Aufgabenebene. Durch definierte und dokumentierbare Zerlegungs- und Konsolidierungsschritte unterstützt die SOM-Methodik diesen durchgehenden GP-Bezug. In der Arbeit von Wolf (2015) wird ein möglicher Ansatz zur notwendigen Dokumentation der Entwicklungsschritte vorgestellt. Über das Konzept der betrieblichen Aufgabe sowie die Differenzierung von Außen- und Innensicht wird im Rahmen der SOM-Methodik der Wechsel zwischen Aufgaben- und AT-Ebene beschrieben. Dieses Konzept findet auch im Rahmen der gpDWH-Entwicklung zur Einordnung des AT-DWH-Schemas in den Kontext der betrieblichen Leistungserstellung Anwendung. Es ermöglicht neben der technischen (AT-spezifischen) und der fachlichen (aufgabenspezifischen) Perspektive auf die multidimensionalen Prozessdaten auch den Wechsel zwischen diesen. Anhand der Eigenschaften einer betrieblichen Aufgabe können aufgabenorientierte Sichten auf das AT-DWH-Schema gebildet und anhand ihrer Positionierung im Geschäftsprozess anhand von Leistungs-, Lenkungs- und Vorgangssicht eingeordnet werden. Dies in Summe führt zur Spezialisierung eines pDWH-Systems zu einem *geschäftsprozessorientierten DWH-System*.

Nutzer des gpDWH-Systems werden über das struktur- und das verhaltensorientierte pDWH-Szenario bestimmt. Der jeweilige Informationsbedarf wird, je nach Perspektive, über das GP-DWH- bzw. das AT-DWH-Schema gedeckt und in Data Marts bereitgestellt. Gerade im Falle des verhaltensorientierten Szenarios kann der operative Workflow selbst, wenn auch transitiv über Entscheidungskomponenten, gpDWH-Nutzer sein. In diesem Fall stellt weniger die Entwicklung als vielmehr die Integration in das ganzheitliche operativ-dispositive System ein Gestaltungskriterium dar. Anhand des *OpDW-Cycles* werden drei Spezialisierungen des verhaltensorientierten Szenarios mit Referenzcharakter vorgestellt.

7.7.2 Diskussion der Entwicklungsmethodik

Zur Diskussion der Entwicklungsmethodik werden zudem generische Anforderungen nach Sinz (1998) herangezogen.

Unterstützung bei der Aufstellung umfassender Modelle Umfassende Modelle beschreiben Struktur und Verhalten eines betrieblichen Informationssystems (Sinz 1998). Bei der Entwicklung des operativen AwS werden im Rahmen der SOM-Methodik bereits sowohl struktur- (Daten-, Funktions-, Interaktionssicht) als auch verhaltensorientierte (Ablaufsicht) Sichten erfasst. Wie in Abschn. 7.4.2 aufgezeigt, werden diese auch im komponentenorientierten Anwendungsmodell angemessen berücksichtigt. Bei der Spezifikation des dispositiven Teilsystems stehen zunächst Datenstrukturen im Fokus. Mit Spezifikation von Workflows zur Datenintegration sowie der Interaktion zwischen Komponenten im Rahmen der Datenbereitstellung werden auch die übrigen Sichten betrachtet. Die Unterstützung der Evolution betrieblicher Informationssysteme stellt nach Sinz (1998) einen weiteren Aspekt einer umfassenden Modellierungsmethodik dar. Die *flexible Anpassbarkeit* stellt eine grundsätzliche Anforderung an Referenzarchitektur und Datenmodell dar und wird durch beide Artefakte unterstützt. Die Berücksichtigung dynamischer Strukturmerkmale in den Modellen auf Aufgaben- bzw. AT-Ebene (z.B. deren Evolution oder Versionierung) ist nicht Gegenstand der vorliegenden Arbeit, liefert jedoch Ansatzpunkte für künftige Forschungsarbeiten. Für das *Semantische Objektmodell* wird beispielsweise bei Wolf (2015) die Strukturflexibilität untersucht und in einer Methodik zusammengefasst.

Unterstützung bei der Aufstellung richtiger Modelle Richtige Modelle zeichnen sich zum einen dadurch aus, dass sie konsistent und vollständig zum Begriffssystem (Metapher und Metamodell) sind. Als anderer Bezugspunkt zur Bewertung der *Richtigkeit* gilt das abzubildende reale System. Dieses ist struktur- und verhaltenstreu abzubilden (Sinz 1998). Das *Semantische Objektmodell* zur GP-orientierten Systementwicklung sowie das *Semantische Data-Warehouse-Modell* zur Spezifikation des pDWH-Schemas basieren auf Metamodellen mit semantisch klar definierten Bausteinen und Regeln zu deren Verknüpfung. Durch das explizite Metamodell wird die Spezifikation konsistenter und vollständiger Modelle auch in der vorliegenden Arbeit unterstützt. Die struktur- und verhaltenstreu Modellbildung wird u.a. durch explizit angegebene Metaphern, aber auch durch die klar definierte Semantik der Modellbausteine ermöglicht. Unterstützt durch die Metapher können relevante Konzepte der betrieblichen Realität identifiziert und auf die Modellbausteine übertragen werden. Die Entwicklungsmethodik für operativ-dispositive Systeme setzt zudem auf einer Referenzarchitektur mit abgrenzbaren

Komponententypen sowie der Markt-Metapher zur Strukturierung von Systemen anhand von Angebot und Nachfrage auf. In Kombination mit etablierten Modellierungsansätzen sind zahlreiche Konzepte zur Spezifikation *richtiger* Modelle berücksichtigt.

Unterstützung bei der Aufstellung geeigneter Modelle Die Eignung eines Modells wird anhand des Modellierungszwecks bewertet (Sinz 1998). Dieser kann im vorliegenden Fall mit der Entwicklung GP-orientierter operativ-dispositiver AwS benannt werden. In Abschn. 7.1 werden explizit die Anforderungen *Geschäftsprozessbezug* und *Konsistenz der Teilsysteme* genannt. Die Erfüllung dieser beiden Anforderungen durch die vorgestellte Entwicklungsmethodik ist bereits in Abschn. 7.7.1 zusammengefasst. Ausgehend vom Konzept des *Geschäftsprozesses* erfolgt zum einen die durchgehend dokumentierte Entwicklung des operativen System sowie davon ausgehend die Spezifikation von pDWH-System und GP-orientierten Sichten. Zum anderen stellt das Konzept des *Geschäftsprozesses* aber auch den gemeinsamen Bezugspunkt zur konsistenten Entwicklung der Teilsysteme dar. Aufgrund der Erfüllung der Anforderungen wird die vorgestellte Methodik als geeignet zur Entwicklung operativ-dispositiver AwS bewertet. Die zugrunde liegenden Modellierungsansätze (SOM, SDWM) sind, jeweils für sich genommen, geeignet zur Spezifikation von Geschäftsprozess- und Anwendungsmodellen bzw. von DWH-Schemata. Diese werden im Rahmen der vorliegenden Methodik lediglich kombiniert und in ihren ursprünglich intendierten Nutzungskontext eingesetzt.

Unterstützung bei der Bewältigung der Komplexität von Modellen Als Abbild der betrieblichen Realität besitzen Modellsysteme i.d.R. eine hohe Komplexität, die es im Rahmen der Modellbildung zu beherrschen gilt (Sinz 1998). Als Mechanismen zur Komplexitätsbeherrschung finden die Bildung von *Teilsystemen*, *Modellebenen* sowie *Modellsichten* Anwendung. Als Teilsysteme werden das *operative AwS* sowie das *angebots-* und das *nachfrageorientierte pDWH-System* abgegrenzt. Die Entwicklung erfolgt jeweils über eine eigenständige Methodik mit Modellarchitektur und Vorgehensmodell. Übergreifend werden die Modellarchitekturen einheitlich nach Aufgaben- und AT-Ebene strukturiert. So wird ein einheitliches Begriffssystem in den Modellarchitekturen bzw. deren Modellebenen zugrunde gelegt. Auch die Beziehungen zwischen Modellebenen sind spezifiziert. In der Modellarchitektur des angebotsorientierten pDWH-Systems wird anhand des Konzeptes der betrieb-

lichen Aufgabe beispielsweise die Einordnung der AT-Informationen in den Kontext eines Geschäftsprozesses beschrieben. Zur Beherrschung der Komplexität der Entwicklung des operativen AwS tragen zudem Sichten (struktur- und verhaltensorientiert) bei. Zusätzliche Komplexität bei der Entwicklung operativ-dispositiver Systeme entsteht durch die Abhängigkeiten bzw. Konsistenzanforderungen zwischen den Teilsystemen. Diese sind als Beziehung zwischen Modellarchitekturen spezifiziert. Anhand von Regeln wird der Entwickler bei der Ableitung von Modellen auf Basis dieser Beziehungen unterstützt. Als Beispiel kann die Erstellung des pDWH-Schemas oder die Definition aufgabenorientierter Sichten auf Basis der Modelle des operativen Teilsystems genannt werden.

8 Extension der Entwicklungsmethodik

Kapitelzusammenfassung: Mit der Entwicklungsmethodik aus Kap. 7 wird die Anwendung der Referenzarchitektur (Kap. 5) einschl. des pDWH-Schemas (Kap. 6) in betrieblichen Szenarien ermöglicht. Nachfolgend wird das Anwendungsgebiet für diese Artefakte über die Extension der Entwicklungsmethodik abgesteckt. Das Kapitel wird als *Proof-of-Concept* verstanden. Die Extension der Entwicklungsmethodik wird anhand von drei Aspekten aufgezeigt: In Abschn. 8.1 erfolgt zunächst die Positionierung der Ergebnisse anhand verwandter Themen wie *Business Intelligence* oder *Geschäftsprozess-Management*. Die Gültigkeit der Methodik auf Basis abweichender Modellierungssprachen wird in Abschn. 8.2 bewertet. In Abschn. 8.3 werden schließlich Merkmale von Geschäftsprozessen aufgezeigt, über die sich die Anwendbarkeit operativ-dispositiven AwS unabhängig von der betrieblichen Domäne bestimmen lässt.

8.1 Themenorientierte Positionierung der Ergebnisse

Zunächst wird eine Einordnung der Methodik sowie der damit entwickelten Artefakte anhand verwandter, wenn auch oft unterschiedlich abgegrenzter Konzepte vorgenommen. Hierzu werden die Themen *Business Intelligence* (BI) sowie *Geschäftsprozess-Management* (GPM) aufgegriffen. Nachfolgend werden diese Begriffe bestimmt und Ergebnissen der Arbeit gegenübergestellt. Damit sollen Potenziale der Arbeit nochmals explizit aufgezeigt werden. Zunächst erfolgt die Positionierung des gpDWH-Konzeptes (Architektur und Entwicklung) im Kontext der Business Intelligence (Abschn. 8.1.1). In Abschn. 8.1.2 wird zudem das ganzheitliche Konzept GP-orientierter operativ-dispositiver AwS im Rahmen des GPM bewertet.

8.1.1 Positionierung im Kontext der Business Intelligence

Die Positionierung des gpDWH-Konzeptes im Kontext der BI erfolgt sowohl allgemein als auch spezifisch für den Teilbereich der *operational Business Intelligence*.

Generelle Einordnung des gpDWH-Konzeptes als BI-Konzept

In der verwandten Literatur zu gpDWH-Systemen besteht eine Vielzahl z.T. heterogen abgegrenzter Themen. Einige davon werden dem gpDWH-Konzept nachfolgend gegenübergestellt. Die Begriffe *Closed-loop*, *Active* sowie *Real-time Data-Warehousing* beschreiben Ansätze, die auf Basis des vorgestellten gpDWH-Konzeptes realisierbar sind. *Closed-loop Data-Warehousing* bezeichnet das Grundkonzept, aus DWH-Systemen gewonnene Informationen rückgekoppelt in operativen oder anderen analytischen Systemen zu verwenden. Die Rückkopplung kann automatisiert erfolgen (Kemper et al. 2010, S. 93). Ein *aktives DWH-System* erweitert dieses Konzept um die automatisierte Entscheidungsunterstützung mit positiven Auswirkungen auf die Latenz von der Erfassung eines Ereignisses bis hin zur Reaktion (Kemper et al. 2010, S. 95). Sowohl die Rückkopplung von analytischen Informationen als auch die automatisierte Entscheidungsfindung sind in der Referenzarchitektur sowie in der Entwicklungsmethodik berücksichtigt. Beide Ansätze sind auf Basis des gpDWH-Konzeptes realisierbar. Unter dem Begriff *Real-time Data-Warehousing* werden Ansätze zusammengefasst, die auf die Minimierung der Latenz zur Datenbereitstellung im Sekundenbereich abzielen (Kemper et al. 2010, S. 93). Die Minimierung der Latenz ist auch eine Anforderung an die Referenzarchitektur. Grundsätzlich ist die Konzeption auf die Bereitstellung von Daten und die Reaktion mit angemessener Latenz (*right-time*) ausgerichtet. Für die Realisierung nahe Echtzeit kann die Architektur als fachlicher Rahmen dienen, der auf technologischer Ebene entsprechend auszugestaltet ist. Entsprechende Entscheidungen sind im Entwicklungsprozess zu berücksichtigen.

Der Begriff *Business Process Intelligence* (BPI) stellt den übergeordneten Kontext für das gpDWH-Konzept dar. Zur Einordnung wird zunächst allgemein der Begriff des *DWH-Systems* im Kontext der BI eingeordnet. Als Konsolidierung verschiedener Interpretationen wird die Definition nach Kemper et al. (2010, S. 9) verwendet: „*Business Intelligence (BI) bezeichnet einen integrierten, unternehmensspezifischen, IT-basierten Gesamtansatz zur betrieblichen Entscheidungsunterstützung. (...)*“ Basierend auf dieser Definition beschreiben Bauer & Günzel (2013, S. 13f.) die Erweiterung vom DWH- hin zum BI-Kontext v.a. anhand der Aspekte Integration und Analyse. Integration bezieht sich neben Daten auch auf Strategien, Prozesse und Technologien. Der Analyse-Aspekt schließt auch die Erzeugung von Wissen

im betrieblichen Kontext mit ein (Bauer & Günzel 2013, S. 13f.). Anhand des Ordnungsrahmens nach Gluchowski et al. (2008, S. 92) wird ein DWH-System als ein technologisches Konzept klassifiziert, das im Rahmen der BI die Aufgaben der Speicherung und Bereitstellung analyserelevanter Daten übernimmt. Diese Abgrenzung wird auf die spezifischen Formen *gpDWH-System* und BPI übertragen: *gpDWH-Systeme* übernehmen im BPI-Kontext die Bereitstellung und die Verwaltung von Daten für das struktur- und das verhaltensorientierte Szenario. Ein Anwendungsbereich des stukturorientierten DWH-Szenarios wird unter dem Begriff *Process Performance Managements* beschrieben. Ziel ist es, analytischen Zugang zu Geschäftsprozessen zu schaffen, um diese kontinuierlich anzupassen und zu verbessern (Gluchowski et al. 2008, S. 360). Für das verhaltensorientierte Szenario findet sich der spezifischere Anwendungsbereich des *operational BI*, der nachfolgend betrachtet wird.

Einordnung des pDWH-Konzeptes als operational BI-Konzept

Bei Eckerson (2007) wird ein Framework vorgeschlagen, um Technologien, Ansätze und Architekturen des *operational BI* zu bewerten. Die verschiedenen Ausbaustufen lassen sich dabei anhand des Wertbeitrages sowie der Datenlatenz beschreiben. Anhand dieses Frameworks, das in Abb. 8.1 abgebildet ist, soll auch die GP-orientierte Entwicklung operativ-dispositiver AwS basierend auf der Referenzarchitektur aus Kap. 5 eingeordnet werden. Der Begriff *operational BI* wird bei Gluchowski et al. (2009) wie folgt definiert: „*Operational BI bezeichnet integrierte geschäftsprozessorientierte Systeme, die mithilfe klassischer Business-Intelligence-Methoden auf Basis zeitnaher, prozessualer Ablaufdaten und in aller Regel auch historischer Daten eine Realtime- (Near-Realtime-) Unterstützung für zeitkritische Entscheidungen während des Prozessablaufes bieten.*“ Beschrieben werden Merkmale, die auch auf das verhaltensorientierte pDWH-Szenario zutreffen. Somit erscheint das Framework nach Eckerson (2007) geeignet, um dessen Realisierung auf Basis der Referenzarchitektur GP-orientierter operativ-dispositiver AwS zu bewerten. Nach Eckerson (2007) sind hierzu je Ausbaustufe folgende Fragen zu beantworten: *Erfolgt die Datenbereitstellung durch das DWH-System für die jeweilige Ausbaustufe in angemessener Art und Weise? Wie können operative Teilsysteme, EU-Systeme sowie deren Integration zur jeweiligen Ausbaustufe beitragen?*

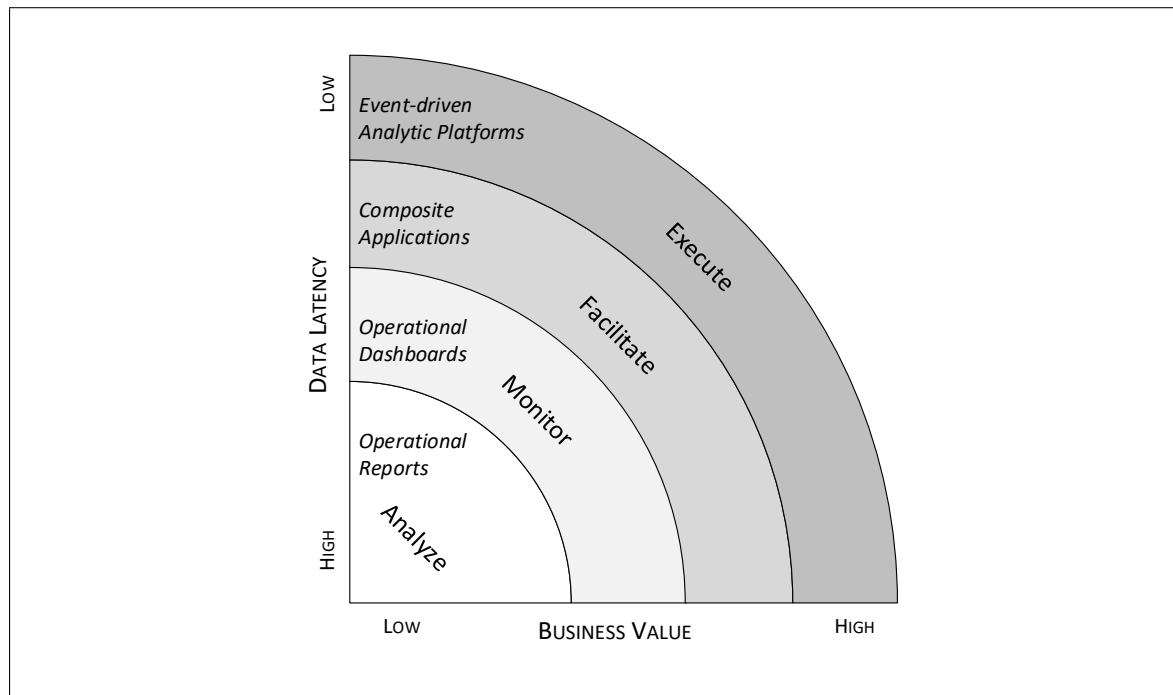


Abb. 8.1: Ausbaustufen operativer Business-Intelligence-Lösungen (Eckerson 2007) (eigene Darstellung)

Wie in Abb. 8.1 gezeigt, geht mit steigender Ausbaustufe des *operational BI* auch ein steigender Wertbeitrag für das Unternehmen einher. Zur Realisierung ist jedoch auch die Latenz der Datenbereitstellung mit zunehmender Ausbaustufe zu reduzieren (Eckerson 2007). Nachfolgend werden die Ausbaustufen vorgestellt und die Unterstützung durch das vorgestellte *gpDWH*-Konzept thematisiert.

- **Analyze Processes:** Die Prozessanalyse erfolgt in Form operativer Berichte. Operative Mitarbeiter können auf Basis aktueller Daten bei kurzfristigen Entscheidungen unterstützt werden, um Prozessabläufe effizienter zu gestalten. Die Realisierung auf Basis eines DWH-Systems setzt die Datenbereitstellung mit geringer Latenz voraus (Eckerson 2007). Diese Ausbaustufe wird durch das *gpDWH-Konzept* unterstützt. Zur kurzfristigen Datenbereitstellung können Ereignisse spezifiziert werden. Je nach akzeptierter Latenz eines Anwendungsfalles sind ggf. weitere Anpassungen bzw. Entscheidungen auf technologischer Ebene notwendig. Als Vorteil gegenüber der Realisierung auf Basis eines ODS bzw. direkt auf Basis des operativen Systems sind Analysen entlang der Zeitachse sowie multidimensionale Operationen zu nennen.
- **Monitor Processes:** Zielsetzung der zweiten Ausbaustufe ist die Prozessüberwachung mit Erkennung, Meldung und Analyse von Ausnahmeverhal-

ten. Als Grundlagen dienen auch hier sofort verfügbare Daten, die als Key Performance Indikatoren (KPI) in *Dashboards* grafisch aufbereitet vorliegen. Wesentlicher Unterschied zu Berichten der Stufe *Analyze Processes* ist die Aufbereitung und Darstellung der Daten (Eckerson 2007). Dashboards sind Teil der Präsentationsschicht und werden entsprechend der Anforderungen an ein Data Mart der *flexiblen Data-Mart-Architektur* realisiert. Die Berechnung von KPIs kann auf Basis des pDWH-Schemas erfolgen. Die Bereitstellung der Daten erfolgt analog zur Stufe *Analyze Prozesse* auf Basis fachlicher Ereignisse sowie technologischer Überlegungen. Die Überwachung der KPIs selbst sowie die Meldung von Ausnahmeverhalten erfolgt über die *Dashboards*. Ein Beispiel zur Analyse von Ausnahmeverhalten ist im Rahmen GP-orientierter Sichten in Kap 7.5.2 vorgestellt.

- **Facilitate Processes:** Die Zusammenführung von operativen Prozessen mit deren Analyse in einem Anwendungssystem oder Portal stellt die dritte Ausbaustufe von *operational BI* dar. Die Serviceorientierung bildet ein mögliches Paradigma zur Realisierung als sog. *Composite Application*. Operative Mitarbeiter können ohne Wechsel des Systems sowohl operative als auch zugehörige analytische Aufgaben übernehmen (Eckerson 2007). Das Konzept der *operativ-dispositiven Referenzarchitektur* identifiziert Komponenten anhand ihrer Funktionen. Die Nutzung operativ und dispositiv abgegrenzter Komponenten über ein gemeinsames Portal ist dabei ebenso möglich wie die verteilte Nutzung. Durch das komponentenorientierte Paradigma (Verwandtschaft zu SOA) ist die Bildung angesprochener *Composite Applications* möglich.
- **Execute Processes:** Zielsetzung der vierten Ausbaustufe ist die Erfassung und (teil-) automatisierte, regel-basierte Reaktion auf Ereignisse in Geschäftsprozessen. Die Automatisierung basiert auf Plattformen zur Ereignisverarbeitung und Analyse sowie Vorhersagemodellen. Operational BI wird als Teil des Prozesses angesehen. In diese Ausbaustufe fallen auch Konzepte wie *Business Activity Monitoring* oder *Business Process Management* (Eckerson 2007). Zur Einordnung in diese Stufe wird das Konzept der *operativ-dispositiven Referenzarchitektur* ganzheitlich betrachtet. Zur (teil-) automatisierten Entscheidungsfindung dient das EU-System. Vorhersage- und Entscheidungsmodelle sowie Plattformen zur Ereignisverarbeitung sind hier

verortet. Die Betrachtung des Teilsystems aus Innensicht ist nicht Gegenstand der Arbeit. Durch die Integration des EU-Systems mit operativen Vorgangskomponenten sowie den Data Marts des gpDWH-Systems kann die BI-Funktionalität in den operativen Prozess integriert werden. In Kap. 7.6.2 werden verschiedene Formen zur Integration automatisierter Entscheidungen in operative Abläufe vorgestellt.

Es ist festzuhalten, dass das gpDWH-Konzept mit *flexibler Data-Mart-Architektur* sowie dessen Einbettung in eine Referenzarchitektur operativ-dispositiver AWS eine solide und erweiterbare Basis für *operational BI* der vier Ausbaustufen darstellt. Bzgl. der kurzfristigen Bereitstellung von Daten sind auf den Einzelfall ausgerichtete technologische Entscheidungen zu treffen.

8.1.2 Positionierung im Geschäftsprozess-Management

Geschäftsprozess-Management (GPM) beschreibt Konzepte, Methoden und Technologien zur Unterstützung von Administration, Konfiguration, Einsatz, Betrieb und Analyse von Geschäftsprozessen (Weske 2012, S. 5). Die einzelnen Phasen des GPM sind in Abb. 8.2 als Prozesslebenszyklus angegeben und können wie folgt charakterisiert werden: Die Phase *Gestaltung und Analyse* hat sowohl die Identifikation und Modellierung von Geschäftsprozessen als auch deren Validierung (z.B. im Rahmen von Workshops), Simulation und Verifikation zum Gegenstand (Weske 2012, S. 11ff.). Die Phase *Konfiguration* umfasst die Auswahl des Laufzeitsystems sowie die Implementierung, das Testen und das Bereitstellen von Softwarebausteinen zur Prozessausführung (Weske 2012, S. 13f.). Die Laufzeit von Prozessinstanzen selbst wird in der Phase *Einsatz/Betrieb* mit Ausführung, Überwachung und Wartung beschrieben. Unter Überwachung wird die visuelle Aufbereitung von Ausführungszuständen der Prozessinstanzen verstanden (Weske 2012, S. 14f.). Die zur Laufzeit erhobenen Ablaufdaten stellen den Input für die nachfolgende *Evaluationsphase* dar. *Process Mining* oder *Business Activity Monitoring* stellen dabei Konzepte zur Bewertung von Prozessausführung und Laufzeitumgebung dar (Weske 2012, S. 15). Die phasenübergreifende Verwaltung von Prozessartefakten (z.B. Modelle, Dokumentationen) sowie von Verantwortlichkeiten wird durch die zentrale Funktion *Administration & Stakeholder* beschrieben (Weske 2012, S. 15).

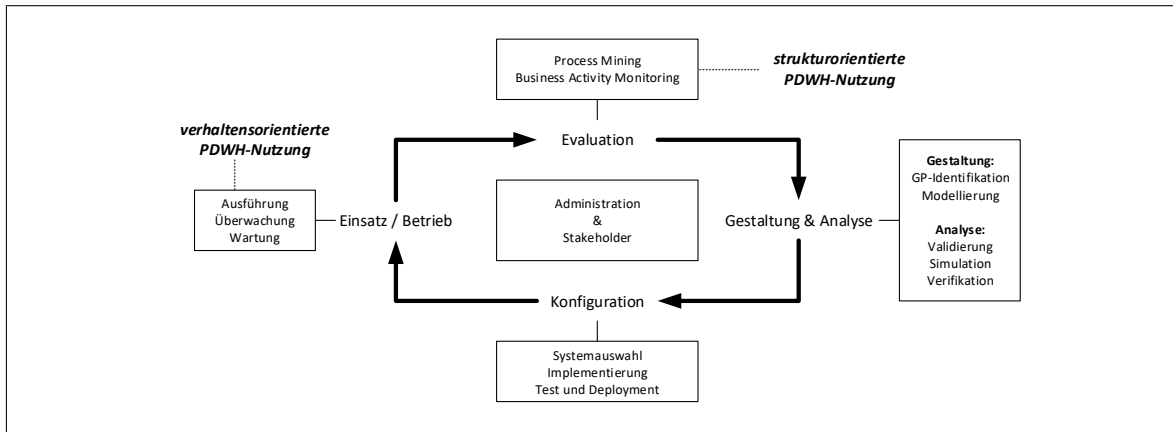


Abb. 8.2: Einordnung der pDWH-Szenarien im Prozesslebenszyklus nach Weske (2012, S. 12)

Die Zielsetzung des gpDWH-Konzeptes ist die Unterstützung des struktur- sowie des verhaltensorientierten pDWH-Szenarios. Beide Szenarien werden, wie ebenfalls in Abb. 8.2 dargestellt, nachfolgend in den GPM-Zyklus eingeordnet. Die Zielsetzung der verhaltensorientierten Nutzung eines gpDWH-Systems ist in Kap. 3.5.2 mit der aktiven Steuerung von Prozessen zur Laufzeit beschrieben und wird der Phase *Einsatz/Betrieb* zugeordnet. Das gpDWH-System stellt für diese Phase eine aufbereitete und historisierte Datengrundlage bereit, die nicht nur die reine Überwachung, sondern auf Basis der Referenzarchitektur operativ-dispositiver AwS (Kap. 5) auch steuernden Maßnahmen zur Laufzeit ermöglicht. Die zur Laufzeit aufgezeichneten sowie im gpDWH-System historisierten und multidimensional aufbereiteten Ablaufdaten können im Weiteren als Grundlage der Phase *Evaluation* genutzt werden. Dem strukturorientierten Szenario entsprechend wird die Identifikation von Schwachstellen im Prozess sowie die Ableitung gestalterischer Maßnahmen unterstützt. Das gpDWH-System kann als geeignete Grundlage zur Realisierung von *Business Activity Monitoring* oder *Process Mining* angesehen werden. Gerade die Möglichkeiten der hierarchischen Analyse vom Geschäftsprozessmodell über einzelne Aufgaben hin zu Bestandteilen des Lösungsverfahrens mit konsequentem Bezug zu übergeordneten GP-Zielen stellt einen Mehrwert in dieser Phase dar.

8.2 Abstraktion verwendeter Modellierungsansätze

Die Entwicklung GP-orientierter operativ-dispositiver AwS ist auf Basis der Workflow-Sprache BPMN sowie der SOM-Methodik konzipiert. Nachfolgend wird ge-

zeigt, dass die Entwicklung von Teilsystemen unabhängig von diesen Ansätzen erfolgen kann. In Abschn. 8.2.1 wird die Abhängigkeit des pDWH-Schemas von der BPMN zur Spezifikation von Workflows diskutiert. Abschn. 8.2.2 leitet anhand des vorgestellten SOM-basierten Vorgehens generelle Anforderungen an eine Entwicklungsmethodik für GP-orientierte operativ-dispositive AwS ab. In Abschn. 8.2.3 wird schließlich die Modellierung eines pDWH-Systems nach dem *Data-Vault*-Ansatz als Alternative zum multidimensionalen Datenmodell diskutiert.

8.2.1 Verallgemeinerung des prozessorientierten Data-Warehouse-Schemas

Das pDWH-Schema (vgl. Kap. 6) ist auf Basis des Metamodells der BPMN abgeleitet und somit zunächst auf die Analyse von BPMN-Workflows ausgerichtet. Nachfolgend werden die identifizierten Dimensionen hinsichtlich ihrer Allgemeingültigkeit näher betrachtet. Abstrahiert von den spezifischen Begriffen *Workflow* und *Geschäftsprozess* kann unter einem Prozess eine zeitlich-sachlogische Folge von Aktivitäten bzw. Zuständen verstanden werden, die inhaltlich abgeschlossen und auf die Bearbeitung eines betriebswirtschaftlich relevanten Objektes ausgerichtet sind (Becker & Schütte 2004, S. 107). Ein Prozess kann abstrakt als Abfolge von Knoten und Kanten beschrieben werden. Knoten können nicht nur Aktivitäten oder Zustände, sondern u.a. auch Ereignisse oder Entscheidungsknoten sein. Ein konkreter Ansatz zur Prozessmodellierung, wie beispielsweise die BPMN, legt über sein Metamodell die zur Verfügung stehenden Knotentypen fest.

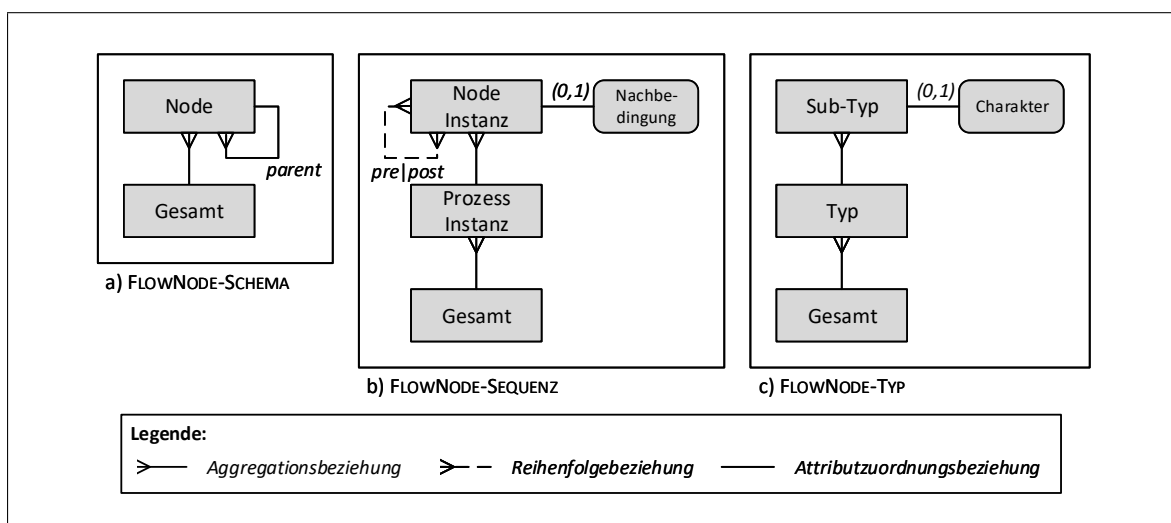


Abb. 8.3: Sprachunabhängige Dimensionen des pDWH-Schemas

Das vorgestellte pDWH-Schema abstrahiert von spezifischen Knotentypen. In den Dimensionen *FlowNode-Schema*, *FlowNode-Sequenz* und *FlowNode-Typ* wird der generische Begriff des Knotens (*Node*) verwendet. Abb. 8.3 zeigt diese drei Dimensionen. In der Dimension *FlowNode-Typ* werden Knotentypen eines konkreten Modellierungsansatzes als Instanzen verwaltet. Die Dimensionshierarchie ist ggf. an das Typ-System des jeweiligen Ansatzes anzupassen. Das zugehörige Metamodell bzw. darin spezifizierte Generalisierungsbeziehungen können hierzu Hinweise geben. Die zeitlich-sachlogische Reihenfolge von Knoten wird über die Dimension *FlowNode-Sequenz* für Prozessinstanzen aufgezeichnet. Die Reihenfolge der Knotenausführungen ist dabei unabhängig von deren Typ. Die Zuordnung eines Knoten zu einem Prozessschema erfolgt ebenfalls unabhängig von der Modellierungssprache in der Dimension *FlowNode-Schema*. Die hierarchische Strukturierung von Prozessen ist hingegen spezifisch für die BPMN und bei Anwendung des pDWH-Schemas im Zusammenhang mit einem anderen Modellierungsansatz auf Anwendbarkeit zu überprüfen. Die Dimensionen *Participant* sowie *Swimlane* des pDWH-Schemas aus Kap. 6 stellen BPMN-spezifische Dimensionen dar. Die Dimensionen aus Abb. 8.3 sind hingegen sprachunabhängig und bilden die Grundlage zur Konzeption von pDWH-Schemata für heterogene Prozesssprachen durch Erweiterung. Die Erweiterung kann durch Anpassung sprachunabhängiger oder durch Hinzufügen sprachspezifischer Dimensionen (z.B. *Participant* am Beispiel der BPMN) erfolgen.

8.2.2 Verallgemeinerung der Entwicklungsmethodik

Die Entwicklung des GP-orientierten operativ-dispositiven AwS stellt eine Erweiterung der Entwicklung operativer AwS dar. Die vorgestellte Methodik profitiert von Artefakten und Eigenschaften der hierzu erweiterten SOM-Methodik. Im Folgenden liegt der Fokus auf diesen Eigenschaften, die exemplarisch auch anhand der *Architektur integrierter Informationssysteme* (ARIS) diskutiert werden. Die Abgrenzung von operativen Komponenten ist nicht Gegenstand der Arbeit und wird im Weiteren nicht näher betrachtet.

Eigenschaften der Entwicklungsmethodik

Zunächst werden drei zentrale Eigenschaften der SOM-Methodik bei der Entwicklung des operativen Systems aufgezeigt, die (aufeinander aufbauend) auch die

Entwicklung und Nutzung des gpDWH-Systems positiv beeinflussen bzw. ermöglichen.

- **Beschreibung auf mehreren Modellebenen:** Die SOM-Methodik beschreibt ein betriebliches System auf mehreren Modellebenen, im Speziellen auf *Aufgaben-* und *Aufgabenträgerebene* (vgl. Kap. 2.2). Während die Aufgabenebene einen Geschäftsprozess fachlich unter Berücksichtigung von Struktur und Verhalten beschreibt, hat die AT-Ebene im Wesentlichen die Spezifikation des operativen AwS zum Gegenstand. Diese Trennung ermöglicht nicht nur während der Entwicklung unterschiedliche Perspektiven auf das betriebliche System. Auch zur Analyse operativer Abläufe können unterschiedliche Perspektiven eingenommen werden. Das gpDWH-System berücksichtigt neben einer technisch geprägten Berichtsebene (*AT-Ebene*) auch die Einordnung der Daten in den fachlichen Kontext der übergeordneten betrieblichen Aufgabe (*Aufgabenebene*).
- **Durchgehender Bezug zum Geschäftsprozess:** Alle Artefakte der Spezifikation des operativen AwS leiten sich transitiv aus der GP-Spezifikation auf Aufgabenebene ab und sind auf diese zurückzuführen. Dies stellt die Voraussetzung zur Realisierung unterschiedlicher Perspektiven (Aufgaben- und AT-Perspektive) auf den prozessorientierten Datenbestand dar. Das zentrale Konzept zur GP-Einordnung bildet die *betriebliche Aufgabe*. Mit dem Wechsel zwischen Außen- und Innensicht auf eine Aufgabe erfolgt der Wechsel der Berichtsperspektive (Aufgaben- vs. AT-orientiert) sowie der Modellebene.
- **Hierarchisches Vorgehen zur Spezifikation von Struktur und Verhalten:** Die SOM-Methodik gibt ein Regelwerk zur hierarchischen Zerlegung betrieblicher Systeme vor. Diesem folgend werden sukzessive ausgetauschte Leistungen (Leistungssicht), dazugehörige Lenkungsnachrichten (Lenkungssicht) sowie betriebliche Objekte aufgedeckt und aus Struktur- und Verhaltenssicht beschrieben (vgl. Kap. 4.1). Die hierarchische Zerlegung im Rahmen der Strukturmodellierung wird in Abschn. 7.5.2 genutzt, um Ablaufinformationen eines gpDWH-Systems phasenorientiert einzuordnen (z.B. differenziert nach Anbahnungs-, Vereinbarungs- u. Durchführungsphase) und im Kontext der betrieblichen Leistungserstellung zu bewerten. Die strukturorientierte Sicht der SOM-Methodik bietet mit dem KOS zudem die

Grundlage zur Ergänzung des pDWH-Schemas um betriebswirtschaftliche Kennzahlen und Dimensionen.

Diese drei Eigenschaften der Entwicklung des operativen Teilsystems sind ausschlaggebend für die voll umfängliche Nutzung und Entwicklung eines gpDWH-Systems. Die SOM-Methodik stellt aber keine zwingende Voraussetzung dar. Nachfolgend wird ARIS exemplarisch als Alternative zur Entwicklung GP-orientierter operativ-dispositiver AwS diskutiert.

Exemplarische Diskussion anhand von ARIS

Zunächst wird ARIS kurz eingeführt, bevor anhand der drei Eigenschaften die Eignung zur Spezifikation eines gpDWH-Systems bewertet wird. Für vertiefende Informationen wird auf Scheer (2001, 2002) verwiesen. ARIS ist grundsätzlich als Rahmenkonzept zur ganzheitlichen Spezifikation von Informationssystemen vom fachlichen Geschäftsprozessmodell hin zur Implementierung von AwS konzipiert, kann aber auch für stärker betriebswirtschaftliche Themen, wie Prozesskostenrechnung oder Qualitätsmanagement, eingesetzt werden (Scheer 2002, S. 1). Nachfolgend ist v.a. das Entwicklungskonzept von Bedeutung. Beeinflusst von Informations- und Kommunikationstechnik erfolgt zunächst die globale strategische Geschäftsprozessanalyse und Sollkonzeption (Scheer 2002, S. 39f.). Auf dieser Grundlage erfolgt die Spezifikation des betrieblichen Informationssystems entlang des *ARIS-Hauses* (Abb. 8.4). Die Sichtenbildung stellt dabei ein zentrales Hilfsmittel zur Strukturierung und Komplexitätsbeherrschung von Geschäftsprozessmodellen dar (Scheer 2002, S. 33). Die hierzu unterstützten Sichten werden, basierend auf Scheer (2002, S. 36f.), nachfolgend vorgestellt:

- **Funktionssicht:** Funktionen transformieren Input- zu Outputleistungen und folgen dabei bestimmten Zielsetzungen. Zusammen mit automatisierten Lösungsverfahren der Funktionen erfolgt die Beschreibung in der Funktionssicht.
- **Datensicht:** Die Datensicht beschreibt Umfelddaten, Nachrichten und Ereignisse, die für die Funktionsdurchführung assoziiert werden können.
- **Organisationssicht:** Über die Organisationssicht wird die Aufbauorganisation beschrieben. Aufgabenträger (z.B. Personen, Betriebsmittel, Computer)

mit vergleichbarer Funktion oder gleichem Aufgabenobjekt werden zu Organisationseinheiten zusammengefasst.

- **Leistungssicht:** In der Leistungssicht werden materielle und immaterielle Leistungen, unterschieden nach Input- und Outputleistungen, von Funktionen beschrieben.
- **Steuerungssicht:** Die Steuerungssicht führt die zuvor eingeführten, isolierten (strukturorientierten) Sichten zum gesamten Geschäftsprozess zusammen. Die Steuerungssicht stellt den Rahmen zur vollständigen Prozessbeschreibung aus Verhaltenssicht dar.

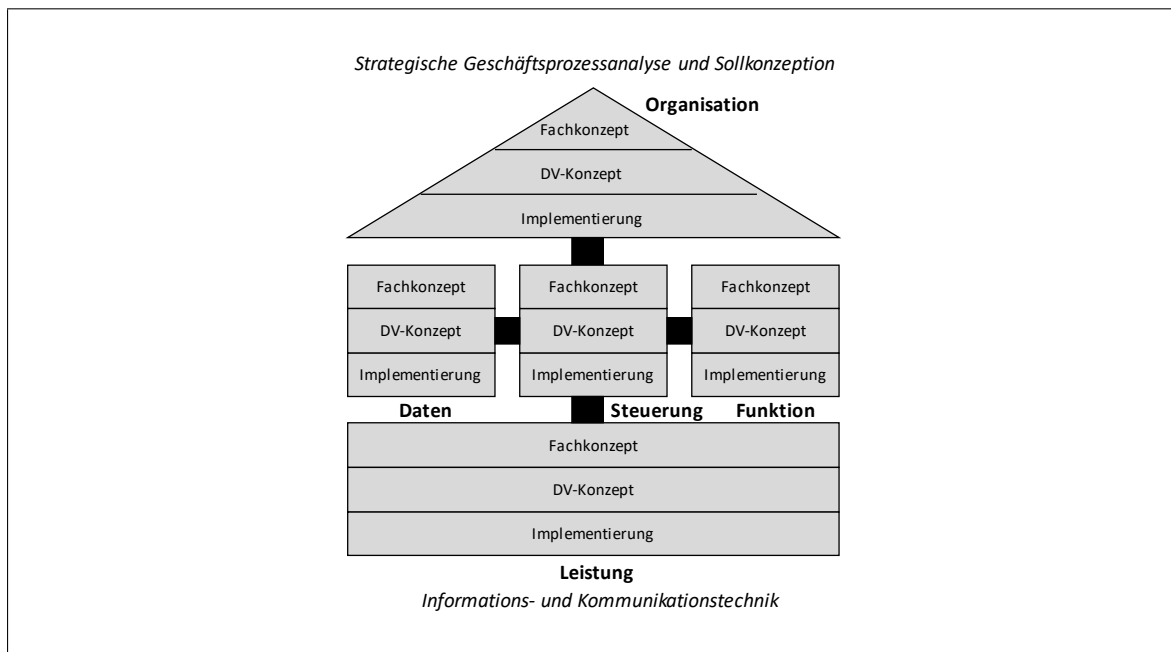


Abb. 8.4: Das ARIS-Haus (Scheer 2002, S. 41)

Die Beschreibung der einzelnen Sichten erfolgt nach Scheer (2002, S. 40f.) jeweils vom *Fachkonzept* über das *DV-Konzept* hin zur technischen *Implementierung*. Dies gilt auch für die Steuerungssicht, so dass auf jeder Ebene der Gesamtzusammenhang der Sichten beschrieben werden kann. Das *Fachkonzept* beschreibt Geschäftsprozesse aus fachlicher, betriebswirtschaftlich-organisatorischer Perspektive und bildet den Ausgangspunkt für das *DV-Konzept*. Dort wird die fachliche Spezifikation an die Anforderungen potenzieller Anwendungssysteme angepasst, in die sie in der Phase der *Implementierung* überführt werden. Das generelle Phasenkonzept ist dabei nicht auf ein spezifisches Vorgehensmodell festgelegt (Scheer 2002, S. 40f.). Die *Ereignisgesteuerte Prozesskette* (EPK) stellt den zentralen Modellierungsansatz für das Fachkonzept der Steuerungssicht dar und führt Daten-,

Funktions-, Leistungs- und Organisationssicht zusammen. Innerhalb der Sichten können spezifische, auf die Sicht ausgerichtete Modellierungsansätze Verwendung finden (Scheer 2002, S. 33).

Nachfolgend werden die generellen Eigenschaften einer Entwicklungsmethodik operativer Systeme aus dem vorherigen Abschnitt aufgegriffen und mit Bezug auf ARIS diskutiert:

- **Beschreibung auf mehreren Modellebenen:** ARIS betrachtet die Modellebenen Fach-, DV-Konzept und Implementierung jeweils getrennt nach Sichten. Gerade mit Fach- und DV-Konzept werden Perspektiven auf einen Geschäftsprozess eingenommen, die mit der Aufgaben- und AT-Ebene der SOM-Methodik vergleichbar sind. Die verhaltensorientierte Steuerungssicht als zentrale Sicht zur Integration der strukturorientierten ARIS-Sichten bietet eine geeignete Beschreibung auf den verschiedenen Modellebenen, die zum einen zur Ableitung des prozessorientierten DWH-Schemas und zum anderen zur GP-orientierten Interpretation genutzt werden kann.
- **Durchgehender Bezug zum Geschäftsprozess:** Die EPK stellt den zentralen Modellierungsansatz für die Steuerungssicht des Fachkonzeptes dar und manifestiert das ARIS-GP-Verständnis. Bei der Spezifikation des DV-Konzeptes sind grundlegende Eigenschaften bzw. Restriktionen des AwS zu berücksichtigen (Scheer 2001). Als solche Eigenschaften können Merkmale des komponentenorientierten Paradigmas (vgl. Kap. 2.3) oder der Workflow-Spezifikation (vgl. Kap. 2.2.1) verstanden werden. Möglich ist die Einordnung der BPMN zur Spezifikation der Steuerungssicht des *DV-Konzeptes*. Sie bildet damit auch die Grundlage zur Ableitung des pDWH-Schemas (vgl. Kap. 6). Wie in Abschn. 8.2.1 gezeigt, ist das Datenschema aber auch auf alternative Workflow-Sprachen anpassbar. Fach- und DV-Konzept bauen aufeinander auf, wodurch der durchgehende GP-Bezug bei Dokumentation aller Entwicklungsschritte als erfüllt betrachtet werden kann. Zentrales Konzept zum Wechsel der Berichtsperspektive in Kap. 7.5.2 bildet das Konzept der *betrieblichen Aufgabe*. Die EPK beschreibt einen Geschäftsprozess in Form von Funktionen, die über Ereignisse gekoppelt sind. Das in Kap. 2.1.2 kommunizierte Aufgabenverständnis kann grundsätzlich auf das Konzept der Funktion übertragen werden. Bei der Einordnung AT-spezifischer Er-

gebnisse in den Kontext des Geschäftsprozesses ist die jeweilige Abbildung von Funktionen im DV-Konzept zu berücksichtigen.

- **Hierarchisches Vorgehen zur Spezifikation von Struktur und Verhalten:** Anhand der verhaltensorientierten ARIS-Steuerungssicht kann ein Geschäftsprozess (pfad- oder abschnittsbezogen) auf Basis eines gpDWH-Systems analysiert werden. Weitere Analysemöglichkeiten hängen von der Realisierung der strukturorientierten ARIS-Sichten ab. Dies betrifft v.a. die Bewertung von Geschäftsprozessen anhand seiner durch hierarchische Zerlegung aufgedeckten Phasen. Bei Seidlmeier (2015, S. 21ff.) wird beispielsweise der Funktionsbaum zur Spezifikation der Funktionssicht vorgeschlagen. Dieser ermöglicht die schrittweise hierarchische Verfeinerung von Funktionen, wenn gleich die phasenorientierte Zerlegung der SOM-Methodik nicht in vergleichbarer Form berücksichtigt ist. Die ARIS-Datensicht, z.B. als *Entity-Relationship-Modell* modelliert, kann zur Identifikation fachlicher Kennzahlen und Dimensionen herangezogen werden.

Anhand grundlegender Eigenschaften der Entwicklung operativer AwS kann die Entwicklung von gpDWH-Systemen losgelöst von der SOM-Methodik auch auf alternative Methodiken übertragen werden. Im vorliegenden Abschnitt ist dies am Beispiel ARIS gezeigt. Die Trennung von Beschreibungsebenen, Sichtenbildung oder ein hierarchisches Entwicklungsvorgehen als Maßnahmen zur Komplexitätsbeherrschung finden sich in zahlreichen Entwicklungsprojekten. Unter Berücksichtigung des GP-Bezuges ist, in Abhängigkeit der Ausprägungen dieser Eigenschaften in den Projekten, auch die Erweiterung bestehender operativer AwS um gpDWH-Konzepte möglich.

8.2.3 Data-Vault als Alternative zur Modellierung des prozessorientierten Data-Warehouse-Schemas

DWH-Systeme sind durch die Bereitstellung eines historisierten Datenbestandes sowie die Möglichkeit der (ad-hoc) Aggregation quantitativer Kennzahlen entlang qualitativer Dimensionen charakterisiert (vgl. Kap. 3.1). Referenzarchitektur und Entwicklungsmethodik setzen aus diesem Grund durchgehend auf das multidimensionale Datenmodell zur Strukturierung des pDWH-Schemas (vgl. Kap. 3.3). Zur Modellierung von DWH-Schemata existieren jedoch auch weitere Ansätze.

Als relevant wird der *Data-Vault*-Ansatz nach Linstedt & Graziano (2010) angesehen, der in Version 2.0 neben dem eigentlichen Modellierungsansatz zudem eine agile Entwicklungsmethode, ein Architekturkonzept (einschl. NoSQL-Systeme) sowie einen Implementierungsansatz umfasst (Linstedt & Olschimke 2016, S. 12). Nachfolgend wird dieser Ansatz kurz vorgestellt und anschließend in die *flexible Data-Mart-Architektur* aus Kap. 5.3.2 eingeordnet. *Data-Vault* basiert auf relationalen Datenbankstrukturen, die normalisiert spezifiziert werden. Hierzu sind die nachfolgenden Tabellenkonzepte vorgesehen (z.B. Hahne 2014, S. 227ff., Inmon & Linstedt 2015, S. 139ff., Linstedt & Olschimke 2016, S. 90ff.):

- **Hub:** Hubs dienen der Verwaltung geschäftsbezogener Schlüssel (sog. *Business Keys*) mit Datenquelle und Ladezeitpunkt. Als Beispiele können die Kundennummer oder die Auftragsnummer genannt werden.
- **Link:** Link-Tabellen dienen der Verknüpfung der *Business Keys* bzw. der Verknüpfung von Link-Tabellen. Über Links werden somit beispielsweise Konzepte wie Kunde und Auftrag miteinander in Beziehung gesetzt.
- **Satellite:** Satelliten-Tabellen verwalten beschreibende Attribute zu Hubs oder Links. Kundenspezifische Attribute wie Name, Anschrift oder Kontaktdaten werden so den entsprechenden *Business Keys* des Kunden zugeordnet.

Die Modellierung mit *Data-Vault* erfolgt, basierend auf dem Konzept der *Business Keys*, nahe am operativen Geschäft bzw. dessen Systemen. Als Vorteile durch Nutzung des *Data-Vault*-Ansatzes werden Agilität (zeitnahe Reaktionen auf Marktveränderungen), Flexibilität (Ausschöpfung neuer Marktpotenziale), Skalierbarkeit, Integration heterogener Datenquellen, Transparenz und Nachvollziehbarkeit (Auditierbarkeit), die Realisierung spezialisierter Informationsanforderungen sowie die Beherrschung der Komplexität des Geschäftskontextes durch das DWH-System genannt (Linstedt & Olschimke 2016, S. 90). *Data-Vault* stellt einen Ansatz zu Modellierung des *Core Data-Warehouses* (Hahne 2014, S. 227ff.) bzw. des *Data Warehouse Layers* (Linstedt & Olschimke 2016, S. 26f.) dar. Als wichtigste Aufgaben des *Core Data-Warehouses* nennt Hahne (2014, S.218) die Datenbereitstellung in granularer und vollständig integrierter Form, die Qualitätssicherungsfunktion sowie die Distributionsfunktion für nachgelagerte Schichten (z.B. Data Marts). Während das multidimensionale Datenmodell an Entscheidungsfeldern orientiert ist (*subject-oriented*), erfolgt die Spezifikation des *Data-Vault* ausgerichtet auf die genannten Funktionen der Integration aus heterogenen Quellsystemen

und Bereitstellung der Daten für heterogene Zielsysteme (auch NoSQL-Systeme) mit Aufzeichnung der Datenhistorie und niedriger Latenz. Die Realisierung nachgelagerter Systeme erfolgt weiterhin multidimensional, ausgerichtet anhand der Entscheidungsfelder der Nutzer (Linstedt & Olschimke 2016, S. 171).

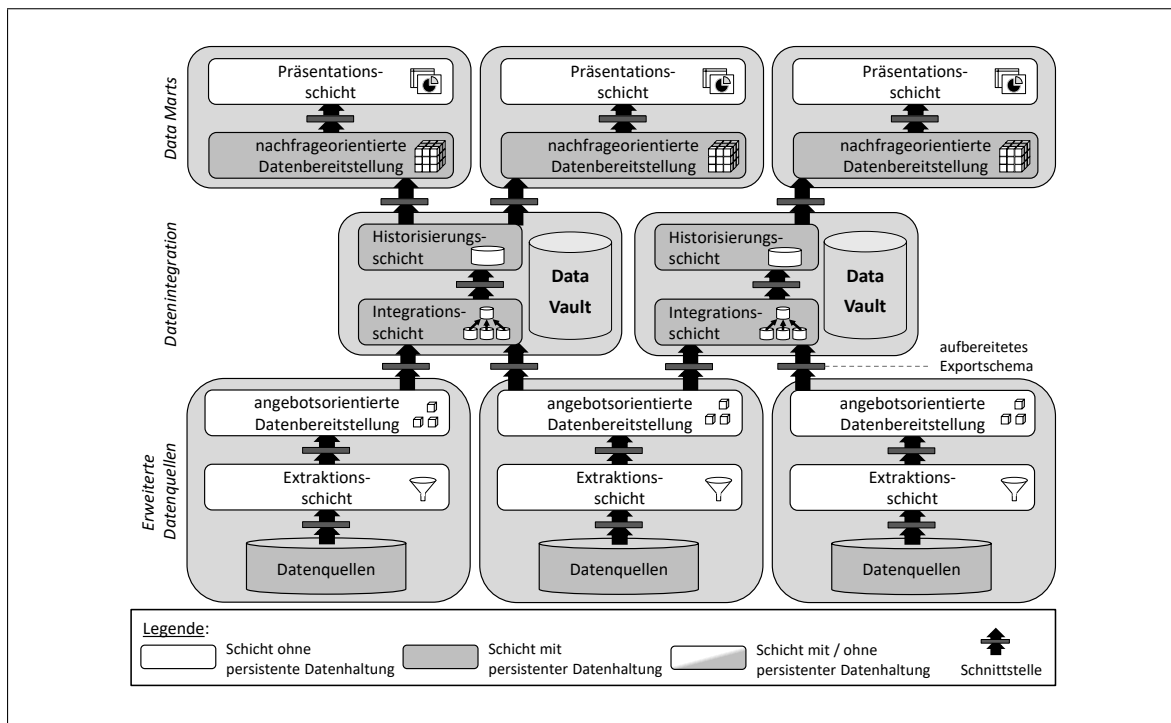


Abb. 8.5: Einordnung von Data-Vault in die flexible Data-Mart-Architektur

Dieser Argumentation folgend ist der *Data-Vault*-Ansatz eine Alternative zur Strukturierung der Datenbestände der *angebotsorientierten Datenbereitstellung*. Abb. 8.5 zeigt die Einordnung in die *flexible Data-Mart-Architektur*. *Data-Vault* beschreibt die Strukturierung relationaler Datenstrukturen und ist als solches nur bei persistenter Datenhaltung der Datenintegrationsschicht sinnvoll anwendbar. Die *Data-Vault*-Strukturen stellen hier einen integrierten und historisierten Datenbestand dar. Die *technische Historisierung* ist mit dem *Data-Vault*-Ansatz bereits möglich, zur *fachlichen Historisierung* sind jedoch noch einige geringe Anpassungen der Datenstrukturen notwendig (Hahne 2014, S. 243). Mit der persistenten Datenhaltung der *Datenintegration* auf Basis des *Data-Vault*-Ansatzes werden auch Anpassungen benachbarter Schichten notwendig: *Extraktion* und *Datenbereitstellung* der *erweiterten Datenquellen* sind auf die *Data-Vault*-Konzepte hin anzupassen. Es ist ein Wechsel von der Bildung von Dimensionsstrukturen hin zur Bildung von Satelliten-, Hub- und Linktabellen zu vollziehen. Durch die persistente Speicherung innerhalb der *Datenintegration* wird die Bereitstellung der Daten als aufberei-

tetes Exportschema überflüssig. Vielmehr werden Daten aus den erweiterten Datenquellen extrahiert und im *Data-Vault* integriert persistent gespeichert. Durch die Historisierungsmöglichkeiten des *Data-Vault*-Ansatzes kann auch die *Historisierungsschicht* aus den *Data Marts* in die *Datenintegration* vorgezogen werden. Die Bereitstellung der Daten für *Data Marts* erfolgt damit bereits historisiert und integriert. Aus Sicht der *Data Marts* liegt eine flexible Datengrundlage vor, die die Datenbereitstellung mit angemessener Latenz und Aktualität ermöglicht. Zudem wird die Nachvollziehbarkeit (Auditierbarkeit) durch die persistente Datenerhaltung auf Basis des *Data-Vault*-Ansatzes innerhalb der Datenintegration ermöglicht. Die *nachfrageorientierte Datenbereitstellung* erfolgt weiterhin multidimensional oder auch in alternativen Strukturen basierend auf NoSQL-Systemen (Linstedt & Olschmke 2016, S. 171). Die multidimensionale Aufbereitung der Daten hat sich, sofern notwendig, von der angebotsorientierten hin zur nachfrageorientierten Datenbereitstellung verschoben.

Die *flexible Data-Mart-Architektur* (vgl. Kap. 5) ist nicht an die durchgängige Anwendung des multidimensionalen Datenmodells gebunden. Sie ist, wie gezeigt, auch auf alternative DWH-Strukturen anpassbar. Dies gilt im Wesentlichen für die Schicht der angebotsorientierten Datenbereitstellung. Die eingangs erwähnten zentralen Merkmale eines DWH-Systems, wie (ad-hoc) Aggregation oder Analysen im Zeitverlauf, erfordern i.d.R. multidimensionale Strukturen und Ausrichtung auf Entscheidungsfelder der Nutzer. Aus Sicht der Entwicklungsmethodik (Kap. 7) ist eine zusätzliche Modellebene für das angebotsorientierte DWH-System zur Spezifikation des *Data-Vault-Schemas* aufzunehmen und in Beziehung mit dem multidimensionalen AT-DWH-Schema sowie den operativen Strukturen zu setzen.

8.3 Abstraktion von der Anwendungsdomäne

Referenzarchitektur und Entwicklungsmethodik werden im Rahmen der Arbeit anhand der durchgehenden Fallstudie *Mobility Provision* demonstriert. Zielsetzung des nachfolgenden Abschnittes ist die Ableitung allgemeiner Merkmale von Geschäftsprozessen, anhand derer sich die Anwendbarkeit von Methodik und Architektur bestimmen lässt. Hierzu werden zunächst wesentliche Eigenschaften des struktur- (Abschn. 8.3.1) sowie des verhaltensorientierten Szenarios (Abschn.

8.3.2) aufgezeigt. Im Weiteren (Abschn. 8.3.3) werden Anwendungsfälle aus der Literatur anhand dieser bewertet und damit Potenziale zur Nutzung von Architektur und Entwicklungsmethodik gegeben. Die Merkmale beschreiben Geschäftsprozesse (oder Ausschnitte dieser) als Gegenstand eines gpDWH-Systems. Im Rahmen eines GP-orientierten operativ-dispositiven AwS können diese sowohl das Quellsystem, den Gegenstand von Analysen als auch das Ziel von Reaktionen darstellen.

8.3.1 Allgemeine Charakterisierung des strukturorientierten Szenarios

In Abb. 8.6 wird der morphologische Kasten zur allgemeinen Charakterisierung von Anwendungsfällen des gpDWH-Systems im strukturorientierten Szenario dargestellt. Zum einen sind Merkmale des Prozesses als Gegenstand des DWH-Systems sowie als Ziel daraus abgeleiteter Reaktionen aufgeführt. Zum anderen wird das Szenario selbst anhand seiner Merkmale charakterisiert. Der in Abb. 8.6 genannte Aspekt der *Entscheidungsintensität* ist für das strukturorientierte Szenario ohne Relevanz.

Prozessmerkmale			
Automatisierungsgrad	nicht automatisiert	teilautomatisiert	vollautomatisiert
Prozessstruktur	unstrukturiert	semi-strukturiert	strukturiert
Prozessart	Hauptprozess	Serviceprozess	Managementprozess
Entscheidungsintensität	niedrig	mittel	hoch
Prozessverantwortung	intra-organisational		inter-organisational
Flexibilitätsbedarf (v.a. Strukturflexibilität)	niedrig	mittel	hoch
Merkmale des strukturorientierten pDWH-Szenarios			
Zielsetzung	Prozessgestaltung		Prozesssteuerung
GPM-Phase	Gestaltung	Konfiguration	Betrieb Evaluation
Ausrichtung	Aufgabenebene		Aufgabenträgerebene
Managementebene	operativ	taktisch	strategisch
Aktualität	kurzfristig (innerhalb eines Tages)	mittelfristig (Vortag)	langfristig (> ein Tag)
Nutzer	wenig	mittel	viel
Informationsbedarf	Ereignisse	Prozessablauf	Prozessergebnis
Datenquelle	Prozess-intern	Prozess-extern	Unternehmens-extern

Abb. 8.6: Allgemeine Charakterisierung des strukturorientierten pDWH-Szenarios

Prozessmerkmale für das strukturierte Szenario

Das strukturierte Szenario ist auf die Analyse des Prozessverhaltens und sowie dessen innovativere und effizientere Gestaltung ausgerichtet. Hierzu ist es notwendig, dass Informationen zum Ablauf eines Prozesses vorliegen und diese anhand seiner Spezifikation analysiert werden können. Anhand nachfolgender Merkmale und Merkmalsausprägungen (Abb. 8.6) kann die Eignung eines Prozesses als Gegenstand operativ-dispositiver AwS bewertet werden.

Automatisierungsgrad Der *Automatisierungsgrad* als Merkmal eines Geschäftsprozesses kann anhand der Automatisierung seiner Aufgaben bestimmt werden. Bei Weske (2012, S. 19) werden v.a. voll- und teilautomatisierte Prozesse unterschieden, die auch für die Eignung eines Prozesses als Gegenstand eines gpDWH-Systems relevant sind. Der *Automatisierungsgrad* eines Prozesses hat entscheidenden Einfluss auf die Datengrundlage. Dies kann anhand des Vorgehens zur Spezifikation des operativen Workflows aus Kap. 7.4.2 verdeutlicht werden. Automatisierbare Prozessbestandteile werden (hierarchisch) weiter verfeinert, während manuelle Aktivitäten lediglich grob-granular und aus Außensicht betrachtet werden. Im Rahmen des gpDWH-Systems können nur Daten zur Analyse angeboten werden, die während der automatisierten Durchführung auch aufgezeichnet werden. Vollautomatisierte Prozesse stellen den Idealzustand zur Prozessanalyse dar. Für teilautomatisierte Prozesse sollte die Aufgabensteuerung beim Workflow liegen, dann können für manuell durchgeführte Aktivitäten Daten zu deren Start- und Endzeitpunkt erfasst werden.

Prozessstruktur Als weiteres Merkmal ist die *Prozessstruktur* zu betrachten. Bei Huth et al. (2001) werden *ad-hoc*, *semi-strukturierte* und *strukturierte* Prozesse unterschieden. Ad-hoc-Prozesse (*unstrukturiert*) sind kurzfristig und kurzlebig und damit nur eingeschränkt plan- und spezifizierbar. Aufgrund des fehlenden Schemas sowie eingeschränkter Automatisierbarkeit sind diese nicht als Gegenstand für das gpDWH-System geeignet. *Semi-strukturierte* Prozesse sind wiederkehrend, vorab planbar und sehen in der Spezifikation explizite Ansatzpunkte für ad-hoc-Modifikationen vor. Beispiele können zur Laufzeit verfeinerbare Aktivitäten oder nicht planbares Ausnahmeverhalten sein (Huth et al. 2001). Die strukturierten Bestandteile können Gegenstand prozessorientierter Analysen sein. Für ad-hoc-Modifikationen kann die Argumentation manueller Aktivitäten aufgegriffen wer-

den. In Abhängigkeit von der Berücksichtigung von Ansatzpunkten zur ad-hoc-Modifikation im Prozessschema ist eine Berücksichtigung aus Außensicht im DWH-System möglich. *Strukturierte* (Routine-)Prozesse sind vollständig spezifiziert (Huth et al. 2001) und stellen damit die ideale Grundlage zur Ableitung der Datenbasis des gpDWH-Systems dar.

Prozessart Die *Prozessart* ist verglichen mit *Prozessstruktur* und *Automatisierungsgrad* nur von sekundärer Bedeutung. Nach Bucher & Dinter (2008a) weisen jedoch operative Prozesse tendenziell einen höheren Grad der Strukturierung und damit auch der Automatisierbarkeit auf als Managementprozesse. Als operative Prozesse mit direktem oder indirektem Beitrag zur Sachzielerfüllung werden, beispielsweise bei Ferstl & Sinz (2013, S.196), Haupt- und Serviceprozesse unterschieden. Managementprozesse hingegen übernehmen die Aufgaben Planung, Steuerung und Kontrolle (Earl 1994) und sind vorab nur eingeschränkt planbar.

Prozessverantwortung Wie in Kap. 5.3.2 gezeigt, ist die Referenzarchitektur mit angebots- und nachfrageorientierter Datenbereitstellung gerade für Anwendungsdomänen mit homogenen Datenstrukturen geeignet. Zielsetzung des strukturorientierten Szenarios ist zudem die Prozessgestaltung. Als vorteilhaft wird ein *intra-organisationaler* Prozess(-ausschnitt) angesehen, bei dem Homogenität der Datenstrukturen aber v.a. die Realisierbarkeit von gestalterischen Maßnahmen sichergestellt ist. Unter *intra-organisational* wird ein Prozess(-ausschnitt) verstanden, der zwar mit der betrieblichen Umwelt interagieren kann, jedoch vollständig unter Kontrolle der Organisation steht. Prozess sowie Interaktion mit der Umwelt bilden die Grundlage des gpDWH-Systems. Bei *inter-organisationalen* Prozessen liegen Teile des zu überwachenden Prozesses im Verantwortungsbereich von Umweltobjekten (z.B. Lieferanten), was sich negativ auf die Realisierbarkeit gestalterischer Maßnahmen auswirken kann.

Flexibilitätsbedarf Wagner et al. (2011a) differenzieren Geschäftsprozesse anhand ihres *Flexibilitätsbedarfs*, also anhand des Ausmaßes der Fähigkeiten eines Prozesses, „(...) um erfolgreich auf System- und Umweltveränderungen zu reagieren oder sie zu antizipieren, so dass das Prozessziel trotz dieser Veränderungen erfüllt wird“ (Wagner et al. 2011a, S. 56). Flexibilität kann sich dabei sowohl auf die Struktur als auch auf das Verhalten eines Prozesses beziehen (Wagner et al. 2011b). Im vorliegen-

den Fall ist der Prozess überwiegend anhand seines Bedarfs an *Strukturflexibilität* zu bewerten. Es ist die Frage zu beantworten, ob ein Prozess durch Anpassung seiner Strukturen (z.B. Modifikation von Komponenten) auf Veränderungen aus seiner Umwelt heraus reagieren können muss (Bartmann et al. 2011, Wagner et al. 2011b). Für die Klassifikation des Flexibilitätsbedarfs eines Geschäftsprozesses (niedrig, mittel, hoch) wird auf die Methoden nach Wagner et al. (2011a,b) verwiesen. Sinnvoll erscheint die Unterstützung gestalterischer Maßnahmen durch ein gpDWH-System dann, wenn der Bedarf an Strukturflexibilität als *mittel* oder *hoch* klassifiziert wird. Die Flexibilität von Geschäftsprozessen findet sich auch in der Anforderung der *flexiblen Anpassbarkeit* an die Referenzarchitektur (Kap. 5.1) wieder.

Merkmale des strukturorientierten pDWH-Szenarios

Prozesse mit den vorgestellten Merkmalsausprägungen sind potenzielle Anwendungsfälle für das strukturorientierte pDWH-Szenario. Auch dieses Szenario wird in Abb. 8.6 zusammenfassend charakterisiert. Als Zielsetzung ist in Kap. 3.5.2 die innovativere und effizientere *Prozessgestaltung* spezifiziert. Im Rahmen des GPM ist es der Phase *Evaluation* zugeordnet (vgl. Abschn. 8.1.2). Wird das gpDWH-System zur Ableitung gestalterischer Maßnahmen genutzt, ist auch die Phase *Gestaltung* betroffen. Sowohl *Aufgaben-* als auch *Aufgabenträgerbene* können Gegenstand von Analysen und von betriebswirtschaftlich geprägten Entscheidungen betroffen sein. Durch das strukturorientierte Szenario werden tendenziell taktische Entscheidungen getroffen, mit möglicherweise langfristigen Auswirkungen auf die Strategie (*Managementebene*). Insofern ist die *Aktualität* der Daten nicht als kritisch zu bewerten. Die *Nutzergruppe* umfasst wenige Nutzer. In Kap. 7.6.1 sind die Nutzerrollen *Chief Process Officer*, *Process Owner* und *Process Designer* beschrieben. Zur Deckung des *Informationsbedarfs* werden Ereignisse, der Prozessablauf aber auch das Prozessergebnis benötigt. Als *Datenquellen* werden bei Bucher et al. (2009) *Prozess-interne* und *Prozess-externe* Quellen, bei Linden et al. (2011) allgemeiner interne und externe Datenquellen unterschieden. Zur Prozessanalyse sind zwingend Prozess-interne Daten notwendig, können aber mit Prozess-externen, jedoch Unternehmens-internen Informationen angereichert werden. Bei der Nutzung Unternehmens-externer Daten führt die Homogenisierung der Daten bei der Integration zu erhöhtem Aufwand. Im Integrationskonzept aus Kap. 5.4.2 sind keine Homogenisierungsschritte berücksichtigt und wäre um diese zu erweitern.

8.3.2 Allgemeine Charakterisierung des verhaltensorientierten Szenarios

In Abb. 8.7 wird das verhaltensorientierte Szenario charakterisiert. Nachfolgend wird v.a. auf im Vergleich zum strukturorientierten Szenario (vgl. Abschn. 8.3.1) abweichende Merkmale eingegangen.

Prozessmerkmale für das verhaltensorientierte Szenario

Bei Bucher & Dinter (2008a) wird durch Experteninterviews aufgedeckt, dass die Komplexität zur Realisierung des verhaltensorientierten Szenarios mit dem Grad der *Strukturierung* der Prozesse zusammenhängt. Analog zu Abschn. 8.3.1 werden (teil- bzw. vollautomatisierte) *Haupt-* und *Serviceprozesse* als geeignet angesehen. *Semi-strukturierte* Prozesse können, wie im vorherigen Abschnitt erläutert, nicht vollständig über ihr Ablaufschema analysiert werden. Auf ein vollständig spezifiziertes Ablaufschema kann in der Datenbasis ggf. verzichtet werden, reduziert aber den zur Verfügung stehenden Entscheidungskontext bis hin zur reinen Einbettung fachlicher Kennzahlen in betriebswirtschaftliche Dimensionen. Der größte Nutzen des verhaltensorientierten Szenarios wird bei Bucher & Dinter (2008a) bei der Unterstützung prozessbezogener, von Personen getroffenen Entscheidungen zur Laufzeit gesehen. Die Gestaltung von *ad-hoc*-Modifikationen semi-strukturierter Prozesse stellt einen möglichen Anwendungsfall dar. Der Aspekt der *Entscheidungsintensität* wird bei Linden et al. (2011) als Merkmal für das Konzept *Business Process Intelligence* verwendet. In der vorliegende Arbeit wird dieses Merkmal auf einen Geschäftsprozess übertragen. Unter Intensität wird primär die Komplexität von Entscheidungen im Prozessablauf verstanden. Komplexe Entscheidungen sind durch Bereitstellung entsprechender Datenbestände und Entscheidungsmodelle zu unterstützen. Zudem wird die Entscheidungsintensität durch die Häufigkeit derartiger Entscheidungen beeinflusst. Anhand der Häufigkeit kann die Zuordnung zu *mittel* bzw. *hoch* vorgenommen werden. Die Häufigkeit von Entscheidungen stellt einen Einflussfaktor auf den Entschluss für die Entwicklung und Einführung entsprechender Systeme dar. Die Zielsetzung des verhaltensorientierten Szenarios ist mit aktiver *Prozesssteuerung* spezifiziert. Zu analysierende und v.a. steuernde Prozessausschnitte sollten im Verantwortungsbereich des Unternehmens (*intra-organisational*) liegen. Für das verhaltensorientierte Szenario sind Geschäftsprozesse anhand ihrer Verhaltensflexibilität zu be-

werten. Die grundsätzliche Bewertung (mittel und hoch) des *Flexibilitätsbedarfs* kann aus dem strukturorientierten Szenario übertragen werden. Verhaltensflexibilität beschreibt die Auswahl bestimmter Prozessvarianten zur Lauf- oder Gestaltungszeit als Reaktion auf Veränderungen in der Prozessumwelt (Bartmann et al. 2011, Wagner et al. 2011b). In Kap. 7.6.2 werden mit *Entscheidungsunterstützung* (i) als Prozessbaustein, (ii) zur Prozesssteuerung sowie (iii) zur Gestaltung der Prozessumgebung drei Szenarien zur Unterstützung der Verhaltensflexibilität aufgezeigt.

Prozessmerkmale				
Automatisierungsgrad	nicht automatisiert	teilautomatisiert	vollautomatisiert	
Prozessstruktur	unstrukturiert	semi-strukturiert	strukturiert	
Prozessart	Hauptprozess	Serviceprozess	Managementprozess	
Entscheidungsintensität	niedrig	mittel	hoch	
Prozessverantwortung	intra-organisational		inter-organisational	
Flexibilitätsbedarf (v.a. Verhaltensflexibilität)	niedrig	mittel	hoch	
Merkmale des verhaltensorientierten pDWH-Szenarios				
Zielsetzung	Prozessgestaltung		Prozesssteuerung	
GPM-Phase	Gestaltung	Konfiguration	Betrieb	Evaluation
Ausrichtung	Aufgabenebene		Aufgabenträgerebene	
Managementebene	operativ	taktisch	strategisch	
Aktualität	kurzfristig (innerhalb eines Tages)	mittelfristig (Vortag)	langfristig (> ein Tag)	
Nutzer	wenig	mittel	viel	
Informationsbedarf	Ereignisse	Prozessablauf	Prozessergebnis	
Datenquelle	Prozess-intern	Prozess-extern	Unternehmens-extern	

Abb. 8.7: Allgem. Charakterisierung des verhaltensorientierten pDWH-Szenarios

Merkmale des verhaltensorientierten pDWH-Szenarios

Neben den Prozessmerkmalen ist in Abb. 8.7 auch das verhaltensorientierte Szenario zusammenfassend dargestellt. In Abschn. 8.1.2 ist dieses der Phase *Betrieb* des GPM zugeordnet und steht auf operativer oder taktischer Ebene zahlreichen Nutzern (vgl. Anforderung zur *Informationsversorgung heterogener Zielgruppen*, Kap. 5.1) zur Verfügung. Gerade hinsichtlich operativer, steuernder Reaktionen ist die kurzfristige Bereitstellung von Informationen erforderlich (vgl. Anforderung *Datenbereitstellung mit angemessener Latenz*). Operative Entscheidungen können sowohl auf Daten der *Aufgaben-* als auch der *AT-Ebene* basieren. Ereignisse und Ab-

laufinformationen einer Prozessinstanz werden in einen historisierten Entscheidungskontext eingebettet. Die Homogenität der Daten, wie in Abschn. 8.3.1 angeführt, ist auch hier von Relevanz. Zusätzlicher Aufwand bei der Homogenisierung der Daten (explizit bei externen Datenquellen) führt zu Verzögerungen bei der Datenbereitstellung.

8.3.3 Bewertung beispielhafter Geschäftsprozesse

Im Weiteren werden die eingeführten Kriterien zur Bewertung von Geschäftsprozessen anhand konkreter Beispiele angewendet und damit verschiedene Anwendungsgebiete für gpDWH-Systeme aufgezeigt. Zunächst wird der Vertriebsprozess der durchgehenden Fallstudie *Mobility Provision* charakterisiert, bevor Geschäftsprozesse verwandter Arbeiten aufgegriffen werden.

Vertriebsprozess der Fallstudie *Mobility Provision* Die Charakterisierung ist in Abb. 8.8 dargestellt. Der Vertriebsprozess ist direkt an der Leistungserstellung bzw. deren Lenkung beteiligt und stellt als solches einen Hauptprozess dar. Die Prozessverantwortlichkeit liegt im Unternehmen (*intra-organisational*). Es handelt sich um einen strukturierten und vollautomatisierten Geschäftsprozess mit hohen Anforderungen an Struktur- als auch Verhaltensflexibilität (vgl. Fallstudie Kap. 4). Die hohe Entscheidungsintensität, die durch ein gpDWH-System zu unterstützen ist wird in Kap. 7.6.2 gezeigt.

<i>Mobility Provision - Vertrieb</i>			
Prozessart	Hauptprozess	Serviceprozess	Managementprozess
Prozessstruktur	unstrukturiert	semi-strukturiert	strukturiert
Automatisierungsgrad	nicht automatisiert	teilautomatisiert	vollautomatisiert
Entscheidungsintensität	niedrig	mittel	hoch
Prozessverantwortung	intra-organisational		inter-organisational
Verhaltensflexibilität	niedrig	mittel	hoch
Strukturflexibilität	niedrig	mittel	hoch

Abb. 8.8: Charakterisierung des Vertriebsprozesses der Fallstudie *Mobility Provision*

Immatrikulationsprozess an Hochschulen Der *Immatrikulationsprozess an Hochschulen* wird bei Baars & Sun (2013) als Anwendungsfall von Business Process Intelligence (BPI) motiviert. Getrieben durch den intensiven Wettbewerb zwischen

Hochschulen um Studierende ist der Prozess schnell und fehlerfrei durchzuführen. Der motivierte Prozessabschnitt fokussiert auf die ggf. nebenläufige Prüfung von Zulassungsanträgen bei unterschiedlichen Stellen sowie die Ermittlung und Kommunikation des gesamten Zulassungsergebnisses. Die Prozesscharakterisierung wird in Abb. 8.9 dargestellt.

<i>Immatrikulation</i>			
Prozessart	Hauptprozess	Serviceprozess	Managementprozess
Prozessstruktur	unstrukturiert	semi-strukturiert	strukturiert
Automatisierungsgrad	nicht automatisiert	teilautomatisiert	vollautomatisiert
Entscheidungsintensität	niedrig	mittel	hoch
Prozessverantwortung	intra-organisational		inter-organisational
Verhaltensflexibilität	niedrig	mittel	hoch
Strukturflexibilität	niedrig	mittel	hoch

Abb. 8.9: Charakterisierung des Immatrikulationsprozesses an Hochschulen

Der Hauptprozess kann als semi-strukturiert bezeichnet werden. Die Ablaufsteuerung zur Durchführung von beispielsweise Vollständigkeitsprüfung, Zulassungsprüfungen oder zur Ermittlung des Gesamtergebnisses kann automatisiert werden, die einzelnen Prüfungen erfolgen fallspezifisch durch Mitarbeiter. Die Entscheidungsintensität im Prozessablauf wird lediglich als *mittel* bewertet und spiegelt sich als solches auch in der Verhaltensflexibilität wieder. Zwar sind zahlreiche Entscheidungen bzgl. der Zulassung von Studierenden zu treffen, diese Entscheidungen erfolgen jedoch nicht auf Grundlage eines gpDWH-Systems. Hingegen stellen das Erkennen von Verzögerung und die Ableitung von Ressourcenanpassungen oder auch die Anpassung der Prozessstruktur Anwendungsfälle eines gpDWH-Systems dar, um Prozesslaufzeiten und die Fehlerfreiheit im Prozess zu verbessern.

Logistikprozess von Express- und Paketdienstleistern Von Vogt et al. (2008) wird als verhaltensorientierter Anwendungsfall der *Logistikprozess* von Kurier-, Express- und Paketdienste (KEP) vorgeschlagen. Nachfolgend wird aber auch das strukturorientierte Szenario aufgegriffen. Eine Analyse des KEP-Marktes wird durch den *Bundesverband Paket & Express Logistik* in der *KEP-Studie 2015* (Bundesverband Paket & Express Logistik 2015) vorgestellt. Als Beispiel wird das *Business-to-Customer-Geschäft* (B2C) herausgegriffen. Getrieben durch das starke Wachs-

tum im Online-Handel ist auch das B2C-Segment im starken Wachstum (bzgl. Sendungsvolumen und Umsatz) begriffen und stellt eine Schlüsselfunktion für Online-Händler dar. Liefermodelle mit definierten Lieferfenstern, verbindlichen Laufzeiten (*over night, same day delivery*) setzen voraus, dass die Dienstleister ihre Prozesse und Ressourcen effizient gestalten und steuern sowie auf Markttrends (z.B. *same day delivery*) anpassen können (Bundesverband Paket & Express Logistik 2015).

Logistikprozess von Express- und Paketdienstleistern			
Prozessart	Hauptprozess	Serviceprozess	Managementprozess
Prozessstruktur	unstrukturiert	semi-strukturiert	strukturiert
Automatisierungsgrad	nicht automatisiert	teilautomatisiert	vollautomatisiert
Entscheidungsintensität	niedrig	mittel	hoch
Prozessverantwortung	intra-organisational		inter-organisational
Verhaltensflexibilität	niedrig	mittel	hoch
Strukturflexibilität	niedrig	mittel	hoch

Abb. 8.10: Charakterisierung des Logistikprozesses von KEP-Dienstleistern

In Abb. 8.10 wird der Hauptprozess charakterisiert. Für den Prozess vom Online-Händler hin zum Kunden wird die vollständige Strukturierung angenommen, jedoch erfolgt die Durchführung teilautomatisiert (z.B. Boten als personelle Aufgabenträger). Zahlreiche komplexe Entscheidungen (hohe Entscheidungsintensität), beispielsweise zur Ressourcenanpassung im Lieferprozess sind auf Basis eines gpDWH-Systems zu treffen. Die Prozessverantwortlichkeit liegt beim Logistikdienstleister, wenngleich auch externe Sub-Unternehmer in die Prozessanalyse eingebunden werden können. Sowohl die Verhaltens- als auch die Strukturflexibilität werden als hoch eingestuft. Zum einen sind Anforderungen von Auftraggebern und Kunden (z.B. verbindliche Lieferzeiten, Reaktion auf abweichende Empfängeranschriften) zwingend einzuhalten. In einem intensiven Wettbewerb zwischen den Dienstleistern sind Verzögerungen im Prozess rechtzeitig zu erkennen sowie entsprechende Reaktionen notwendig. Zum anderen ist die ständige Analyse der Prozesse sowie deren effizientere Gestaltung unter Berücksichtigung von Markt- und Kundenanforderungen notwendig.

Leistungsprozess eines Call Centers Call Center, wie auch bereits im Fallbeispiel *Mobility Provision* als Vertriebskanal eingebunden, werden bei Hrach & Alt (2012)

als Anwendungsfall von gpDWH-Systemen genannt. Das Marktumfeld von Call-Center-Betreibern, als Dienstleister für telefonische Kundeninteraktion, wird als wettbewerbsintensiv beschrieben. Der Prozess (vgl. Abb. 8.11) kann a priori vollständig strukturiert werden, wird aber lediglich teilautomatisiert durchgeführt. Handelnden Agenten kommt hier eine zentrale Rolle bzgl. des erfolgreichen Gesprächsabschlusses vor. *Abschlussquoten* oder *durchschnittliche Gesprächszeiten* (ggf. in unterschiedlichen Gesprächsphasen) stellen mögliche Kennzahlen zur Bewertung der Prozessqualität und zur Aufdeckung von Mängeln dar.

Dienstleistungsprozess eines Call Center			
Prozessart	Hauptprozess	Serviceprozess	Managementprozess
Prozessstruktur	unstrukturiert	semi-strukturiert	strukturiert
Automatisierungsgrad	nicht automatisiert	teilautomatisiert	vollautomatisiert
Entscheidungsintensität	niedrig	mittel	hoch
Prozessverantwortung	intra-organisational		inter-organisational
Verhaltensflexibilität	niedrig	mittel	hoch
Strukturflexibilität	niedrig	mittel	hoch

Abb. 8.11: Charakterisierung des Dienstleistungsprozesses von Call Centern

Der grundsätzliche Gesprächsablauf wird durch ein AwS mit Gesprächsschema vorgegeben. Die Kommunikation selbst erfolgt manuell durch einen Agenten. Zum einen stellen interne Team- oder Projektleiter potenzielle gpDWH-Nutzer zur Bewertung der Mitarbeiter-Perfomanz dar. Zum anderen sind auch Auftraggeber des Call Centers (z.B. *e-Car Net* im Fallbeispiel *Mobility Provision*) an der zeitnahen Auswertung der Prozessperfomance interessiert. Mögliche Zielsetzungen stellen die Analyse von Wartezeiten, Abschlussquoten oder Gesprächsdauern gegliedert nach Gesprächsphasen dar. Als Reaktionen sind sowohl die Anpassung von Gesprächsprozessen (Strukturflexibilität) als auch Eingriffe in Gesprächsabläufe oder Anpassung von Ressourcen zur Gesprächsführung (Verhaltensflexibilität) denkbar.

9 Reflexion und Ausblick

Kapitelzusammenfassung: Abschließend werden die Hypothesen der Arbeit aus Kap. 1.3 reflektiert sowie ein Ausblick auf mögliche künftige Forschungsaktivitäten gegeben. Dies erfolgt jeweils spezifisch für die einzelnen Hypothesen in den Abschn. 9.1 bis 9.5. Die Reflexion basiert auf jeweils relevanten Ergebnissen der Arbeit und stellt die Grundlage zur Identifikation offener Aspekte dar. In Abschn. 9.6 werden die Erkenntnisse der Reflexion mit Bezug zur Problemstellung der Arbeit abschließend zusammengefasst.

9.1 Komponentenorientierung als Architekturparadigma für operativ-dispositive Anwendungssysteme

Gegenstand der Reflexion ist nachfolgende Hypothese der Arbeit (vgl. Kap. 1.3): *Die Komponentenorientierung stellt ein geeignetes Paradigma zur Strukturierung operativ-dispositiver Anwendungssysteme dar.*

Komponentenorientierung meint die Strukturierung von AwS anhand von wiederverwendbaren und abgeschlossenen Komponenten, die Funktionen über eine wohldefinierte Schnittstelle zur Verfügung stellen (vgl. Kap. 2.3.1). Zur Realisierung der übergeordneten AwS-Funktionalität sind Komponenten lose gekoppelt und interagieren über ihre Schnittstellen. Als zentraler Grundsatz bei der Abgrenzung von Komponenten eines AwS gilt die *Maximierung der Kohäsion innerhalb einer Komponente bei gleichzeitiger Minimierung der Kopplung zwischen diesen*. Die Ausgestaltung dieser Faktoren hat maßgeblichen Einfluss auf die Flexibilität bzw. Evolutionsfähigkeit eines AwS und damit auf die in Kap. 2.3.1 vorgestellten übergeordneten Ziele der Wiederverwendbarkeit, Portierbarkeit, Kosteneffektivität und Wartbarkeit. Die flexible Anpassbarkeit an sich stetig verändernde Geschäftsprozesse, durch innovativere Gestaltung oder neue Nutzeranforderungen, stellt auch eine zentrale Anforderung an operativ-dispositive AwS dar (vgl. 5.1). Die Eignung der Komponentenorientierung für diese Art AwS wird deshalb anhand der Faktoren *Kohäsion* und *Kopplung* bewertet.

In Kap. 5.3.3 werden, basierend auf der *flexiblen Data-Mart-Architektur*, abgeschlossene Komponententypen operativ-dispositiver AwS vorgestellt. Der Gegenstand *operativer Datenkomponenten* definiert sich über die fachlichen Funktionen zur Verwaltung zusammengehöriger Daten. *Data-Mart-Komponenten* sind anhand der nachfrageseitigen Datenbereitstellung, Komponenten zur *Datenintegration* anhand ihres Versorgungsauftrages abgegrenzt. EU-Komponenten sind über ihre Entscheidungsfunktion definiert. Die *Kohäsion* innerhalb dieser Komponenten definiert sich anhand des jeweils gültigen Kriteriums. Die *Kopplung* zwischen den Komponenten wird als minimal angesehen. Hier bestehen klar definierte Kommunikationsbeziehungen über Schnittstellen mit, wenn möglich, generischen Funktionssignaturen. Eine konkrete Parametrisierung erfolgt erst zur Laufzeit. Im Falle von Anpassungen am operativen Prozess werden dadurch gerade im pDWH-System Anpassungen an Schnittstellen minimiert. Die gewählte Ausgestaltung der Faktoren *Kohäsion* und *Kopplung* ist zur Realisierung evolutionsfähiger operativ-dispositiver AwS geeignet. Die abgegrenzten, wiederverwendbaren Komponententypen wirken sich positiv auf die Wartbarkeit des Gesamtsystems aus. Als weitere Anforderungen an die Referenzarchitektur wird in Kap. 5.1 die Informationsversorgung verschiedener Nutzergruppen sowie der bidirektionale Datenaustausch zwischen operativem und dispositivem Teilsystem genannt. Gerade zu Letzterem trägt die Komponentenorientierung durch das Prinzip der losen Kopplung sowie der Interaktion über wohldefinierte Schnittstellen bei. In Kap. 7.6.2 werden Spezialisierungen des verhaltensorientierten pDWH-Szenarios vorgestellt. Gerade zur Nutzung dispositiver Dienste in operativen Prozessen sind lose gekoppelte, wohldefinierte Schnittstellen geeignet. Dies kann zusätzlich auch durch verschiedene Ansätze aus dem verwandten SOA-Kontext unterstrichen werden (vgl. Kap. 5.2).

Zusammenfassend kann die Hypothese bestätigt werden. Die Komponentenorientierung ist ein geeignetes Systemparadigma für operativ-dispositive AwS und dessen Strukturierung anhand der Markt-Metapher von Angebot und Nachfrage. Es ergeben sich jedoch auch offene Fragestellungen an den Grenzen der Arbeit. Die Referenzarchitektur stellt einen fachlichen Rahmen dar, den es durch entsprechende Technologieentscheidungen auszugestalten gilt. Die Auswahl geeigneter Technologien liefert Raum für künftige Arbeiten. Dies betrifft zum einen die Auswahl von Komponententechnologien und die Abbildung bzw. Realisierung

des Fachkonzeptes. Ansatzpunkte bieten hier beispielsweise JavaEE³ oder Microsoft .NET⁴. Zum anderen sollten geeignete Technologien zur Realisierung niedriger Latenzen von der Datenbereitstellung bis hin zur Reaktion evaluiert werden. Das Hadoop-Ecosystem⁵ stellt hier einen vielversprechenden Ansatz gerade für operational-BI-Anwendungen dar. Eine weitere Abgrenzung stellen intra-organisatorische Geschäftsprozesse dar. Die Verantwortung für Prozesse, der Zugriff auf Ablaufinformationen sowie deren Syntax und Semantik liegen innerhalb der Grenzen des betrieblichen Systems. Mit Ausweitung der Reichweite eines pDWH-Systems bzw. des operativ-dispositiven AwS auf mehrere an einem Geschäftsprozess beteiligte Akteure entstehen neue Herausforderungen und Fragestellungen bzgl. der angesprochenen Aspekte.

9.2 Die betriebliche Aufgabe als Konzept zur geschäftsprozessorientierten Einordnung von Workflow-Abläufen

Gegenstand der Reflexion in diesem Abschnitt bildet folgende Hypothese (vgl. Kap. 1.3): *Über das Konzept der betrieblichen Aufgabe können aus pDWH-Systemen abgeleitete Informationen auch im Kontext der betrieblichen Leistungserstellung bewertet werden.*

Die *betriebliche Aufgabe* stellt ein Konzept zur Abgrenzung und Beschreibung von Aufgaben im Rahmen eines Geschäftsprozesses dar. Aus Außensicht ist diese über Vor- und Nachereignisse, Zielvorgaben sowie das Aufgabenobjekt beschrieben. Die Innensicht beschreibt das Lösungsverfahren zur Aufgabendurchführung durch einen personellen oder automatisierten Aufgabenträger (vgl. Kap. 2.1.2). Die Bewertung der betrieblichen Aufgabe als Brücke zwischen AT-spezifischen Informationen und deren Einordnung in den Kontext der betrieblichen Leistungserstellung ist im Wesentlichen von zwei Aspekten abhängig: Zunächst wird das Potenzial des Konzeptes der *betrieblichen Aufgabe* über deren Bestandteile bewertet. Ein Geschäftsprozess wird als Netz betrieblicher Aufgaben verstanden, die über Vor- bzw. Nachereignisse in Beziehung stehen. Ein Workflow hingegen beschreibt

³<http://www.oracle.com/technetwork/java/javaee/overview/index.html> – Zugriff: 29.07.2017

⁴<https://www.microsoft.com/net> – Zugriff: 29.07.2017

⁵<http://hadoop.apache.org/> - Zugriff: 29.07.2017

die Realisierung ausgerichtet auf einen AT-Typ (vgl. Kap. 2.2.1). Dieser Wechsel zwischen Aufgaben- und AT-Perspektive wird durch den Wechsel zwischen Außen- und Innensicht auf eine betriebliche Aufgabe unterstützt. Eine Aufgabe eines Geschäftsprozesses wird zur Abgrenzung relevanter Daten im AT-spezifischen DWH-Modell genutzt. Hierzu werden Ereignisse, Sach- und Formalziele oder auch das betriebliche Aufgabenobjekt herangezogen. Die darauf basierende Bildung aufgabenorientierter Sichten wird in Kap. 7.5.2 beschrieben. Die SOM-Methodik unterstützt explizit das Konzept der *betrieblichen Aufgabe*. ARIS benennt dieses zwar nicht explizit, mit Funktionen der EPK wird jedoch ein vergleichbares Konzept unterstützt (vgl. Kap. 8.2.2). Ein weiterer Aspekt zur Bewertung der *betrieblichen Aufgabe* als Brücke zwischen AT-Ebene und betrieblicher Leistungserstellung ist die methodische Einbettung. In Kap. 4.1 werden mit Lenkungs-, Leistungs- und Ablaufsicht drei Sichten auf einen Geschäftsprozess vorgestellt. Die Unterstützung dieser Sichten durch die übergeordnete Methodik zur Systementwicklung ist dabei für die Einordnung in den Kontext der betrieblichen Leistungserstellung ausschlaggebend. Durch die Ablaufsicht werden pfad- oder abschnittsbezogene Analysen ermöglicht. Die Bewertung der Ergebnisse vor dem Hintergrund der Gesamtaufgabe erfolgt entlang von Lenkungs- und Leistungssicht (vgl. Kap. 7.5.2).

Die *betriebliche Aufgabe* stellt damit, unter der Voraussetzung, dass diese in eine entsprechende Methodik zur GP-orientierten Systementwicklung eingebettet ist, das zentrale Konzept zur Einordnung technischer Ablaufinformationen in den Kontext der betrieblichen Leistungserstellung dar. Auf dieser Basis erfolgt die Erweiterung eines prozessorientierten hin zu einem GP-orientierten DWH-System. In Kap. 7.5.2 wird die Bildung aufgabenorientierter Sichten sowie deren verhaltens- und strukturorientierte Einordnung in den GP-Kontext beschrieben. Die Unterstützung von Entwicklern und gpDWH-Nutzern durch integrierte Werkzeuge zur Prozessmodellierung und -analyse ist aufgrund der Gesamtkomplexität des Aufgabenbereiches notwendig. Neben der Konsistenz von operativem und dispositivem Teilsystem kann die von Becker & Chamoni (2008) vorgeschlagene Visualisierung von Analyseergebnissen in Prozessmodellen, aber auch die Bildung aufgabenorientierter Sichten sowie der Wechsel zwischen Aufgaben- und Aufgabenträgerperspektive unterstützt werden. Konzeption und Entwicklung entsprechender Werkzeuge auf Basis des Konzeptes der *betrieblichen Aufgabe* stellt einen vielversprechenden Ansatz für künftige Arbeiten dar.

9.3 Das multidimensionale Datenmodell als Grundlage zur nachfrageorientierten Datenbereitstellung

Im Weiteren wird die nachfolgende Hypothese reflektiert (vgl. Kap. 1.3): *Das für themenorientierte DWH-Systeme etablierte multidimensionale Datenmodell ist auch zur Realisierung prozessorientierter DWH-Systeme anwendbar.*

Das multidimensionale Datenmodell wird in themenorientierten DWH-Systemen zur Datenbereitstellung angewendet. Es ermöglicht ad-hoc Analysen quantitativer Kennzahlen entlang qualitativer Dimensionen (vgl. Kap. 3.3). Mit der umfassenden Abbildung des Prozessablaufes sowie der sich stetig verändernden Prozessspezifikationen ergeben sich zwei Herausforderungen, die es bei der Anwendung in pDWH-Systemen zu berücksichtigen gilt. Auch in themenorientierten DWH-Systemen sind Veränderungen der inhaltlichen und technischen Anforderungen im Zeitverlauf bekannt (Bauer & Günzel 2013, S. 228). Vor der übergeordneten Zielsetzung der stetig effizienteren und innovativeren Gestaltung von Geschäftsprozessen nach Becker & Kahn (2012, S. 1f.) gilt diese Herausforderung in besonderem Maße auch für die Gestaltung von pDWH-Strukturen (vgl. Kap. 6.2). In Kap. 6.4 wird ein generisches multidimensionales Datenschema unter Berücksichtigung beider Herausforderungen vorgestellt. Der Ablauf von BPMN-Workflows kann umfassend (anhand der Prozessperspektiven *Funktion, Verhalten, Organisation, Information*) abgebildet werden. In Kap. 8.2.1 wird zudem die Übertragung auf andere Workflow-Sprachen thematisiert. Aufgrund seines generischen Charakters kann das vorgestellte Datenschema zudem auch als stabil im Falle angepasster Prozessabläufe angesehen werden.

Neben der Ausgestaltung des pDWH-Schemas ist auch die Verortung innerhalb der DWH-Architektur zur Bewertung der Anwendbarkeit notwendig. Mit der angebots- und der nachfrageorientierten Datenbereitstellung bietet die *flexible Data-Mart-Architektur* zwei Ansatzpunkte zur Anwendung des multidimensionalen Datenmodells. Die Gestaltung multidimensionaler Datenschemata themenorientierter DWH-Systeme erfolgt ausgerichtet auf Entscheidungsfelder (*subject-oriented*, vgl. Kap. 3.1.2). Mit dem Konzept der Data Marts und der nachfrageorientierten Datenbereitstellung findet sich eine Analogie in der *flexiblen Data-Mart-Architektur*. Die angebotsorientierte Datenbereitstellung erfolgt nicht ausgerich-

tet an Entscheidungsfeldern, sondern auf Basis zur Verfügung stehender Datenstrukturen. Das multidimensionale Schema wird entlang seiner Kennzahlen und Dimensionen fragmentiert und operativen Komponenten zugeordnet (vgl. Kap. 7.3). Die Allokation erfolgt anhand der verwalteten operativen Datenstrukturen. Hierzu ist nicht zwingend ein multidimensionaler Datensatz erforderlich, wie auch die Anwendung des *Data-Vault-Ansatzes* als Grundlage der angebotsorientierten Datenbereitstellung und Integration in Kap. 8.2.3 zeigt.

Zusammenfassend bleibt festzuhalten, dass die Anwendung des multidimensionalen Datenmodells gerade zur nachfrageorientierten Datenbereitstellung zu empfehlen ist. Auch zur angebotsorientierten Datenbereitstellung ist das multidimensionale Datenmodell anwendbar, wenn auch nicht zwingend notwendig. Gerade mit dem *Data-Vault-Ansatz* findet sich hier eine Alternative. Für Empfehlungen sind jedoch weitere praktische Erfahrungen zur *flexiblen Data-Mart-Architektur* und deren Datenstrukturen notwendig. Die eingangs formulierte Hypothese kann grundsätzlich bestätigt werden, die abgeleitete Aussage sollte jedoch präzisiert werden: *Das für themenorientierte DWH-Systeme etablierte multidimensionale Datenmodell ist sowohl zur angebots- als auch zur nachfrageorientierten Datenbereitstellung anwendbar. Gerade zur angebotsorientierten Bereitstellung können jedoch auch alternative, in themenorientierten Systemen etablierte Formen der Datenstrukturierung angewendet werden.*

9.4 Aussagenkonsistenz durch hierarchische Integration prozess- und themenorientierter Data-Warehouse-Systeme

Als Nächstes bildet die nachfolgende Hypothese den Gegenstand der Reflexion (vgl. Kap. 1.3): *Prozessorientierte DWH-Systeme stellen eine konsistente Erweiterung prozessübergreifender themenorientierter DWH-Systeme dar.*

Der Hypothese liegt die Forderung von Baars & Sun (2013) zugrunde, pDWH-Systeme mit unternehmensweiten Systemen abzugleichen, so dass diese hinsichtlich der abgeleiteten Aussagen frei von Widersprüchen sind. Dieser Anforderung nach Konsistenz wird in der Arbeit durch Integration der beiden Systemtypen begegnet. In Kap. 3.5.1 ist die Abgrenzung prozess- und themenorientierter DWH-

Systeme anhand von Granularität, Reichweite sowie Prozessbezug beschrieben. Aus dieser leitet sich eine hierarchische Anordnung beider Systeme ab und beschreibt damit auch die Einordnung eines Geschäftsprozesses in das übergeordnete betriebliche System. Das pDWH-System mit feingranularen Prozessdaten bildet eine Vorstufe und damit auch eine Datenquelle des prozessübergreifenden, themenorientierten DWH-Systems. Die Spezifikation der Integration erfolgt auf Basis der Schichten der *flexiblen Data-Mart-Architektur*. Hieraus resultieren zwei grundsätzliche Integrationsformen über die angebots- und über die nachfrageorientierte Datenbereitstellungsschicht (vgl. Kap. 5.5). Unabhängig von der gewählten Form ist durch die Integration mit angemessener Aktualisierungsstrategie eine konsistente Datenbasis in den DWH-Systemen und damit auch die Widerspruchsfreiheit in daraus abgeleiteten Aussagen möglich. Grundsätzlich ist auch eine nicht-hierarchische Form der Integration nicht ausgeschlossen, wenn z.B. mit alleiniger Ausrichtung auf das strukturorientierte Szenario von der *flexiblen Data-Mart-Architektur* abgewichen wird. Eine *Hub-and-Spoke-Architektur* (vgl. Sinz & Ulbrich-vom Ende 2010) stellt eine alternative Form zur Sicherstellung der Konsistenz dar. Als problematisch könnten sich jedoch die unterschiedliche Granularität der Daten im zentralen DWH erweisen. Die eingangs beschriebene Hypothese wird damit bestätigt.

Bei Bauer & Günzel (2013, S. 345) wird neben der Konsistenz als Kriterium der Datenqualität auch die Nachvollziehbarkeit genannt. Gemeint ist damit die Nachvollziehbarkeit des Weges eines Datenelementes vom Quellsystem bis hin ins multidimensionale Schema der Auswertungsdatenbank. Damit wird die Überprüfung der Genauigkeit sowie die Rekonstruierbarkeit der Daten ermöglicht. Eine Voraussetzung stellt die Aufzeichnung entsprechender Metadaten dar (Bauer & Günzel 2013, S. 345). Hierzu bilden die Modellarchitektur der Entwicklungsmethodik (Kap. 7.3) sowie die komponentenorientierte pDWH-Struktur eine geeignete Grundlage. Die Aufzeichnung entsprechender Daten würde die Nachvollziehbarkeit von Daten aus dem themenorientierten DWH-System über das pDWH-System hin zur operativen Quelle ermöglichen und einen Beitrag zur Konsistenzsicherung leisten. Eine entsprechende Erweiterung durch künftige Arbeiten erscheint sinnvoll.

9.5 Das Geschäftsprozessmodell als Grundlage zur Erweiterung operativer Anwendungssysteme um dispositive Funktionen

Abschließend erfolgt die Reflexion der nachfolgenden Hypothese (vgl. Kap. 1.3): *Bestehende Ansätze der geschäftsprozessorientierten Systementwicklung können zur Entwicklung operativ-dispositiver Anwendungssysteme erweitert werden.*

In Forschung und Praxis haben sich zahlreiche Methodiken zur GP-orientierten Systementwicklung etabliert. In der Arbeit werden sowohl das *Semantische Objektmodell* (SOM) als auch die *Architektur integrierter Informationssysteme* (ARIS) aufgegriffen. Diese Methoden können und sollten anhand allgemeiner Konzepte zur Entwicklung operativ-dispositiver Anwendungssysteme erweitert werden. Das vorgestellte Entwicklungsvorgehen wird entlang der Markt-Metapher von Angebot und Nachfrage zeitlich entkoppelt und in Form eigenständiger Modellarchitekturen beschrieben (vgl. Kap. 7). Voraussetzung für die Anwendbarkeit der Entwicklungsmethodik ist die Verwendung des GP-Modells mit betrieblichem Aufgabenmodell (vgl. Reflexion in Abschn. 9.2). Als Kriterien zur Bewertung der Eignung einer Entwicklungsmethodik (vgl. Kap. 8.2.2) werden die modellbasierte Beschreibung des AwS auf Aufgaben- und Aufgabenträgerebene, der durchgehende Bezug zum GP-Modell sowie das hierarchische Vorgehen zur Spezifikation von Struktur und Verhalten des GP genannt. Über die Ausprägung dieser Merkmale bestimmt sich der Grad der Anwendbarkeit des gpDWH-Konzeptes. Die Ergebnisse der Arbeit basieren grundsätzlich auf der SOM-Methodik. Die Tragfähigkeit wird jedoch auch durch die Anwendung auf ARIS in Kap. 8.2.2 nachgewiesen. So kann die Hypothese der Erweiterbarkeit bestehender Methoden bestätigt werden.

Die vollständige Spezifikation von Geschäftsprozessen oder auch die Ableitung von AwS-Spezifikationen über mehrere Modellebenen stellen Grundannahmen dieser Arbeit dar. Künftige Fragestellungen sollten die Anwendbarkeit der einzelnen Konzepte (Architektur, Datenmodell, Entwicklungsmethodik) beleuchten, wenn Prozessmodelle nicht explizit oder nur mit hohem Abstraktionsgrad vorliegen. Potenziale des gpDWH-Ansatzes sollten auch in semi- oder unstrukturierten Prozessumgebungen nutzbar sein. Mitunter kann das gpDWH-System auch zur Explikation und besseren Strukturierung in diesen Umgebungen beitragen.

9.6 Zusammenfassung

Gegenstand der Arbeit bildet ein Konstruktionsproblem, zu dessen Lösung eine Referenzarchitektur operativ-dispositiver AwS einschließlich Datenschema sowie eine zugehörige Entwicklungsmethodik bereitgestellt werden. Der Beitrag dieser Artefakte zur Problemlösung wird anhand der aufgezeigten Forschungslücken bzgl. Architektur, Prozessperspektive und Entwicklungsmethodik (vgl. Kap. 1.1.3) zusammengefasst: Durch die angesprochene Referenzarchitektur wird, basierend auf dem komponentenorientierten Paradigma, die Integration operativer und dispositiver Systeme sowie deren Realisierung als ganzheitlicher Ansatz beschrieben. Sowohl das verhaltens- als auch das strukturorientierte pDWH-Szenario finden Berücksichtigung. Einen wesentlichen Beitrag bzgl. der Verringerung der Lücke zur Prozessperspektive liefert zum einen das multidimensionale Datenmodell. Es ermöglicht die umfassende Bereitstellung von Workflow-Daten kombiniert mit Datenstrukturen der Fachdomäne. Zum anderen wird durch die Differenzierung über das betriebliche Aufgabenmodell fachlich- und technischerorientierten Nutzern Zugang zu analyserelevanten Daten ermöglicht und die differenzierte Betrachtung aus den Perspektiven von Workflow und Geschäftsprozess erreicht. Mit der vorgestellten Entwicklungsmethodik wird schließlich ein wesentlicher Beitrag zur bislang sehr eingeschränkt berücksichtigten und komplexen Entwicklung GP-orientierter operativ-dispositiver AwS geleistet. Die Ergebnisartefakte müssen nicht zwingend in Kombination Anwendung finden. Sie können auch vom ganzheitlichen gpDWH-Ansatz losgelöst genutzt werden.

Die zur Lösung des Untersuchungsproblems zugrunde gelegten Hypothesen werden in den Abschn. 9.1 bis 9.5 weitgehend bestätigt, an wenigen Stellen präzisiert. Aus den Hypothesen abgeleitete zukünftige, noch offene Fragestellungen betreffen zum einen die technische Realisierung, zum anderen die praktische Anwendung in heterogenen Domänen. Von beiden Aspekten wird eine Rückkopplung auf die Gesamtkonzeption erwartet. Die Voraussetzungen zur Anwendung sind mit Beispielen und Handlungsempfehlungen geschaffen. Die zwingend notwendige praktische Anwendungserfahrung kann jedoch nur mittelfristig in möglichst vielen Anwendungsfällen im Nachgang dieser Arbeit erworben werden.

Anhang

A.1 Relationale Realisierung des prozessorientierten Data- Warehouse-Schemas

In Code A.1 ist die relationale Implementierung der prozessspezifischen Dimensionstabellen basierend auf dem Datenbankverwaltungssystem PostgreSQL⁶ (Version 9.4.) als *Star-Schema* dargestellt. Zunächst werden die Schema-spezifischen Tabellen *FlowNodeSchema* und *FlowNodeTyp*, im Anschluss die Instanz-spezifischen Tabellen *FlowNodeSequenz* und *Participant* angegeben.

```
1 -- Schema-spezifische Tabellen (Dimensionen):
2 -- Dimension: FlowNodeSchema
3 create table flownodeschema(
4 nodeid int primary key,
5 nodedesc text,
6 parent int references flownodeschema (nodeid) );
7
8 -- Dimension: FlowNodeTyp
9 create table flownodetyp(
10 typeid char(3) primary key,
11 subtyp text,
12 charakter text,
13 typ text );
14
15 -- Instanz-spezifische Tabellen (Dimensionen)
16 -- Dimension: FlowNodeSequence
17 create table flownodesequenz(
18 nodeinstance int primary key,
19 prenode int references flownodesequenz (nodeinstance),
20 postnode int references flownodesequenz (nodeinstance),
21 postcondition text,
22 processinstance int );
23
24 -- Dimension: Participant
25 create table participant(
26 participantid char(3) primary key,
27 partinstance text,
28 partmessage text,
29 parttyp text );
```

Code A.1: SQL-Implementierung prozessspezifischer Dimensionstabellen

⁶<https://www.postgresql.org/> - Zugriff: 29.07.2017

In Code A.2 werden exemplarisch die Dimensionstabellen *Vertriebsart* und *Geographie* der betrieblichen Domäne spezifiziert.

```
1 -- Domänen-spezifische Tabellen (Dimensionen)
2 -- Vertriebsart
3 create table dim_vertriebsart(
4 vartid int primary key,
5 vartdesc text );
6
7 -- Geographie
8 create table dim_geographie(
9 ortid int primary key,
10 ortdesc text,
11 regionid int,
12 regiondesc text,
13 landid int,
14 landdesc text );
```

Code A.2: SQL-Implementierung domänenspezifischer Dimensionstabellen

Code A.3 zeigt die Implementierung der Faktentabelle mit Zusammenführung der prozess- und domänenspezifischen Dimensionstabellen über deren Primärschlüssel.

```
1 -- Faktentabelle
2 create table fakten (
3 nodeinstance int references flownodesequenz(nodeinstance),
4 nodeid int references flownodeschema(nodeid),
5 typeid char(3) references flownodetyp (typeid),
6 participantid char(3)
7     references participant(participantid),
8 vartid int references dim_vertriebsart(vartid),
9 ortid int references dim_geographie(ortid),
10
11 n int,
12 c decimal(10,2),
13 t int,
14
15 primary key (nodeinstance, nodeid, participantid, typeid,
16             vartid, ortid) );
```

Code A.3: SQL-Implementierung der Faktentabelle

A.2 Implementierung aufgabenorientierter Sichten

In Code A.4 ist die SQL-basierte Implementierung der aufgabenorientierten Sicht zur Bewertung der Zeiteffizienz angegeben. Zunächst wird die Gesamtzeit aller Prozessinstanzen einer Vertriebsart, im Anschluss deren Anzahl je Vertriebsart berechnet. Im letzten Schritt werden die Teilergebnisse zur Berechnung der Kennzahl *durchschnittliche Laufzeit* zusammengeführt. Über die Selektionsbedingung wird auf die Aktivität *Empfang Verfügbarkeitsliste* (node-id = 3) eingeschränkt. Auf eine rekursive Implementierung bzgl. beliebig tiefer Schachtelung von Sub-Prozessen wird verzichtet.

```

1 CREATE VIEW zeiteffizienz (
2 WITH
3
4 % 1) Berechnung der Gesamtzeit nach Vertriebsart
5 summezeit AS (
6 SELECT vart.vartdesc, sum(t) as zeit
7 FROM fakten f, flownodeschema schema, flownodetyp typ,
8     participant par, flownodesequenz seq, dim_vertriebsart vart,
9     dim_geographie geo
10 WHERE f.nodeid = schema.nodeid
11     AND f.typeid = typ.typeid
12     AND f.nodeinstance = seq.nodeinstance
13     AND f.participantid = par.participantid
14     AND f.ortid = geo.ortid
15     AND f.vartid = vart.vartid
16     AND parent = 3
17 GROUP BY vart.vartdesc),
18
19 % 2) Berechnung Anz. der Prozessinstanzen nach Vertriebsart
20 anzahl AS (
21 SELECT vart.vartdesc, count(*) as anzahl
22 FROM fakten f, flownodeschema schema, flownodetyp typ,
23     participant par, flownodesequenz seq, dim_vertriebsart vart,
24     dim_geographie geo
25 WHERE f.nodeid = schema.nodeid
26     AND f.typeid = typ.typeid
27     AND f.nodeinstance = seq.nodeinstance
28     AND f.participantid = par.participantid
29     AND f.ortid = geo.ortid
30     AND f.vartid = vart.vartid
31     AND schema.nodeid = 3
32 GROUP BY vart.vartdesc )
33
34 % 3) Berechnung der abgeleiteten Kennzahl durchschnittliche
35     Laufzeit nach Vertriebsart für Aktivität 3
36 SELECT a.vartdesc as Vertriebsart, s.zeit/a.anzahl as
37     durchschnittl_laufzeit
38 FROM anzahl a, summezeit s
39 WHERE a.vartdesc = s.vartdesc)

```

Code A.4: SQL-Implementierung der aufgabenorient. Sicht *Zeiteffizienz*

In Code A.5 ist eine beispielhafte Implementierung der aufgabenorientierten Sicht zur Bewertung der Sachzielerreichung der Aufgabe *Empfang Verfügbarkeitsliste* (node-id = 3) angegeben. Differenziert wird anhand der Dimension *Geographie* zur Identifikation regionaler Unterschiede. Zunächst werden die unterschiedlichen Endzustände der Aktivität, aufgeschlüsselt nach Orten, berechnet. Im zweiten Schritt wird die Gesamtzahl der Aufgabendurchführungen nach Orten berechnet, bevor im dritten Schritt der prozentuale Anteil unterschiedlicher Endzustände ermittelt wird. Auf eine Implementierung der rekursiven Schachtelung wird ebenfalls verzichtet.

```

1 CREATE VIEW sachzielerreichung AS (
2 WITH
3
4 % 1) Berechnung der Anz. unterschiedlicher Endzustände nach Orten
5 endzustand AS (
6 SELECT geo.ortdesc, schema.nodedesc, count(*) as anz_endzustände
7 FROM fakten f, flownodeschema schema, flownodetyp typ,
8     participant par, flownodesequenz seq, dim_vertriebsart vart,
9     dim_geographie geo
10 WHERE f.nodeid = schema.nodeid AND f.typeid = typ.typeid
11     AND f.nodeinstance = seq.nodeinstance
12     AND f.participantid = par.participantid
13     AND f.ortid = geo.ortid AND f.vartid = vart.vartid
14     AND typ.subtyp = 'end_event' AND schema.parent = 3
15 GROUP BY schema.nodedesc, geo.ortdesc),
16
17 % 2) Berechnung der Aufgabendurchführungen nach Orten
18 gesamt AS (
19 SELECT geo.ortdesc, count(*) as anz_endzustände
20 FROM fakten f, flownodeschema schema, flownodetyp typ,
21     participant par, flownodesequenz seq, dim_vertriebsart vart,
22     dim_geographie geo
23 WHERE f.nodeid = schema.nodeid AND f.typeid = typ.typeid
24     AND f.nodeinstance = seq.nodeinstance
25     AND f.participantid = par.participantid
26     AND f.ortid = geo.ortid AND f.vartid = vart.vartid
27     AND typ.subtyp = 'end_event' AND schema.parent = 3
28 GROUP BY geo.ortdesc)
29
30 % 3) Berechnung des prozent. Anteils von Endzuständen nach Orten
31 SELECT e.ortdesc, e.nodedesc as Zustand, round(cast(e.
32     anz_endzustände as decimal(4,2))/cast(g.anz_endzustände as
33     decimal(4,2)),2)*100 as ProzentualerAnteil
34 FROM gesamt g, endzustand e
35 WHERE g.ortdesc = e.ortdesc )

```

Code A.5: SQL-Implementierung der aufgabenorient. Sicht *Sachzielerreichung*

A.3 Zerlegungsprotokoll zur Erweiterung der Fallstudie *Mobility Provision*

Abb. A.1 zeigt das Zerlegungsprotokoll zur erweiterten Fallstudie *Mobility Provision*. Hervorgehoben sind Transaktionen bzw. betriebliche Objekte, die sich aufgrund der Erweiterung ergeben. Die Erweiterung beschreibt die Information bzw. das Unterbreiten erweiterter Angebote gegenüber dem Kunden während der Ausleihphase. Das betriebliche Objekt *Auftragsverwaltung* wird hierzu in einen *Verwaltungs-* sowie einen *Kommunikationsdienst* zerlegt. Die Leistungen zur *Information* bzw. *Erweiterung* ergeben sich durch Homogenisierung der Gesamtleistung *D: Nutzung_Mobilität*.

<p>TRANSAKTIONSZERLEGUNG:</p> <p>D: Mobilität</p> <p>V: Buchung_Mobilität</p> <p>Vseq: Anfrage</p> <p>Vseq: individuelles Angebot</p> <p>Vseq: Buchung</p> <p>Vseq: Buchungsbestätigung</p> <p>D: Nutzung_Mobilität</p> <p>Dseq: Mobilitätsleistung</p> <p>Dseq: Bereitstellung e-Car</p> <p>V: Anforderung e-Car</p> <p>D: e-Car</p> <p>Dseq: Rückgabe e-Car</p> <p>Dseq: Mobilitätskommunikation</p> <p>D: Informationsleistung</p> <p>D: Erweiterungsleistung</p> <p>V: Erweiterungsvereinbarung</p> <p>V: Erweiterungsangebot</p> <p>V: Erweiterungsbuchung</p> <p>D: Erweiterung</p> <p>Dseq: Mobilitätsabrechnung</p> <p>V: Rechnung</p> <p>D: Zahlung</p>	<p>OBJEKTZERLEGUNG:</p> <p>Mobility Provision</p> <p>Vertrieb</p> <p>Buchungsdurchführung</p> <p>Auftragsverwaltung</p> <p>Verwaltungsdienst</p> <p>Kommunikationsdienst</p> <p>S: Kommunikationsanweisung</p> <p>S: Anweisung_Information</p> <p>S: Anweisung_Angebot</p> <p>K: Meldung_Angebot</p> <p>Finanzwesen</p> <p>D: Zahlungsabwicklung</p> <p>V: Forderung</p> <p>D: Zahlungsmeldung</p> <p>Vertriebskanal</p> <p>D: Buchungsabwicklung</p> <p>V: Buchungsauftrag</p> <p>D: Buchungsdokumentation</p> <p>Mobilitätsleistung</p> <p>Flottenlenkung</p> <p>Flottenbereitstellung</p> <p>S: Anweisung_Bereitstellung</p> <p>K: Rückmeldung_Bereitstellung</p> <p>D: Flottenverwaltung</p> <p>D: Verfügbarkeit</p> <p>V: Verfügbarkeitsprüfung</p> <p>D: Verfügbarkeitsliste</p> <p>D: Vorreservierung</p> <p>V: Auftrag_Vorreservierung</p> <p>D: Bestätigung_Vorreservierung</p> <p>D: Reservierung</p> <p>V: Auftrag_Reservierung</p> <p>D: Bestätigung_Reservierung</p> <p>D: Fahrtenprotokoll</p>
---	---

Abb. A.1: Zerlegungsprotokoll der erweiterten Fallstudie *Mobility Provision*

Literaturverzeichnis

- [Ackermann et al. 2002] Ackermann, Jörg; Brinkop, Frank; Conrad, Stefan; Fetteke, Peter; Frick, Andreas; Glistau, Elke; Jaekel, Holger; Kotlar, Otto; Loos, Peter; Mrech, Heike; Ortner, Erich; Raape, Ulrich; Overhage, Sven; Sahm, Stephan; Schmietendorf, Andreas; Teschke, Thorsten; Turowski, Klaus (2002): *Vereinheitlichte Spezifikation von Fachkomponenten*. Memorandum des Arbeitskreises 5.10.3, Komponententorientierte betriebliche Anwendungssysteme, Gesellschaft für Informatik. URL <http://www.fachkomponenten.de/docs/MEMO/Memorandum-final-2-44-mit-literatur-Web.pdf> – Zugriffsdatum: 29.07.2017.
- [Albani et al. 2008] Albani, Antonia; Overhage, Sven; Birkmeier, Dominik (2008): *Towards a Systematic Method for Identifying Business Components*. In: Chaudron, Michel R. V.; Szyperski, Clemens; Reussner, Ralf (Hrsg.): *Component-Based Software Engineering*. Bd. 5282, Springer, Berlin, S. 262-277.
- [Andresen 2003] Andresen, Andreas (2003): *Komponentenbasierte Softwareentwicklung mit MDA, UML und XML*. Hanser, München.
- [Baars & Sun 2013] Baars, Henning; Sun, Xuanpu (2013): *Wo sind die Klippen im Prozess? Business Process Intelligence - das unterschätzte Potenzial*. In: BI Spektrum, 8(2013), Nr. 4, S. 10-12.
- [Balzert 2008] Balzert, Helmut (2008): *Lehrbuch der Softwaretechnik – Softwaremanagement*. 2. Aufl., Spektrum Akademischer Verlag, Heidelberg.
- [Bange & Fuchs 2011] Bange, Carsten; Fuchs, Christian (2011): *Wege zu prozessorientierter BI*. In: Computerwoche. URL <http://www.cowo.de/a/1905827> – Zugriffsdatum: 29.07.2017
- [Bange et al. 2012] Bange, Carsten; Mack, Melanie; Seidler, Larissa (2012): *Business Intelligence im Mittelstand 2011/2012: Status quo, Ausblick und Empfehlungen*. Business Application Research Center (BARC), Würzburg. URL http://de.cubeware.com/fileadmin/redaktion/dokumente/service/Business_Intelligence_im_Mittelstand_2011-2012.pdf – Zugriffsdatum: 29.07.2017
- [Bartmann et al. 2011] Bartmann, Dieter; Bodendorf, Freimut; Ferstl, Otto K.; Sinz, Elmar J. (2011): *Merkmale, Systemarchitekturen und Management hochflexibler Geschäftsprozesse*. In: Sinz, Elmar J.; Bartmann, Dieter; Bodendorf, Freimut; Ferstl, Otto K. (Hrsg.): *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. Schriften aus der Fakultät für Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg, Bd. 9, Univ. of Bamberg Press, Bamberg, S. 1-13.
- [Bass et al. 2013] Bass, Len; Clements, Paul; Kazman, Rick (2013): *Software Architecture in Practice*. 3. Aufl., Addison-Wesley/Pearson, Upper Saddle River, NJ.
- [Baster et al. 2001] Baster, Greg; Konana, Prabhudev; Scott, Judy E. (2001): *Business components: a case study of bankers trust Australia limited*. In: Commun. ACM, 44(2001), Nr. 5, S. 92-98.

- [Bauer & Günzel 2013] Bauer, Andreas; Günzel, Holger (2013): *Data-Warehouse-Systeme – Architektur, Entwicklung, Anwendung*. 4., überarb. u. erw. Aufl., dpunkt.verlag, Heidelberg.
- [Bauer et al. 1991] Bauer, Friedrich L.; Goos, Gerhard; Dosch, Walter (1991): *Informatik 1 – Eine einführende Übersicht*. 4., verbesserte Aufl., Springer, Berlin.
- [Becker 2007] Becker, Michael (2007): *Geschäftsprozess-Controlling auf der Basis von Business-Intelligence-Konzepten und Data-Warehouse-Systemen – Architektur, Datenmodell, Vorgehensmodell und Anwendungsszenarien*. Essen, Universität Duisburg-Essen, Dissertation. Shaker, Aachen.
- [Becker 2010] Becker, Jörg (2010): *Prozess der gestaltungsorientierten Wirtschaftsinformatik*. In: Österle, Hubert; Winter, Robert; Brenner, Walter (Hrsg.): *Gestaltungsorientierte Wirtschaftsinformatik*. Infowerk, Nürnberg, S. 13-17.
- [Becker & Chamoni 2008] Becker, Michael; Chamoni, Peter (2008): *Ein modellbasierter, integrierter Ansatz zur Gestaltung und Nutzung eines Process Warehouse*. In: Bichler, Martin; Hess, Thomas; Krcmar, Helmut; Lechner, Ulrike; Matthes, Florian; Picot, Arnold; Speitkamp, Benjamin; Wolf, Petra (Hrsg.): *Multikonferenz Wirtschaftsinformatik (MKWI)*, 2008, München, Tagungsband. GITO-Verlag, Berlin, S. 155-166.
- [Becker & Kahn 2012] Becker, Jörg; Kahn, Dieter (2012): *Der Prozess im Fokus*. In: Becker, Jörg; Kugeler, Martin; Rosemann, Michael (Hrsg.): *Prozessmanagement*. Springer Gabler, Berlin, S. 3-16.
- [Becker & Schütte 2004] Becker, Jörg; Schütte, Reinhard (2004): *Handelsinformationssysteme – Domänenorientierte Einführung in die Wirtschaftsinformatik*. 1. Aufl., mi-Wirtschaftsbuch, München.
- [Becker et al. 2011] Becker, Wolfgang; Lutz, Stefan; Back, Christian (2011): *Gabler Kompaktlexikon Modernes Rechnungswesen*. 3., überarb. Aufl., Springer Gabler, Wiesbaden.
- [Benker 2010] Benker, Thomas (2010): *Modellgetriebene Entwicklung von multidimensionalen Datenschemata auf Basis des Semantischen Data-Warehouse-Modells*. Bamberg, Otto-Friedrich-Universität, Diplomarbeit.
- [Benker 2013] Benker, Thomas (2013): *A Hybrid OLAP & OLTP Architecture Using Non-Relational Data Components*. In: Jung, Reinhard; Reichert, Manfred (Hrsg.): *Fifth International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA)*, 2013, St. Gallen, Proceedings. Köllen, Bonn, S. 41-57.
- [Benker 2015] Benker, Thomas (2015): *Konzeption einer Komponentenarchitektur für prozessorientierte OLTP- & OLAP-Anwendungssysteme*. In: *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik*, Bd. 97. Bamberg, Otto-Friedrich-Universität Bamberg.
- [Benker 2016] Benker, Thomas (2016): *A Generic Process Data Warehouse Schema for BPMN Workflows*. In: Abramowicz, Witold; Alt, Rainer; Franczyk, Bogdan (Hrsg.): *19th International Conference on Business Information Systems (BIS)*, 2016, Leipzig, Proceedings. Springer, Berlin, S. 222-234.

- [Benker & Jürck 2012] Benker, Thomas; Jürck, Carsten (2012): *A Case Study on Model-Driven Data Warehouse Development*. In: Cuzzocrea, Alfredo; Dayal, Umeshwar (Hrsg.): *14th International Conference on Data Warehousing and Knowledge Discovery (DAWAK)*, 2012, Vienna, Proceedings. Springer, Berlin, S. 54-64.
- [Benker & Jürck 2016] Benker, Thomas; Jürck, Carsten (2016): *Das betriebliche Informationssystem*. In: Benker, Thomas; Jürck, Carsten; Wolf, Matthias (Hrsg.): *Geschäftsprozessorientierte Systementwicklung*. Springer Vieweg, Wiesbaden, S. 25-32.
- [Birkmeier & Overhage 2009] Birkmeier, Dominik; Overhage, Sven (2009): *On Component Identification Approaches – Classification, State of the Art, and Comparison*. In: Lewis, Grace A.; Poernomo, Iman; Hofmeister, Christine (Hrsg.): *12th International Symposium on Component-Based Software Engineering (CBSE)*, 2009, East Stroudsburg, Proceedings. Springer, Berlin, S. 1-18.
- [Birkmeier & Overhage 2012a] Birkmeier, Dominik; Overhage, Sven (2012): *A Semi-Automated Approach to Support the Architect during the Generation of Component-Based Enterprise Architectures*. In: *20th European Conference on Information Systems (ECIS)*, 2012, Barcelona, Proceedings. S. 218-229, URL <http://aisel.aisnet.org/ecis2012/218> – Zugriffsdatum: 29.7.2017.
- [Birkmeier & Overhage 2012b] Birkmeier, Dominik; Overhage, Sven (2012): *A method to support a reflective derivation of business components from conceptual models*. In: *Information Systems and e-Business Management*, 11(2012), Nr. 3, S. 403-435
- [Boehnlein & Ulbrich-vom Ende 1999] Boehnlein, Michael; Ulbrich-vom Ende, Achim (1999): *Deriving Initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems*. In: *2nd International Workshop on Data Warehousing and OLAP (DOLAP)*, 1999, New York, Proceedings. ACM, New York, S. 15-21.
- [Böhnlein 2001] Böhnlein, Michael (2001): *Konstruktion semantischer Data-Warehouse-Schemata*. Bamberg, Otto-Friedrich-Universität Bamberg, Dissertation. Dt. Univ.-Verl., Wiesbaden.
- [Böhnlein & Ulbrich-vom Ende 2000] Böhnlein, Michael; Ulbrich-vom Ende, Achim (2000): *Grundlagen des Data Warehousing – Modellierung und Architektur*. In: *Bamberger Beiträge zur Wirtschaftsinformatik*, Bd. 55. Bamberg, Otto-Friedrich-Universität Bamberg.
- [Bork & Sinz 2013] Bork, Domenik; Sinz, Elmar J. (2013): *Bridging the Gap from a Multi-View Modelling Method to the Design of a Multi-View Modelling Tool*. In: *Enterprise Modelling and Information Systems Architectures*, 8(2013), Nr. 2, S. 25-41.
- [Bosch 2000] Bosch, Jan (2000): *Design and use of software architectures: Adopting and evolving a product-line approach*. Addison-Wesley, Harlow.
- [vom Brocke 2003] vom Brocke, Jan (2003): *Referenzmodellierung – Gestaltung und Verteilung von Konstruktionsprozessen*. Logos-Verlag, Berlin.
- [vom Brocke & Buddendick 2004] vom Brocke, Jan; Buddendick, Christian (2004): *Konstruktionstechniken für die Referenzmodellierung – Systematisierung, Sprachgestaltung und Werkzeugunterstützung*. In: Becker, Jörg; Delfmann, Patrick (Hrsg.): *Referenzmodellierung – Grundlagen, Techniken und domänenbezogene Anwendung*. Physica-Verlag HD, Heidelberg, S. 19-49.

- [Bucher & Dinter 2008a] Bucher, Tobias; Dinter, Barbara (2008): *Anwendungsfälle der Nutzung analytischer Informationen im operativen Kontext*. In: Bichler, Martin; Hess, Thomas; Krcmar, Helmut; Lechner, Ulrike; Matthes, Florian; Picot, Arnold; Speitkamp, Benjamin; Wolf, Petra (Hrsg.): *Multikonferenz Wirtschaftsinformatik (MKWI)*, 2008, München, Tagungsband. GITO-Verlag, Berlin, S. 167-179.
- [Bucher & Dinter 2008b] Bucher, Tobias; Dinter, Barbara (2008): *Process Orientation of Information Logistics - An Empirical Analysis to Assess Benefits, Design Factors, and Realization Approaches*. In: *41st Annual Hawaii International Conference on System Sciences (HICSS)*, 2008, Hawaii, Proceedings. S. 392-401.
- [Bucher et al. 2009] Bucher, Tobias; Gericke, Anke; Sigg, Stefan (2009): *Process-centric business intelligence*. In: *Business Process Management Journal*, 15(2009), Nr. 3, S. 408-429.
- [Bundesverband Paket & Express Logistik 2015] Bundesverband Paket & Express Logistik (2015): *KEP-Studie 2015: Analyse des Marktes in Deutschland*. URL http://www.biek.de/tl_files/biek/pressemitteilungen/2015/BIEK_KEP-Studie_2015.pdf – Zugriffsdatum: 29.07.2017.
- [Cai et al. 2011] Cai, Zhen-gong; Yang, Xiao-hu; Wang, Xin-yu; Kavs, Aleksander J. (2011): *A fuzzy formal concept analysis based approach for business component identification*. In: *Journal of Zhejiang University SCIENCE C*, 12(2011), Nr. 9, S. 707-720.
- [Casati et al. 2007] Casati, Fabio; Castellanos, Malú; Dayal, Umeshwar; Salazar, Norman (2007): *A Generic solution for Warehousing Business Process Data*. In: Koch, Christoph; Gehrke, Johannes; Garofalakis, Minos N.; Srivastava, Divesh; Aberer, Karl; Deshpande, Anand; Florescu, Daniela; Chan, Chee Yon.; Ganti, Venkatesh; Kanne, Carl-Christian; Klas, Wolfgang; Neuhold, Erich J. (Hrsg.): *33rd International Conference on Very Large Data Bases (VLDB)*, 2007, Vienna, Proceedings. ACM, S. 1128-1137.
- [Castellanos et al. 2009] Castellanos, Malu; Simitsis, Alkis; Wilkinson, Kevin; Dayal, Umeshwar (2009): *Automating the Loading of Business Process Data Warehouses*. In: *12th International Conference on Extending Database Technology (EDBT)*, 2009, Saint Petersburg, Proceedings. ACM, S. 612–623.
- [Chamoni & Gluchowski 2010] Chamoni, Peter; Gluchowski, Peter (2010): *Entwicklungslinien und Architekturkonzepte des On-Line-Analytical Processing*. In: Chamoni, Peter (Hrsg.): *Analytische Informationssysteme – Business Intelligence-Technologien und -Anwendungen*. 4., vollst. überarb. Aufl., Springer, Berlin, S. 197-228.
- [Chamoni & Gluchowski 2016] Chamoni, Peter; Gluchowski, Peter (2016): *Analytische Informationssysteme: Einordnung und Überblick*. In: Gluchowski, Peter; Chamoni, Peter (Hrsg.): *Analytische Informationssysteme – Business Intelligence-Technologien und -Anwendungen*. 5., vollst. überarb. Aufl., Springer, Berlin, S. 3-12.
- [Chan et al. 2011] Chan, Lee-Kwun; Sim, Yee-Wai; Yeoh, William (2011): *A SOA-Driven Business Intelligence Architecture*. In: *Communications of the IBIMA*, 2011, S. 1-8.
- [Cheesman & Daniels 2001] Cheesman, John; Daniels, John (2001): *UML components – A Simple Process for Specifying Component-based Software*. Addison-Wesley, Boston.

- [Christ 2009] Christ, Ulrich (2009): *An Architecture for Integrated Operational Business Intelligence*. In: Freytag, Johann Chr.; Ruf, Thomas; Lehner, Wolfgang; Vossen, Gottfried (Hrsg.): *13. Fachtagung Datenbanksysteme in Business, Technologie und Web (BTW)*, 2009, Münster, Tagungsband. S. 460-468.
- [Christmann 1996] Christmann, Alfred (1996): *Data-Warehouse-Lösung der Stadt Köln*. In: Lamersdorf, Winfried (Hrsg.): *Online-Congress Band VIII: Data Warehousing, OLAP, Führungsinformationssysteme – Neue Entwicklungen des Informationsmanagements*. Online GmbH, Velbert, S. C822.01-C822.12.
- [Codd et al. 1993] Codd, E. F.; Codd, S. B.; Salley, C. T. (1993): *Providing OLAP to User-Analysts: An IT Mandate*. URL http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf – Zugriffsdatum: 29.07.2017.
- [Codd 1970] Codd, E. F.: *A Relational Model of Data for Large Shared Data Banks*. In: *Commun. ACM*, 13(1970), Nr. 6, S. 377-387.
- [Curtis et al. 1992] Curtis, Bill; Kellner, Marc; Over, Jim (1992): *Process modeling*. In: *Commun. ACM*, 35(1992), Nr. 9, S. 75-90.
- [Dadam 1996] Dadam, Peter (1996): *Verteilte Datenbanken und Client/Server-Systeme – Grundlagen, Konzepte und Realisierungsformen*. Springer, Berlin.
- [Davenport 1993] Davenport, Thomas H. (1993): *Process innovation – Reengineering Work Through Information Technology*. Harvard Business School Press, Boston.
- [Dayal et al. 2009] Dayal, Umeshwar; Wilkinson, Kevin; Simitsis, Alkis; Castellanos, Malú (2009): *Business Processes Meet Operational Business Intelligence*. In: *IEEE Data Engineering Bulletin*, 32(2009), Nr. 3, S. 35-41.
- [Dinter 2008] Dinter, Barbara (2008): *Einsatzmöglichkeiten serviceorientierter Architekturen in der Informationslogistik*. In: Töpfer, Jochen; Winter, Robert (Hrsg.): *Active Enterprise Intelligence*. Springer, Berlin, S. 221-241.
- [Dori 2002] Dori, Dov (2002): *Object-Process Methodology – A Holistic Systems Paradigm*. Springer, Berlin.
- [Dori et al. 2008] Dori, Dov; Feldman, Roman; Sturm, Arnon (2008): *From Conceptual Models to Schemata – An Object-Process-based Data Warehouse Construction Method*. In: *Information Systems*, 33(2008), Nr. 6, S. 567-593.
- [Dorn et al. 2010] Dorn, Christoph; Burkhart, Thomas; Werth, Dirk; Dustdar, Schahram (2010): *Self-adjusting Recommendations for People-Driven Ad-Hoc Processes*. In: Hull, Richard; Mendling, Jan; Tai, Stefan (Hrsg.): *8th International Conference on Business Process Management (BPM)*, 2010, New Jersey, Proceedings. Springer, Berlin, S. 327-342.
- [Dumas et al. 2013] Dumas, Marlon; La Rosa, Marcello; Mendling, Jan; Reijers, Hajo A. (2013): *Fundamentals of Business Process Management*. Springer, Berlin.
- [Earl 1994] Earl, Michael J. (1994): *The new and the old of Business Process Redesign*. In: *The Journal of Strategic Information Systems*, 3(1994), Nr. 1, S. 5-22.

- [Eckerson 2007] Eckerson, Wayne (2007): *Best Practices in Operational BI – Converging Analytical and Operational Processes*. In: TDWI Best Practices Report.
- [Eder et al. 2002] Eder, Johann; Olivotto, Georg E.; Gruber, Wolfgang (2002): *A Data Warehouse for Workflow Logs*. In: Han, Yanbo; Tai, Stefan; Wikarski, Dietmar (Hrsg.): *International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS)*, 2002, Beijing, Proceedings. Springer, Berlin, S. 1-15.
- [Elegant] BI 2008] Elegant] BI (2008): *Operational Business Intelligence (BI) – Integrated Business Intelligence for Performance Management, Operational Business Intelligence and Data Management*. URL <https://www.elegantjbi.com/images/whitepaper/EJBIR1002WP-Operational-Business-Intelligence.pdf> – Zugriffsdatum: 29.07.2017
- [Esswein 2016] Esswein, Werner (2016): *Referenzmodelle und -modellierung*. In: Benker, Thomas; Jürck, Carsten; Wolf, Matthias (Hrsg.): *Geschäftsprozessorientierte Systementwicklung*. Springer Vieweg, Wiesbaden, 2016, S. 51-62.
- [Evans 2004] Evans, Eric (2004): *Domain-driven Design: Tackling Complexity in the Heart of Software*. 2. Aufl., Addison-Wesley, Boston.
- [Färber et al. 2012] Färber, Franz; Cha, Sang Kyu.; Primsch, Jürgen; Bornhövd, Christof; Sigg, Stefan; Lehner, Wolfgang (2012): *SAP HANA Database: Data Management for Modern Business Applications*. In: SIGMOD Rec, 40(2012), Nr. 4, S. 45-51.
- [Felden et al. 2010] Felden, Carsten; Chamoni, Peter; Linden, Markus (2010): *From Process Execution towards a Business Process Intelligence*. In: Abramowicz, Witold; Tolksdorf, Robert (Hrsg.): *13th International Conference on Business Information Systems (BIS)*, 2010, Berlin, Proceedings. Springer, Berlin, S. 195-206.
- [Ferstl 1992] Ferstl, Otto K. (1992): *Integrationskonzepte betrieblicher Anwendungssysteme*. In: *Fachbericht Informatik*, Bd. 1/92. Universität Koblenz-Landau.
- [Ferstl & Sinz 1990] Ferstl, Otto K.; Sinz, Elmar J. (1990): *Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)*. In: *Wirtschaftsinformatik*, 32(1990), Nr. 6, S. 566-581.
- [Ferstl & Sinz 1991] Ferstl, Otto K.; Sinz, Elmar J. (1991): *Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)*. In: *Wirtschaftsinformatik*, 33(1991), Nr. 6, S. 477-491.
- [Ferstl & Sinz 1993] Ferstl, Otto K.; Sinz, Elmar J. (1993): *Der Modellierungsansatz des Semantischen Objektmodells (SOM)*. In: *Bamberger Beiträge zur Wirtschaftsinformatik*, Bd. 18. Bamberg, Otto-Friedrich-Universität Bamberg.
- [Ferstl & Sinz 1995] Ferstl, Otto K.; Sinz, Elmar J. (1995): *Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen*. In: *Wirtschaftsinformatik*, 37(1995), Nr. 3, S. 209-220.
- [Ferstl & Sinz 2006] Ferstl, Otto K.; Sinz, Elmar J. (2006): *Modeling of Business Systems Using SOM*. In: Schmidt, Günter; Mertins, Kai; Bernus, Péter (Hrsg.): *Handbook on Architectures of Information Systems*. Springer, Berlin, S. 347-367.

- [Ferstl & Sinz 2013] Ferstl, Otto K.; Sinz, Elmar J. (2013): *Grundlagen der Wirtschaftsinformatik*. 7., akt. Aufl., de Gruyter Oldenbourg, Berlin.
- [Ferstl et al. 1997] Ferstl, Otto K.; Sinz, Elmar J.; Hammel, Christoph; Schlitt, Michael; Wolf, Stefan: Bausteine für komponentenbasierte Anwendungssysteme. In: *HMD – Theorie und Praxis der Wirtschaftsinformatik, Schwerpunktheft Componentware 197 (1997)*, S. 24–46
- [Fischermanns 2013] Fischermanns, Guido (2013): *Praxishandbuch Prozeßmanagement*. 11., bearb. Aufl., Schmidt, Gießen.
- [Frank 2006] Frank, Ulrich (2006): *Towards a Pluralistic Conception of Research Methods in Information Systems Research*. In: Adelsberger, Heimo; Chamoni, Peter; Dorloff, Frank; Echte, Klaus; Eicker, Stefan; Frank, Ulrich; Goedicke, Michael; Kollmann, Tobias; Müller-Clostermann, Bruno; Pohl, Klaus; Rathgeb, Erwin P.; Unland, Rainer; Zelewski, Stephan (Hrsg.): *ICB-Research*, Bd. 7. URL http://www.icb.uni-due.de/fileadmin/ICB/research/research_reports/ICBReport07.pdf – Zugriffsdatum: 29.07.2017.
- [Friedrich et al. 2016] Friedrich, Thomas; Robel, Martin; Schlauderer, Sebastian: *Jenseits objektorientierter Entwicklung: Systematische Identifikation von Fachkomponenten mit SOM*. In: Benker, Thomas; Jürck, Carsten; Wolf, Matthias (Hrsg.): *Geschäftsprozessorientierte Systementwicklung*. Springer Vieweg, Wiesbaden, S. 241-260.
- [Gabler 2014] Gabler (2014): *Gabler Wirtschaftslexikon*, Bd. 4. 18., akt. und erw. Aufl., Springer Gabler, Wiesbaden.
- [Gabriel 2014] Gabriel, Roland (2014): *Informationssystem*. In: Kurbel, Karl; Becker, Jörg; Gronau, Norbert; Sinz, Elmar J.; Suhl, Leena (Hrsg.): *Enzyklopädie der Wirtschaftsinformatik – Online-Lexikon*. Oldenbourg, München. URL <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/uebergreifendes/Kontext-und-Grundlagen/Informationssystem> – Zugriffsdatum: 29.07.2017.
- [Gaitanides 1983] Gaitanides, Michael (1983): *Prozeßorganisation: Entwicklung, Ansätze und Programme prozeßorientierter Organisationsgestaltung*. Vahlen, München.
- [Ganesan & Sengupta 2001] Ganesan, Rajeshwari; Sengupta, Shubhashis (2001): *O2BC: a Technique for the Design of Component-based Applications*. In: *39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS)*, 2001, Santa Barbara, Proceedings. S. 46-55.
- [Gluchowski et al. 2008] Gluchowski, Peter; Dittmar, Carsten; Gabriel, Roland (2008): *Management Support Systeme und Business Intelligence – Computergestützte Informationssysteme für Fach- und Führungskräfte*. 2., vollst. überarb. Aufl., Springer, Berlin.
- [Gluchowski et al. 2009] Gluchowski, Peter; Kemper, Hans-Georg; Seufert, Andreas (2009): *Innovative Prozess-Steuerung: Was ist neu an Operational BI?* In: *BI Spektrum*, 4(2009), Nr. 1, S. 8-12.
- [Golfarelli et al. 1998] Golfarelli, Matteo; Maio, Dario; Rizzi, Stefano (1998): *Conceptual Design of Data Warehouses from E/R Schemes*. In: *31st Annual Hawaii International Conference on System Sciences (HICSS)*, 1998, Hawaii, Proceedings. S. 334-343.

- [Golfarelli & Rizzi 2009] Golfarelli, Matteo; Rizzi, Stefano (2009): *Data Warehouse Design – Modern Principles and Methodologies*. McGraw-Hill, New York.
- [Gordon et al. 2006] Gordon, S.; Grigg, R.; Horne, M.; Turman, S. (2006): *Service-Oriented Business Intelligence*. In: *The Architecture Journal*, 6(2006), Nr. 1, S. 23-32.
- [Grigoria et al. 2004] Grigoria, Daniela; Casati, Fabio; Castellanos, Malu; Dayal, Umeshwar; Sayal, Mehmet; Shan, Ming-Chien (2004): *Business Process Intelligence*. In: *Computers in Industry*, 53(2004), Nr. 3, S. 321-343.
- [Gutenberg 1983] Gutenberg, Erich (1983): *Grundlagen der Betriebswirtschaftslehre – Die Produktion*, Bd. 1. 24. Aufl., Springer, Berlin.
- [Hackathorn 2004] Hackathorn, Richard (2004): *The BI Watch: Real-Time to Real-Value*. URL <http://www.information-management.com/issues/20040101/7913-1.html?zkPrintable=1&nopagination=1> – Zugriffsdatum: 29.07.2017
- [Hahne 2012] Hahne, Michael (2012): *Historisierung im Data Warehouse* (Teil IV zur Serie Datenmodellierung). In: *BI Spektrum*, 7(2012), Nr. 3, S. 35-38.
- [Hahne 2014] Hahne, Michael (2014): *Modellierung von Business-Intelligence-Systemen – Leitfaden für erfolgreiche Projekte auf Basis flexibler Data-Warehouse-Architekturen*. dpunkt.verlag, Heidelberg.
- [Herzum & Sims 2000] Herzum, Peter; Sims, Oliver (2000): *Business Component Factory – A Comprehensive Overview of Component-based Development for the Enterprise*. John Wiley, New York.
- [Hopkins 2000] Hopkins, Jon (2000): *Component Primer*. In: *Commun. ACM*, 43(2000), Nr. 10, S. 27–30.
- [Hrach & Alt 2012] Hrach, Christian; Alt, Rainer (2012): *Operational Business Intelligence bei Call Centern – Erkenntnisse einer Fallstudienuntersuchung*. In: Mattfeld, Dirk C. (Hrsg.): *Multikonferenz Wirtschaftsinformatik (MKWI)*, 2012, Braunschweig, Tagungsband. Gito, Berlin, S. 1131-1144.
- [Huth et al. 2001] Huth, C.; Erdmann, I.; Nastansky, L. (2001): *GroupProcess: Using Process Knowledge from the Participative Design and Practical Operation of ad hoc Processes for the Design of Structured Workflows*. In: Sprague, Ralph H. (Hrsg.): *34th Annual Hawaii International Conference on System Sciences (HICCS)*, 2001, Hawaii, Proceedings. S. 10-19.
- [Inmon 1999] Inmon, William H. (1999): *Building the Operational Data Store*. 2. Aufl., Wiley, New York.
- [Inmon 2000] Inmon, William H. (2000): *Operational and Informational Reporting*. In: *Information Management Magazine*, URL <http://www.information-management.com/issues/20000701/2349-1.html> – Zugriffsdatum: 29.07.2017.
- [Inmon 2005] Inmon, William H. (2005): *Building the Data Warehouse*. 4. Aufl., Wiley, Indianapolis.
- [Inmon & Linstedt 2015] Inmon, William H.; Linstedt, Dan (2015): *Data Architecture: A Primer for the Data Scientist – Big Data, Data Warehouse and Data Vault*. Morgan Kaufmann, Burlington.

- [Jablonski & Bussler 1996] Jablonski, Stefan; Bussler, Christoph (1996): *Workflow Management – Modeling, Concepts, Architecture and Implementation*. International Thomson Computer Press, London, Boston.
- [Jain et al. 2001] Jain, Hemant; Chalimeda, Naresh; Ivaturi, Navin; Reddy, Balarama (2001): *Business Component Identification - A Formal Approach*. In: *5th IEEE International Conference on Enterprise Distributed Object Computing (EDOC)*, 2001, Seattle, Proceedings. S. 183-188.
- [Jang et al. 2003] Jang, Yoon-Jung; Kim, Eun-Young; Lee, Kyung-Whan (2003): *Object-Oriented Component Identification Method using the Affinity Analysis Technique*. In: Goos, Gerhard; Hartmanis, Juris; van Leeuwen, Jan; Konstantas, Dimitri; Léonard, Michel; Pigneur, Yves; Patel, Shusma (Hrsg.): *9th International Conference on Object-Oriented Information Systems (OOIS)*, 2003, Geneva, Proceedings. Springer, Berlin, S. 317–321.
- [Jarke et al. 2003] Jarke, Matthias; Lenzerini, Maurizio; Vassiliou, Yannis; Vassiliadis, Panos (2003): *Fundamentals of Data Warehouses*. 2. Aufl., Springer, Berlin.
- [Kemper & Eickler 2013] Kemper, Alfons; Eickler, André (2013): *Datenbanksysteme – Eine Einführung*. 9., erw. und akt. Aufl., Oldenbourg, München.
- [Kemper & Neumann 2011] Kemper, A.; Neumann, T. (2011): *HyPer: A Hybrid OLTP & OLAP Main Memory Database System based on Virtual Memory Snapshots*. In: *27th International Conference on Data Engineering (ICDE)*, 2011, Hannover, Proceedings. S. 195–206.
- [Kemper et al. 2010] Kemper, Hans-Georg; Baars, Henning; Mehanna, Walid (2010): *Business Intelligence - Grundlagen und praktische Anwendungen: Eine Einführung in die IT-basierte Managementunterstützung*. 3., überarb. und erw. Aufl., Vieweg Teubner Verlag, Wiesbaden.
- [Kim & Chang 2004] Kim, Soo Don; Chang, Soo Ho (2004): *A Systematic Method to Identify Software Components*. In: *11th Asia-Pacific Software Engineering Conference (APSEC)*, 2004, Busan, Proceedings. S. 538-545.
- [Kimball 2009] Kimball, Ralph (2009): *The Data Warehouse Lifecycle Toolkit – Practical Techniques for Building Data Warehouse and Business Intelligence Systems*. 2. Aufl., Wiley, Indianapolis.
- [Kimball & Ross 2002] Kimball, Ralph; Ross, Margy: *The Data Warehouse Toolkit – The Complete Guide to Dimensional Modeling*. 2. Aufl., Wiley, New York.
- [Korherr 2007] Korherr, Birgit (2007): *Business Process Modelling - Languages, Goals and Variabilities*. Wien, Technische Universität Wien, Dissertation. URL http://publik.tuwien.ac.at/files/PubDat_141379.pdf – Zugriffsdatum: 29.07.2017.
- [Kosiol 1976] Kosiol, Erich (1976): *Organisation der Unternehmung*. 2., durchges. Aufl., Gabler, Wiesbaden.
- [Krücke & Sinz 2011] Krücke, Andree; Sinz, Elmar J. (2011): *Entwurf partieller SOA auf der Grundlage von Geschäftsprozessmodellen*. In: Sinz, Elmar J.; Bartmann, Dieter; Boddendorf, Freimut; Ferstl, Otto K. (Hrsg.): *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. Schriften aus der Fakultät für Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg, Bd. 9, Univ. of Bamberg Press, Bamberg, S. 287-312.

- [Kueng et al. 2001] Kueng, Peter; Wettstein, Thomas; List, Beate (2001): *A Holistic Process Performance Analysis through a Performance Data Warehouse*. In: *American Conference on Information Systems (AMCIS)*, 2001, Boston, Proceedings. S. 349–356.
- [Küpper et al. 2013] Küpper, Hans-Ulrich; Friedl, Gunther; Hofmann, Christian; Hofmann, Yvette; Pedell, Burkhard (2013): *Controlling – Konzeption, Aufgaben, Instrumente*. 6., überarb. Aufl., Schäffer-Poeschel, Stuttgart.
- [Lee et al. 1999] Lee, Sang Duck; Yang, Young Jong; Cho, Eun Sook; Kim, Soo Dong; Rhew, Sung Yul (1999): *COMO: A UML-Based Component Development Methodology*. In: *Sixth Asia Pacific Software Engineering Conference (ASPEC)*, 1999, Takamatsu, Proceedings. S. 54-61.
- [Lehner 2003] Lehner, Wolfgang (2003): *Datenbanktechnologie für Data-Warehouse-Systeme – Konzepte und Methoden*. 1. Aufl., dpunkt.verlag, Heidelberg.
- [Lenz & Shoshani 1997] Lenz, Hans-Joachim; Shoshani, Arie (1997): *Summarizability in OLAP and Statistical Data Bases*. In: *Ninth International Conference on Scientific and Statistical Database Management (SSDBM)*, 1997, Washington, Proceedings. S. 132-143.
- [Leunig et al. 2011] Leunig, Benjamin; Wagner, Daniel; Ferstl, Otto K. (2011): *E-Car-Szenario - Hochflexible Geschäftsprozesse in der Logistik als Integrationsszenario für den Forschungsverbund forFLEX*. In: Sinz, Elmar J.; Bartmann, Dieter; Bodendorf, Freimut; Ferstl, Otto K. (Hrsg.): *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. Schriften aus der Fakultät für Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg, Bd. 9, Univ. of Bamberg Press, Bamberg, S. 15-38.
- [Levi & Arsanjani 2002] Levi, Keith; Arsanjani, Ali (2002): *A goal-driven approach to enterprise component identification and specification*. In: *Commun. ACM*, 45(2002), Nr. 10, S. 45-52.
- [Linden et al. 2010] Linden, Markus; Neuhaus, Sebastian; Kilimann, Daniel; Bley, Tanja; Chamoni, Peter (2010): *Event-Driven Business Intelligence Architecture for Real-Time Process Execution in Supply Chains*. In: Abramowicz, Witold; Tolksdorf, Robert (Hrsg.): *13th International Conference on Business Information Systems (BIS)*, 2010, Berlin, Proceedings. Springer, Berlin, S. 280-290.
- [Linden et al. 2011] Linden, Markus; Felden, Carsten; Chamoni, Peter (2011): *Dimensions of Business Process Intelligence*. In: zur Muehlen, Michael; Su, Jianwen (Hrsg.): *Business Process Management Workshops (BPM 2010)*. Springer, Berlin, S. 208-213.
- [Linstedt & Graziano 2010] Linstedt, Dan; Graziano, Kent (2010): *Super Charge your Data Warehouse – Invaluable Data Modeling Rules to Implement your Data Vault*. CreateSpace, USA.
- [Linstedt & Olschimke 2016] Linstedt, Daniel; Olschimke, Michael (2016): *Building a Scalable Data Warehouse with Data Vault 2.0*. Morgan Kaufmann, Amsterdam.
- [List & Machaczek 2004] List, Beate; Machaczek, Karl (2004): *Towards a Corporate Performance Measurement System*. In: *ACM Symposium on Applied Computing (SAC)*, 2004, New York, Proceedings. S. 1344-1350.

- [List et al. 2000] List, Beate; Schiefer, Josef; Tjoa, A Min; Quirchmayr, Gerald (2000): *Multidimensional Business Process Analysis with the Process Warehouse*. In: Abramowicz, Witold; Zurada, Jozef (Hrsg.): *Knowledge Discovery for Business Information Systems*. Springer, Boston, S. 211-227.
- [Mansmann et al. 2007a] Mansmann, Svetlana; Neumuth, Thomas; Scholl, Marc H. (2007): *Multidimensional Data Modeling for Business Process Analysis*. In: Parent, Christine; Schewe, Klaus-Dieter; Storey, Veda; Thalheim, Bernhard (Hrsg.): *26th International Conference on Conceptual Modeling (ER 2007)*, 2007, Auckland, New Zealand, Proceedings. Springer, Berlin, S. 23-38.
- [Mansmann et al. 2007b] Mansmann, Svetlana; Neumuth, Thomas; Scholl, Marc H. (2007): *OLAP Technology for Business Process Intelligence: Challenges and Solutions*. In: Song, Il Yeal; Eder, Johann; Nguyen, ThoManh (Hrsg.): *9th Internatioal Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, 2007, Regensburg, Proceedings. Springer, Berlin, S. 111-122.
- [Marjanovic 2007] Marjanovic, Olivera (2007): *The Next Stage of Operational Business Intelligence: Creating New Challenges for Business Process Management*. In: *40th Annual Hawaii International Conference on System Sciences (HICSS)*, 2007, Hawaii, Proceedings.
- [Markarian et al. 2007] Markarian, James; Brobst, Stephen; Bedell, Jeff (2007): *Critical Success Factors Deploying Pervasive BI: White Paper*. URL <https://www.microstrategy.com/Strategy/media/downloads/solutions/Pervasive-BI.pdf> – Zugriffsdatum: 29.07.2017.
- [Meng et al. 2005] Meng, Fan-Chao; Zhan, Den-Chen; Xu, Xiao-Fei (2005): *Business Component Identification of Enterprise Information System: A Hierarchical Clustering Method*. In: *IEEE International Conference on e-Business Engineering (ICEBE)*, 2005, Beijing, Proceedings. S. 473-480.
- [Mertens 2005] Mertens, Peter (2005): *Gefahren für die Wirtschaftsinformatik - Risikoanalyse eines Faches*. In: Ferstl, Otto K.; Sinz, Elmar J.; Eckert, Sven; Isselhorst, Tilman (Hrsg.): *Internationale Tagung Wirtschaftsinformatik (WI)*, 2005, Bamberg, Tagungsband. Physica-Verlag, Heidelberg, S. 1733-1754.
- [Mertens 2009] Mertens, Peter (2009): *Integrierte Informationsverarbeitung 1 – Operative Systeme in der Industrie*. 17. Aufl., Gabler, Wiesbaden.
- [Mertens & Meier 2009] Mertens, Peter; Meier, Marco C. (2009): *Integrierte Informationsverarbeitung 2 – Planungs- und Kontrollsysteme in der Industrie*. 10. Aufl., Gabler, Wiesbaden, 2009.
- [zur Muehlen 2001] zur Muehlen, Michael (2001): *Process-driven management information systems combining data warehouses and workflow technology*. In: *4th International Conference on Electronic Commerce Research (ICECR)*, 2001, Dallas, Proceedings. S. 550-566.
- [zur Mühlen 2004] zur Mühlen, Michael (2004): *Workflow-based Process Controlling – Foundation, Design, and Application of Workflow-driven Process Information Systems* Münster, Universität Münster, Dissertation, 2002. Logos-Verlag, Berlin.

- [zur Muehlen & Shapiro 2010] zur Muehlen, Michael; Shapiro, Robert (2010): *Business Process Analytics*. In: Brocke, Jan; Rosemann, Michael (Hrsg.): *Handbook on Business Process Management 2*. Springer, Berlin, S. 137-157.
- [Nguyen et al. 2005] Nguyen, Tho Man.; Schiefer, Josef; Tjoa, A. Min (2005): *Sense & response service architecture (SARESA): An Approach towards Real-time Business Intelligence Solution and its use for a Fraud Detection Application*. In: Song, Il-Yeol; Trujillo, Juan (Hrsg.): *8th ACM International Workshop on Data Warehousing and OLAP (DOLAP)*, 2005, Bremen, Proceedings. S. 77-86.
- [Niedrite et al. 2007] Niedrite, Laila; Solodovnikova, Darja; Treimanis, Maris; Niedritis, Aivars (2007): *Goal-driven Design of a Data Warehouse-based Business Process Analysis System*. In: *6th International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED)*, 2007, Stevens Point, Wisconsin, Proceedings. S. 243-249.
- [Oberhofer & Nijkamp 2009] Oberhofer, Martin; Nijkamp, Erik (2009): *Embedded Analytics in Front Office Applications*. In: Freytag, Johann Chr.; Ruf, Thomas; Lehner, Wolfgang; Vossen, Gottfried (Hrsg.): *13. Fachtagung Datenbanksysteme in Business, Technologie und Web (BTW)*, 2009, Münster, Tagungsband. S. 449-459.
- [Oehler 2006] Oehler, Karsten (2006): *Corporate Performance Management mit Business-Intelligence-Werkzeugen*. Hanser, München.
- [OMG 2011] OMG (2011): *Business Process Model and Notation (BPMN): Version 2.0*. URL <http://www.omg.org/spec/BPMN/2.0/PDF> – Zugriffsdatum: 29.07.2017
- [OMG 2015] OMG (2015): *OMG Unified Modeling Language (UML) – Version 2.5*. URL <http://www.omg.org/spec/UML/2.5/PDF/> – Zugriffsdatum: 29.07.2017
- [Österle et al. 2010] Österle, Hubert; Becker, Jörg; Frank, Ulrich; Hess, Thomas; Karagiannis, Dimitris; Krcmar, Helmut; Loos, Peter; Mertens, Peter; Oberweis, Andreas; Sinz, Elmar J. (2010): *Memorandum zur gestaltungsorientierten Wirtschaftsinformatik*. In: Österle, Hubert; Winter, Robert; Brenner, Walter (Hrsg.): *Gestaltungsorientierte Wirtschaftsinformatik*. Infowerk, Nürnberg, S. 1-6.
- [Overhage et al. 2012] Overhage, Sven; Birkmeier, Dominik Q.; Schlauderer, Sebastian (2012): *Quality Marks, Metrics, and Measurement Procedures for Business Process Models*. In: *Business & Information Systems Engineering*, 4(2012), Nr. 5, S. 229-246.
- [Panian 2008] Panian, Zeljko (2008): *How to Make Business Intelligence Actionable through Service-oriented Architectures*. In: *WSEAS Transactions on Business and Economics*, 5(2008), Nr. 5, S. 210-221.
- [Parnas 1972] Parnas, David L. (1972): *On the Criteria to Be Used in Decomposing Systems into Modules*. In: *Commun. ACM*, 15(1972), Nr. 12, S. 1053-1058.
- [Pau et al. 2007] Pau, Kin-Chan; Si, Yain-Whar; Dumas, Marlon (2007): *Data Warehouse Model for Audit Trail Analysis in Workflows*. In: *IEEE International Conference on e-Business Engineering (ICEBE)*, 2007, Hong Kong, Proceedings.
- [Pendse & Creeth 1995] Pendse, Nigel; Creeth, Richard (1995): *The OLAP report: Succeeding with On-Line Analytical Processing*. 1. Aufl., Business Intelligence, 1995.

- [Porter 1989] Porter, Michael E. (1989): *Wettbewerbsvorteile – Spitzenleistungen erreichen und behaupten*. Campus, New York, 1989.
- [Pospiech & Felden 2013] Pospiech, Marco; Felden, Carsten (2013): *Service-Oriented Business Intelligence Reference Architecture in Face of Advanced BI Concepts – Research in Progress*. In: *Nineteenth Americas Conference on Information Systems (AMCIS)*, 2013, Chicago, Proceedings.
- [Pütz & Sinz 2010] Pütz, Corinna; Sinz, Elmar J. (2010): *Model-driven Derivation of BPMN Workflow Schemata from SOM Business Process Models*. In: *Enterprise Modeling and Information Systems Architectures*, 5(2010), Nr. 2, S. 57-72.
- [Reussner & Hasselbring 2009] Reussner, Ralf; Hasselbring, Wilhelm (2009): *Handbuch der Software-Architektur*. 2., überarb. und erw. Aufl., dpunkt.-Verl., Heidelberg.
- [Rizzi et al. 2006] Rizzi, Stefano; Abelló, Alberto; Lechtenböcker, Jens; Trujillo, Juan (2006): *Research in Data Warehouse Modeling and Design: Dead or Alive?*. In: *9th ACM International Workshop on Data Warehousing and OLAP (DOLAP)*, 2006, Arlington, Proceedings. S. 3-10.
- [Russom 2011] Russom, Philip (2011): *Operational Data Warehousing: The Integration of Operational Applications and Data Warehouses*. In: TDWI Best Practices Report. URL <http://tdwi.org/research/2010/10/bpr-q4-operational-data-warehousing.aspx> – Zugriffsdatum: 29.07.2017.
- [Sadalage & Fowler 2012] Sadalage, Pramod J.; Fowler, Martin (2012): *NoSQL Distilled – A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, Upper Saddle River.
- [Santos & Oliveira e Sá 2016] Santos, Maribel Yas.; Oliveira e Sá, Jorge (2016): *A Data Warehouse Model for Business Processes Data Analytics*. In: Gervasi, Osvaldo; Murgante, Beniamino; Misra, Sanjay; Rocha, Ana Maria A.C.; Torre, Carmelo M.; Taniar, David; Apduhan, Bernady O.; Stankova, Elena; Wang, Shangguang (Hrsg.): *16th International Conference Computational Science and Its Applications (ICCSA)*, 2016, Beijing, Proceedings. Springer, Berlin, S. 241-256.
- [Schaffner et al. 2009] Schaffner, Jan; Bog, Anja; Krüger, Jens; Zeier, Alexander (2009): *A Hybrid Row-Column OLTP Database Architecture for Operational Reporting*. In: Aalst, Will; Mylopoulos, John; Sadeh, Norman M.; Shaw, Michael J.; Szyperski, Clemens; Castellanos, Malu; Dayal, Umesh; Sellis, Timos (Hrsg.): *Second International Workshop on Business Intelligence for the Real-Time Enterprise (BIRTE)*, Auckland, New Zealand, Proceedings. Springer, Berlin, S. 61-74.
- [Scheer 2001] Scheer, August-Wilhelm (2001): *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. 4. Aufl., Springer, Berlin.
- [Scheer 2002] Scheer, August-Wilhelm (2002): *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. 4., durchges. Aufl., Springer, Berlin.
- [Schiefer et al. 2004] Schiefer, J.; Jeng, Jun-Jang; Kapoor, S.; Chowdhary, P. (2004): *Process Information Factory: A Data Management Approach for Enhancing Business Process Intelligence*. In: *IEEE International Conference on e-Commerce Technology (CEC)*, 2004, Lissabon, Proceedings. S. 162-169.

- [Schwingel 2010] Schwingel, Jörg (2010): *Prozessorientierte BI - Strategien für Process Performance Management*. In: *BI Spektrum*, 5(2010), Nr. 3, S. 14-18.
- [Seidlmeier 2015] Seidlmeier, Heinrich (2015): *Prozessmodellierung mit ARIS – Eine beispielorientierte Einführung für Studium und Praxis in ARIS 9*. 4., akt. Aufl., Springer Vieweg, Wiesbaden.
- [Shahzad & Johannesson 2009] Shahzad, Khurram; Johannesson, Paul (2009): *An Evaluation of Process Warehousing Approaches for Business Process Analysis*. In: *ACM International Workshop on Enterprises and Organizational Modeling and Simulation (EOMAS)*, 2009, Amsterdam, Proceedings. S. 1-14.
- [Sharp & Ryan 2005] Sharp, Jason; Ryan, Sherry (2005): *A Review of Component-Based Software Development*. In: *International Conference on Information Systems (ICIS)*, 2005, Las Vegas, Proceedings. S. 151-161.
- [Silver 2011] Silver, Bruce (2011): *BPMN Method and Style: A Structured Approach for Business Process Modeling and Implementation Using BPMN 2*. 2. Aufl., Cody-Cassidy Press, Aptos.
- [Sinz 1988] Sinz, Elmar J. (1988): *Das Strukturierte Entity-Relationship-Modell (SER-Modell)*. In: *Angewandte Informatik*, 30(1988), Nr. 5, S. 191-202.
- [Sinz 1997] Sinz, Elmar J. (1997): *Architektur betrieblicher Informationssysteme*. In: *Bamberger Beiträge zur Wirtschaftsinformatik*, Bd. 40. Bamberg, Otto-Friedrich-Universität Bamberg
- [Sinz 1998] Sinz, Elmar J. (1998): *Modellierung betrieblicher Informationssysteme: Gegenstand, Anforderungen und Lösungsansätze*. In: Pohl, Klaus; Schürr, Andy; Vossen, Gottfried (Hrsg.): *GI-Workshop Modellierung*, 1998, Münster, Proceedings. S. 27–28.
- [Sinz 1999] Sinz, Elmar J. (1999): *Konstruktion von Informationssystemen*. In: *Bamberger Beiträge zur Wirtschaftsinformatik*, Bd. 53. Bamberg, Otto-Friedrich-Universität Bamberg
- [Sinz 2002a] Sinz, Elmar J. (2002): *Architektur von Informationssystemen*. In: Rechenberg, Peter; Pomberger, Gustav (Hrsg.): *Informatik-Handbuch*. Hanser, München, S. 1055-1068.
- [Sinz 2002b] Sinz, Elmar J. (2002): *Data Warehouse*. In: Küpper, Hans-Ulrich (Hrsg.): *Handwörterbuch Unternehmensrechnung und Controlling*. 4. Aufl., Schäffer-Poeschel, Stuttgart S. 309-318.
- [Sinz 2010] Sinz, E. J. (2010): *Konstruktionsforschung in der Wirtschaftsinformatik: Was sind die Erkenntnisziele gestaltungsorientierter Wirtschaftsinformatikforschung?* In: Österle, Hubert; Winter, Robert; Brenner, Walter (Hrsg.): *Gestaltungsorientierte Wirtschaftsinformatik*. Infowerk, Nürnberg, S. 27-33.
- [Sinz & Ulbrich-vom Ende 2010] Sinz, Elmar J.; Ulbrich-vom Ende, Achim (2010): *Architektur von Data-Warehouse-Systemen*. In: Chamoni, Peter (Hrsg.): *Analytische Informationssysteme – Business Intelligence-Technologien und -Anwendungen*. 4., vollst. überarb. Aufl., Springer, Berlin, S. 175-196.

- [Sinz et al. 2001] Sinz, E. J.; Böhnlein, M.; Ulbrich-vom Ende, A.; Plaha, M. (2001): *Architekturkonzept eines verteilten Data Warehouse-Systems für das Hochschulwesen*. In: *Bamberger Beiträge zur Wirtschaftsinformatik*, Bd. 59. Bamberg, Otto-Friedrich-Universität Bamberg.
- [Sommerville 2016] Sommerville, Ian (2016): *Software engineering*. 10., globale Aufl., Pearson Education Limited, Boston.
- [Strahringer 2016] Strahringer, Susanne (2016): *Modelle und Metamodellierung*. In: Benker, Thomas; Jürck, Carsten; Wolf, Matthias (Hrsg.): *Geschäftsprozessorientierte Systementwicklung*. Springer Vieweg, Wiesbaden, S. 11–23.
- [Sturm 2008] Sturm, Arnon (2008): *Enabling Off-line Business Process Analysis: A Transformation-based Approach*. In: *9th International Workshop on Business Process Modeling, Development and Support (BPMDS)*, 2008, Montpellier, Proceedings. S. 67-75.
- [Sturm 2012] Sturm, Arnon (2012): *Supporting Business Process Analysis via Data Warehousing*. In: *Journal of Software: Evolution and Process*, 24(2012), Nr. 3, S. 303-319.
- [Sun 2014] Sun, Xuanpu (2014): *Ein szenario- und prototypingbasiertes Konzept zur Informationsbedarfsanalyse für Business-Process-Intelligence-Systeme – Entwicklung und Evaluation*. Stuttgart, Universität Stuttgart, Dissertation. Eul, Lohmar.
- [Szyperski et al. 2002] Szyperski, Clemens; Gruntz, Dominik; Murer, Stephan (2002): *Component software: Beyond object-oriented programming*. 2. Aufl., ACM Press and Addison-Wesley, New York.
- [Thomas 2006] Thomas, Oliver (2006): *Das Referenzmodellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation*. In: *Veröffentlichungen des Instituts für Wirtschaftsinformatik Bd. 187*. Saarbrücken, Universität des Saarlandes.
- [Thomas et al. 2007] Thomas, Oliver; Leyking, Katrina; Dreifus, Florian (2007): *Prozessmodellierung im Kontext serviceorientierter Architekturen*. In: *HMD Praxis der Wirtschaftsinformatik*, 44(2007), Nr. 1, S. 37-46.
- [Totok 2000] Totok, Andreas: *Modellierung von OLAP- und Data-Warehouse-Systemen*. Braunschweig, Technische Universität Braunschweig, Dissertation. Deutscher Universitätsverlag, Wiesbaden.
- [Vitharana et al. 2004] Vitharana, Padmal; Jain, Hemant; Zahedi, Fatemeh: *Strategy-Based Design of Reusable Business Components*. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 34(2004), Nr. 4, S. 460-474.
- [Vogt et al. 2008] Vogt, Tim; Neuhaus, Sebastian; Linden, Markus; Chamoni, Peter (2008): *Business-Intelligence-Konzept auf Basis einer Event-Driven Service-Oriented Architecture*. In: Dinter, Barbara; Winter, Robert; Chamoni, Peter; Gronau, Norbert; Turowski, Klaus (Hrsg.): *Synergien durch Integration und Informationslogistik (DW 2008)*, 2008, St. Gallen, Tagungsband. S. 217–228.
- [W3C 2007] W3C: *Web Services Description Language (WSDL) Version 2.0: Additional MEPs*. URL <http://www.w3.org/TR/wsdl20-additional-meps/> – Zugriffsdatum: 29.07.2017.

- [Wagner et al. 2011a] Wagner, Daniel; Leunig, Benjamin; Suchan, Christian; Frank, Jochen; Ferstl, Otto K. (2011): *Klassifikation von Geschäftsprozessen anhand ihres Flexibilitätsbedarfs*. In: Sinz, Elmar J.; Bartmann, Dieter; Bodendorf, Freimut; Ferstl, Otto K. (Hrsg.): *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. Schriften aus der Fakultät für Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg, Bd. 9, Univ. of Bamberg Press, Bamberg, S. 53-78.
- [Wagner et al. 2011b] Wagner, Daniel; Suchan, Christian; Leunig, Benjamin; Frank, Jochen (2011): *Towards the Analysis of Information Systems Flexibility: Proposition of a Method*. In: Bernstein, Abraham; Schwabe, Gerhard (Hrsg.): *10. Internationale Tagung Wirtschaftsinformatik (WI)*, 2011, Zürich, Tagungsband. S. 808-817.
- [Weske 2012] Weske, Mathias (2012): *Business process management: concepts, languages, architectures*. 2. Aufl., Springer, Berlin.
- [White 2006] White, Colin (2006): *The Next Generation of Business Intelligence: Operational BI*. URL <https://www.yumpu.com/en/document/view/5850235/the-next-generation-of-business-intelligence-operational-bi> – Zugriffsdatum: 29.07.2017
- [Wilde & Hess 2007] Wilde, Thomas; Hess, Thomas (2007): *Forschungsmethoden der Wirtschaftsinformatik*. In: *Wirtschaftsinformatik*, 49(2007), Nr. 4, S. 280-287.
- [Wolf 2015] Wolf, Matthias (2015): *Modellbasierte Spezifikation von RESTful SOA auf Basis flexibler SOM-Geschäftsprozessmodelle*. Bamberg, Otto-Friedrich-Universität Bamberg, Dissertation.
- [Word 2014] Word, Jeffrey (2014): *SAP Hana Essentials*. 5. Aufl., URL <http://www.saphanabook.com> – Zugriffsdatum: 29.07.2017.
- [Wu et al. 2007] Wu, Liya; Barash, Gilad; Bartolini, Claudio (2007): *A Service-oriented Architecture for Business Intelligence*. In: *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 2007, Newport Beach, Proceedings. S. 279-285.



Geschäftsprozesse bilden die Lösungsverfahren betrieblicher Systeme und gleichzeitig die zentralen Artefakte zur Realisierung der unternehmerischen Ziele. In Zeiten dynamischer Märkte mit intensivem Wettbewerb gilt es, die eigenen Geschäftsprozesse stetig zu hinterfragen und Maßnahmen zu deren effizienterer und innovativerer Durchführung abzuleiten. Hier setzt diese gestaltungsorientierte Dissertation mit dem geschäftsprozessorientierten Data-Warehouse-Konzept (gpDWH-Konzept) an. Das zentrale gpDWH-System stellt technische Ablaufinformationen zu Prozessen bereit und ermöglicht deren Bewertung im Kontext der betrieblichen Leistungserstellung. Ein Geschäftsprozess kann damit umfassend aus unterschiedlichen Perspektiven analysiert werden. Daraus ableitbare Maßnahmen können vom kurzfristigen steuernden Eingreifen in eine Prozessinstanz bis hin zur mittelfristigen Anpassung der Prozessstrukturen reichen. Das gpDWH-Konzept ist ein umfassender Lösungsansatz mit Referenzarchitektur, DWH-Schema und Entwicklungsmethodik. Über die Referenzarchitektur wird sowohl die Architektur des gpDWH-Systems als auch dessen Integration mit operativen Systemen beschrieben. Der Herausforderung sich stetig verändernder Geschäftsprozesse wird durch das komponentenorientierte Paradigma sowie durch ein generisches DWH-Schema begegnet. Die Entwicklungsmethodik ermöglicht die Anwendung des Konzeptes in betrieblichen Szenarien und unterstützt den Entwickler bei der Beherrschung der Komplexität operativ-dispositiver Systeme. Als Grundlage dienen das Semantische Objektmodell (SOM) sowie das Semantische Data-Warehouse-Modell (SDWM), wobei die Methodik auch losgelöst von diesen Ansätzen anwendbar ist. Zur Demonstration der Ergebnisse der Dissertation dient eine durchgehende Fallstudie zur Vermietung von Elektrofahrzeugen.



eISBN: 978-3-86309-558-1



www.uni-bamberg.de/ubp