

Modellierung der Unternehmensarchitektur

Weiterentwicklung einer bestehenden Methode und
deren Abbildung in einem Meta-Modellierungswerkzeug

DISSERTATION

der Universität St. Gallen,
Hochschule für Wirtschafts-,
Rechts- und Sozialwissenschaften (HSG)

zur Erlangung der Würde eines
Doktors der Wirtschaftswissenschaften

vorgelegt von

Christian Braun

aus Deutschland

Genehmigt auf Antrag der Herren

Prof. Dr. Robert Winter

und

Prof. Dr. Walter Brenner

Dissertation Nr. 3285

Logos Verlag, Berlin 2007

Die Universität St. Gallen, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG), gestattet hiermit die Drucklegung der vorliegenden Dissertation, ohne damit zu den darin ausgesprochenen Anschauungen Stellung zu nehmen.

St. Gallen, den 22. Januar 2007

Der Rektor:

Prof. Ernst Mohr, PhD

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner mehrjährigen Forschungstätigkeit als wissenschaftlicher Mitarbeiter am Institut für Wirtschaftsinformatik (IWI-HSG). Die praxisorientierte Forschung am Institut in Form so genannter Kompetenzzentren, in denen zusammen mit Unternehmensvertretern in Workshops und bilateralen Projekten Lösungsansätze zu übergeordneten Forschungsthemen erarbeitet werden, bot ein einzigartiges Forschungsumfeld. Rückblickend haben vor allem die Menschen dieses Umfeldes die erfolgreiche Fertigstellung meiner Arbeit ermöglicht.

Mein besonderer Dank gilt Herrn Prof. Dr. Robert Winter, der die Dissertation von der Themenfindung bis zur Veröffentlichung intensiv betreute und mir ein ausgezeichnetes Arbeits- und Forschungsumfeld am Institut bot. Herrn Prof. Dr. Walter Brenner danke ich für die Übernahme des Korreferats und seine wertvollen Anregungen.

Herrn Dr. Jochen Müller, Studienleiter des Executive MBA in Business Engineering (MBE HSG), und Herrn Dr. Joachim Schelp, Projektleiter des Kompetenzzentrums „Integration Factory“, bin ich für die unkomplizierte und freundschaftliche Zusammenarbeit sowie die fachliche und persönliche Unterstützung dankbar. Desweiteren danke ich dem gesamten MBE-Team mit Monika Schlumpf, Bernadette Mayer, Myriam Groebli und Rudolf Brühwiler für die angenehme Zusammenarbeit, die meine Zeit am Institut positiv prägte.

Prof. Ghiassi, von der Leavey School of Business der Santa Clara University in Kalifornien, sowie dem Schweizerischen Nationalfonds danke ich für die Unterstützung meines einjährigen Forschungsaufenthaltes in den USA. Dieser hat sowohl meine fachlichen Kenntnisse als auch meinen persönlichen und kulturellen Horizont erweitert.

Mein herzlicher Dank gilt meinen Kollegen und Freunden des IWI-HSG Malte Dous, Ronny Fischer, Dr. Malte Geib, Susanne Glissmann, Dr. Clemens Herrmann, Dr. Axel Hochstein, Mario Klesse, Dr. Florian Melchert, Dr. Annette Reichold, Alexander Ritschel, Ragnar Schierholz, Harald Salomann, Dr. Alexander Schwinn, Christian Wilhelmi, Oliver Wilke, Dr. Felix Wortmann und Dr. Gregor Zellner. Gerne blicke ich nicht nur auf die freundschaftliche Zusammenarbeit, sondern vor allem auch auf die schönen gemeinsamen Erlebnisse ausserhalb der Institutsarbeit zurück.

Praxisorientierte Forschung kann nur gelingen, wenn die entwickelten Konzepte auf ihre Praxistauglichkeit überprüft werden können. Prof. Karagiannis der Universität Wien sowie Franz Bayer der BOC Business Object Consulting danke ich deshalb für die Unterstützung bei der technischen Umsetzung der Konzepte meiner Arbeit in einem Softwareprototypen. Dadurch konnten die Konzepte in der Praxis überprüft werden. Hierfür danke ich insbesondere Dario Bee, Urs Bernet und Fiorenzo Maletta, die für mich wertvolle Ansprechpartner für die Aufnahme der Fallstudie bei der Winterthur Group waren.

Ausserdem danke ich herzlich Kuno Braun für das gründliche Korrekturlesen am Ende der Erstellung der Arbeit.

Ein besonders herzlicher Dank gilt meiner Freundin Adele Alimkulova für die Unterstützung, Motivation und das Verständnis während der Dissertationsphase.

Mein grösster Dank gilt schliesslich meinen Eltern Linde und Helmut Braun, ohne deren Unterstützung und Förderung meine akademische Ausbildung sowie die Fertigstellung der Dissertation nicht möglich gewesen wären. Ihnen widme ich diese Arbeit von ganzem Herzen.

Zürich, im Januar 2007

Christian Braun

Inhaltsübersicht

Vorwort	iii
Inhaltsübersicht	v
Inhaltsverzeichnis	viii
Abkürzungsverzeichnis	xiv
Abbildungsverzeichnis	xvi
Tabellenverzeichnis	xx
Kurzfassung	xxii
Abstract	xxiii
1 Einführung	1
1.1 Problemstellung und Handlungsbedarf	1
1.2 Ziele und Adressaten der Arbeit.....	4
1.3 Forschungsmethodik	6
1.4 Aufbau der Arbeit.....	10
2 Konzeptionelle Grundlagen der Unternehmensmodellierung	12
2.1 Methoden.....	12
2.2 Modelle.....	15
2.3 Modellierung der Unternehmensarchitektur	24
2.4 Werkzeugkonstruktion und -unterstützung	33
2.5 Ziele und treibende Kräfte der Unternehmensarchitektur.....	38
2.6 Konsequenzen für das weitere Vorgehen.....	43
3 Bisherige relevante Ansätze	47
3.1 Auswahl der Ansätze.....	48
3.2 Diskussion ausgewählter Methoden aus der Wissenschaft	50
3.3 Diskussion ausgewählter Bezugsrahmen aus der Praxis.....	74
3.4 Zusammenfassung und Übersicht der Bewertung.....	85
4 Ansatz zur Modellierung der Unternehmensarchitektur	89
4.1 Modellierung auf Strategieebene	92
4.2 Modellierung auf Organisationsebene	109

4.3	Modellierung auf Informationssystemebene	132
4.4	Modellierung auf Infrastrukturebene	150
4.5	Vorschlag zur Integration serviceorientierter Konzepte	152
4.6	Ebeneninterne Abhängigkeiten zwischen Modellen	161
4.7	Ebenenübergreifende Abhängigkeiten	171
4.8	Detaillierungsgrad und Schnittstellen zu domänenspezifischen Architekturmodellen und Modellierungssprachen	178
4.9	Zusammenfassung	181
5	Konstruktion eines Prototyps mit Hilfe eines Metamodellierungswerkzeuges.....	182
5.1	Auswahl eines geeigneten Werkzeuges für die Abbildung des Ansatzes	183
5.2	Beschreibung der Abbildung in ADONIS	186
5.3	Überblick über den Aufbau und die generellen Funktionalitäten	192
5.4	Erstellen von Modellen im Modelleditor	194
5.5	Durchgängigkeit und Abbildung der Gesamtzusammenhänge	195
5.6	Analyse der erstellten Modelle.....	200
5.7	Export/Import und Dokumentation	204
6	Multiperspektivische Evaluation.....	206
6.1	Fallstudie bei der Winterthur Group	209
6.2	Metamodellbasierter Vergleich mit bestehenden Ansätzen	235
6.3	Natürlichsprachliche und merkmalsbasierte Evaluation.....	240
7	Zusammenfassung und Ausblick.....	244
7.1	Zusammenfassung der Arbeit.....	244
7.2	Kritische Würdigung	247
7.3	Zukünftiger Forschungs- und Entwicklungsbedarf.....	249
Anhang A:	Dokumentation der Abbildung der Metamodelle in ADONIS.....	252
A.1.	Strategieebene	252
A.2.	Organisationsebene	256
A.3.	Applikationsebene.....	262
A.4.	Softwarekomponenten- und Datenstrukturebene.....	265
A.5.	Infrastrukturebene	267

A.6. IT-Services	268
Anhang B: Darstellung einer Auswahl kommerziell verfügbarer Modellierungswerkzeuge.....	270
Anhang C: Ansprechpartner bei der Winterthur Group.....	279
Literatur	281

Inhaltsverzeichnis

Vorwort	iii
Inhaltsübersicht	v
Inhaltsverzeichnis	viii
Abkürzungsverzeichnis	xiv
Abbildungsverzeichnis	xvi
Tabellenverzeichnis	xx
Kurzfassung	xxii
Abstract	xxiii
1 Einführung	1
1.1 Problemstellung und Handlungsbedarf	1
1.2 Ziele und Adressaten der Arbeit.....	4
1.3 Forschungsmethodik	6
1.4 Aufbau der Arbeit.....	10
2 Konzeptionelle Grundlagen der Unternehmensmodellierung.....	12
2.1 Methoden.....	12
2.1.1 Method Engineering	12
2.1.2 Methodenbestandteile	14
2.2 Modelle.....	15
2.2.1 Modell und Modellierung	15
2.2.2 Referenzmodell und Referenzmodellierung	18
2.2.3 Metamodel und Metamodellierung.....	20
2.2.4 Verwendete Meta-Modellierungssprachen	21
2.2.4.1 Notation für Metamodelle	22
2.2.4.2 Notation für Vorgehensmodelle	23
2.3 Modellierung der Unternehmensarchitektur	24
2.3.1 Darstellung von Gesamtzusammenhängen in Unternehmungen	25
2.3.2 Business Engineering.....	27
2.3.3 Methoden und Konsistenz	31

2.3.4 Die BAI-Methode	32
2.4 Werkzeugkonstruktion und -unterstützung	33
2.5 Ziele und treibende Kräfte der Unternehmensarchitektur.....	38
2.6 Konsequenzen für das weitere Vorgehen.....	43
2.6.1 Anforderungen an einen Ansatz zur Modellierung der Unternehmensarchitektur.....	43
2.6.2 Anforderungen an die Werkzeugunterstützung	44
3 Bisherige relevante Ansätze	47
3.1 Auswahl der Ansätze.....	48
3.2 Diskussion ausgewählter Methoden aus der Wissenschaft.....	50
3.2.1 Architektur integrierter Informationssysteme (ARIS).....	50
3.2.1.1 Übersicht und Architekturebenen.....	50
3.2.1.2 Metamodelle von ARIS.....	52
3.2.1.3 Weitere strukturelle Bestandteile	59
3.2.1.4 Inhaltliche Charakterisierung	60
3.2.2 Multi-Perspective Enterprise Modeling (MEMO).....	61
3.2.2.1 Übersicht und Architekturebenen.....	61
3.2.2.2 Metamodelle von MEMO	62
3.2.2.3 Weitere strukturelle Bestandteile	65
3.2.2.4 Inhaltliche Charakterisierung	65
3.2.3 Semantisches Objektmodell (SOM)	67
3.2.3.1 Übersicht und Architekturebenen.....	67
3.2.3.2 Metamodelle der SOM-Methode.....	68
3.2.3.3 Weitere strukturelle Bestandteile	72
3.2.3.4 Inhaltliche Charakterisierung	73
3.3 Diskussion ausgewählter Bezugsrahmen aus der Praxis.....	74
3.3.1 Das Zachman Framework.....	74
3.3.1.1 Überblick und Architekturebenen	74
3.3.1.2 Methodenbestandteile des Zachman Framework.....	76
3.3.1.3 Inhaltliche Charakterisierung	77

3.3.2	Federal Enterprise Architecture Framework (FEAF)	78
3.3.2.1	Überblick und Architekturebenen	78
3.3.2.2	Methodenbestandteile von FEAF	79
3.3.2.3	Inhaltliche Charakterisierung	80
3.3.3	The Open Group Architecture Framework (TOGAF)	81
3.3.3.1	Überblick und Architekturebenen	81
3.3.3.2	Methodenbestandteile von TOGAF	81
3.3.3.3	Inhaltliche Charakterisierung	84
3.4	Zusammenfassung und Übersicht der Bewertung	85
4	Ansatz zur Modellierung der Unternehmensarchitektur	89
4.1	Modellierung auf Strategieebene	92
4.1.1	Vorgehensmodell	94
4.1.2	Metamodell	95
4.1.2.1	Metamodell des Geschäftsnetzwerks	95
4.1.2.2	Metamodell des Kundenprozessmodells	97
4.1.2.3	Metamodell des Leistungsmodells	98
4.1.2.4	Metamodell des Zielsystems	100
4.1.2.5	Definition der Metaentitätstypen auf Strategieebene	102
4.1.2.6	Festlegung der verwendeten Symbole auf Strategieebene	108
4.2	Modellierung auf Organisationsebene	109
4.2.1	Vorgehensmodell	111
4.2.2	Metamodell	114
4.2.2.1	Metamodell der Prozesslandkarte	114
4.2.2.2	Metamodell der Prozessvision	115
4.2.2.3	Metamodell der Prozessführung	116
4.2.2.4	Metamodell der Aufbauorganisation	118
4.2.2.5	Metamodell der Ablauforganisation	120
4.2.2.6	Metamodell der Informationslandkarte	124
4.2.2.7	Definition der Metaentitätstypen auf Organisationsebene	126

4.2.2.8	Festlegung der verwendeten Symbole auf Organisationsebene	131
4.3	Modellierung auf Informationssystemebene	132
4.3.1	Vorgehensmodell	134
4.3.2	Metamodell	137
4.3.2.1	Metamodell der Applikationslandschaft	137
4.3.2.2	Metamodell des Geschäftsfunktions- und Informationsobjektmodells.....	139
4.3.2.3	Metamodell der Integrationsarchitektur	140
4.3.2.4	Metamodell der Softwarekomponenten und Plattformen	140
4.3.2.5	Metamodell des Datenstrukturmodells.....	142
4.3.2.6	Metamodell der Autorisierung	143
4.3.2.7	Definition der Metaentitätstypen auf Informationssystemebene.....	145
4.3.2.8	Festlegung der verwendeten Symbole auf Informationssystemebene	149
4.4	Modellierung auf Infrastrukturebene	150
4.5	Vorschlag zur Integration serviceorientierter Konzepte	152
4.6	Ebeneninterne Abhängigkeiten zwischen Modellen.....	161
4.6.1	Strategieebene	161
4.6.2	Organisationsebene	164
4.6.3	Informationssystemebene	168
4.7	Ebenenübergreifende Abhängigkeiten	171
4.7.1	Zusammenhänge zwischen Strategie- und Organisationsebene	172
4.7.2	Zusammenhänge zwischen Organisations- und Informationssystemebene.....	174
4.8	Detaillierungsgrad und Schnittstellen zu domänenspezifischen Architekturmodellen und Modellierungssprachen	178
4.9	Zusammenfassung	181
5	Konstruktion eines Prototyps mit Hilfe eines Metamodellierungswerkzeuges.....	182
5.1	Auswahl eines geeigneten Werkzeuges für die Abbildung des Ansatzes.....	183
5.2	Beschreibung der Abbildung in ADONIS	186
5.2.1	Metamodellierungskonzept.....	186
5.2.2	Basiskonzepte in ADONIS	187
5.2.3	Das ADONIS-Administrations-Toolkit.....	188

5.2.4	Abbildung der Metamodelle	190
5.3	Überblick über den Aufbau und die generellen Funktionalitäten	192
5.4	Erstellen von Modellen im Modelleditor	194
5.5	Durchgängigkeit und Abbildung der Gesamtzusammenhänge	195
5.6	Analyse der erstellten Modelle.....	200
5.7	Export/Import und Dokumentation	204
6	Multiperspektivische Evaluation.....	206
6.1	Fallstudie bei der Winterthur Group	209
6.1.1	Allgemeine Informationen zur Winterthur Group	209
6.1.2	Ausgangssituation und Problembeschreibung	210
6.1.3	Ein zentral gepflegtes Metamodell als Lösungsansatz	212
6.1.4	Metamodell der Winterthur Group	214
6.1.5	Vergleich der Metamodelle	216
6.1.6	Abbildung mit Hilfe des Prototyps	220
6.1.7	Fazit	234
6.2	Metamodellbasierter Vergleich mit bestehenden Ansätzen	235
6.2.1	Metamodellbasierter Vergleich mit ARIS	235
6.2.2	Metamodellbasierter Vergleich mit MEMO	237
6.2.3	Metamodellbasierter Vergleich mit SOM.....	239
6.2.4	Fazit	240
6.3	Natürlichsprachliche und merkmalsbasierte Evaluation.....	240
7	Zusammenfassung und Ausblick.....	244
7.1	Zusammenfassung der Arbeit.....	244
7.2	Kritische Würdigung	247
7.3	Zukünftiger Forschungs- und Entwicklungsbedarf.....	249
Anhang A:	Dokumentation der Abbildung der Metamodelle in ADONIS.....	252
A.1.	Strategieebene	252
A.2.	Organisationsebene	256
A.3.	Applikationsebene.....	262

A.4. Softwarekomponenten- und Datenstrukturebene.....	265
A.5. Infrastrukturebene	267
A.6. IT-Services	268
Anhang B: Darstellung einer Auswahl kommerziell verfügbarer Modellierungswerkzeuge.....	270
Anhang C: Ansprechpartner bei der Winterthur Group.....	279
Literatur	281

Abkürzungsverzeichnis

ADL	ADONIS Definition Language
AQL	ADONIS Query Language
ADM	Architecture Development Method
ARIS	Architektur integrierter Informationssysteme
BAI	Bankenarchitekturen im Informationszeitalter
BE	Business Engineering
BPMN	Business Process Management Notation
CASE	Computer Aided Software Engineering
CIM	Computation Independent Model
CWM	Common Warehouse Metamodel
DoDAF	Department of Defense Architecture Framework
DS	Design Science
EA	Enterprise Architecture
EAM	Enterprise Architecture Manager
eEPK	Erweiterte ereignisgesteuerte Prozesskette
EPK	Ereignisgesteuerte Prozesskette
ESP	Exclusive Service Provider
FEAF	Federal Enterprise Architecture Framework
IAS	Interaktionsschema
IFEAD	Institute For Enterprise Architecture Development
IKS	Informations- und Kommunikationssysteme
IO	Informationsobjekt
IS	Informationssystem
ISA	Informationssystem-Architektur
ITIL	IT Infrastructure Library
ITSM	IT-Service-Management
IWI-HSG	Institut für Wirtschaftsinformatik der Universität St. Gallen
J2EE	Java 2 Platform Enterprise Edition
KEF	Kritischer Erfolgsfaktor
KOS	Konzeptuelles Objektschema
MDA	Model Driven Architecture

MEMO	Multi-Perspective Enterprise Modeling
OE	Organisationseinheit
OE des PM	Organisationseinheit des Prozessmanagements
OLA	Operational Level Agreement
OML	OPEN Modeling Language
PIM	Platform Independent Model
PL	Prozesslandkarte
PS	Public Service
PSM	Platform Specific Model
SI	Service Integrator
SITM	Strategic IT Management
SOA	Serviceorientierte Architektur
SOM	Semantisches Objektmodell
SOP	Solution Evaluation Process
SSP	Shared Service Provider
SW	Software
TEAF	Treasury Enterprise Architecture Framework
TOGAF	The Open Group Architecture Framework
UML	Unified Modeling Language
VES	Vorgangs-Ereignisschema
VOS	Vorgangs-Objektschema
WGR	Winterthur Group

Abbildungsverzeichnis

Abbildung 1: Information Systems Research Framework	8
Abbildung 2: Aufbau der Arbeit.....	11
Abbildung 3: Bestandteile einer Methode	14
Abbildung 4: Bestandteile einer Modellierungstechnik	16
Abbildung 5: Strukturierung von Modellen	18
Abbildung 6: Referenzmodellanwendung	19
Abbildung 7: Modellierungshierarchie.....	21
Abbildung 8: Unternehmensarchitektur	27
Abbildung 9: Business-Engineering-Landkarte	29
Abbildung 10: Vereinfachtes Metamodell des BE	30
Abbildung 11: Beispiel für Wechselwirkungen zwischen den Ebenen.....	32
Abbildung 12: Metamodellierungskonzepte für die Werkzeugkonstruktion	35
Abbildung 13: Kategorisierung der Werkzeugunterstützung.....	37
Abbildung 14: ARIS-Haus	51
Abbildung 15: Metamodell des strategischen Geschäftsprozessmodells.....	53
Abbildung 16: Metamodell des strategischen Vorgangskettendiagramms	54
Abbildung 17: Metamodell der Funktionssicht	55
Abbildung 18: Metamodell der Organisationssicht.....	56
Abbildung 19: Metamodell der Datensicht (Makro).....	57
Abbildung 20: Metamodell der Leistungssicht	58
Abbildung 21: Metamodell der Steuerungssicht	59
Abbildung 22: Multi-Perspective Enterprise Modeling (MEMO)	62
Abbildung 23: Metamodell der MEMO-SML (Strategy Modeling Language)	63
Abbildung 24: Metamodell der MEMO-OrgML	64
Abbildung 25: Metamodell der MEMO-OML	65
Abbildung 26: SOM-Ebenen	67
Abbildung 27: Metamodell der Geschäftsprozessebene	70

Abbildung 28: Metamodell der Anwendungsebene	71
Abbildung 29: Das Zachman Framework	75
Abbildung 30: Federal Enterprise Architecture Framework	79
Abbildung 31: TOGAF ADM	82
Abbildung 32: Gestaltungsebenen und Sichten des entwickelten Ansatzes.....	89
Abbildung 33: Vorgehensmodell auf Strategieebene (Level 1)	95
Abbildung 34: Metamodell des Geschäftsnetzwerkmodells	97
Abbildung 35: Metamodell des Kundenprozessmodells.....	98
Abbildung 36: Metamodell des Leistungsmodells	99
Abbildung 37: Metamodell des Zielsystems (BSC).....	101
Abbildung 38: Vorgehensmodell auf Organisationsebene (Level 1)	111
Abbildung 39: Metamodell der Prozesslandkarte	115
Abbildung 40: Metamodell der Prozessvision	116
Abbildung 41: Metamodell der Prozessführung.....	117
Abbildung 42: Metamodell der Aufbauorganisation.....	119
Abbildung 43: Metamodell der Ablauforganisation.....	121
Abbildung 44: Metamodell der Informationslandkarte	125
Abbildung 45: Vorgehensmodell auf Informationssystemebene (Level 1).....	134
Abbildung 46: Metamodell der Applikationslandschaft	138
Abbildung 47: Metamodell des Geschäftsfunktionen- und Informationsobjektmodells.....	139
Abbildung 48: Metamodell der Integrationsarchitektur	140
Abbildung 49: Metamodell der Softwarekomponenten und Datenstrukturen.....	141
Abbildung 50: Vereinfachtes Metamodell des UML-Klassendiagramms	143
Abbildung 51: Metamodell der Autorisierung	144
Abbildung 52: Metamodell des Infrastrukturmodells	151
Abbildung 53: Metamodell der IT-Servicesicht.....	154
Abbildung 54: Integration der IT-Servicesicht.....	157
Abbildung 55: Metamodell der SOA-Sicht.....	160
Abbildung 56: Verknüpfung der Modelle auf Strategieebene (OE).....	163

Abbildung 57: Verknüpfung der Modelle auf Strategieebene (Marktleistung)	164
Abbildung 58: Verknüpfung der Modelle auf Organisationsebene (Prozess).....	166
Abbildung 59: Verknüpfung der Modelle auf Organisationsebene (Organisationseinheit)...	167
Abbildung 60: Verknüpfung der Modelle auf Organisationsebene (Führungsgrösse).....	168
Abbildung 61: Verknüpfung der Modelle auf Informationssystemebene (Applikation)	169
Abbildung 62: Verknüpfung der Modelle auf Informationssystemebene (OE).....	170
Abbildung 63: Zusammenhänge zwischen Strategie- und Organisationsebene.....	173
Abbildung 64: Zusammenhänge zwischen Organisations- und Informationssystemebene	176
Abbildung 65: Schnittstellen zu domänenspezifischen Architekturen.....	180
Abbildung 66: Architektur der ADONIS-Plattform.....	187
Abbildung 67: Basiskonzepte in ADONIS.....	188
Abbildung 68: Konzepte des ADONIS-Administrations-Toolkits	189
Abbildung 69: Schritte bei der Implementierung des Metamodells in ADONIS	190
Abbildung 70: Komponenten und Funktionalitäten des Prototyps	194
Abbildung 71: Anlage eines neuen Modells	195
Abbildung 72: Modellübergreifende Referenzen	196
Abbildung 73: Prozesshierarchie.....	197
Abbildung 74: Beispiel Prozessführung.....	198
Abbildung 75: Referenzbaum.....	200
Abbildung 76: Standardisierte und benutzerdefinierte Abfragen.....	201
Abbildung 77: Beziehungstabelle.....	202
Abbildung 78: Pfadanalyse und Simulation	203
Abbildung 79: Dokumentation und Veröffentlichung im Intranet.....	205
Abbildung 80: Multiperspektivische Evaluation.....	206
Abbildung 81: Vereinfachtes Metamodell der Winterthur Group.....	214
Abbildung 82: Sichten und Inhalte des Metamodells der WGR.....	215
Abbildung 83: Dokumentation des WGR-Metamodells im Intranet.....	216
Abbildung 84: Geschäftsnetzwerk.....	221
Abbildung 85: Kundenprozessmodell	223

Abbildung 86: Prozesslandkarte der WGR	224
Abbildung 87: Ablaufplanung Schadenbearbeitung.....	226
Abbildung 88: Aufbauorganisation	227
Abbildung 89: Geschäftsfunktionen- und Informationsobjektmodell.....	228
Abbildung 90: Applikationsbestandsführung.....	229
Abbildung 91: Softwarekomponenten.....	229
Abbildung 92: Beziehungstabellen (Applikationsdomäne zu Prozess).....	230
Abbildung 93: Auswertung Metamodellbeziehungen.....	232
Abbildung 94: Impact-Analyse (Szenario „Plattform fällt aus“)	233

Tabellenverzeichnis

Tabelle 1: Richtlinien für die Erfüllung des Design-Science-Paradigma.....	10
Tabelle 2: Modellierungskonstrukte für das Metamodell	22
Tabelle 3: Modellierungskonstrukte für das Vorgehensmodell	24
Tabelle 4: Bewertungskriterien	47
Tabelle 5: Übersicht der analysierten Ansätze	49
Tabelle 6: Übersicht über die Bewertung existierender Ansätze	87
Tabelle 7: Domänenspezifische Literaturquellen	90
Tabelle 8: Metaentitätstypen der Strategieebene.....	108
Tabelle 9: Symbole der Strategieebene	109
Tabelle 10: Metaentitätstypen der Organisationsebene.....	131
Tabelle 11: Symbole der Organisationsebene	132
Tabelle 12: Metaentitätstypen der Informationssystemebene	149
Tabelle 13: Symbole der Informationssystemebene.....	150
Tabelle 14: Metaentitätstypen der Infrastrukturebene.....	152
Tabelle 15: Symbole der Infrastrukturebene	152
Tabelle 16: Metaentitätstypen der IT-Servicesicht.....	156
Tabelle 17: Symbole der IT-Servicesicht	156
Tabelle 18: Metaentitätstypen der SOA-Sicht.....	161
Tabelle 19: Wesentliche Anforderungen an die Werkzeugunterstützung.....	183
Tabelle 20: Übersicht über kommerziell verfügbare Werkzeuge.....	184
Tabelle 21: Erfüllung der wichtigsten Kriterien.....	185
Tabelle 22: Modelltyp Geschäftsnetzwerk	192
Tabelle 23: Fakten zur Winterthur Group	210
Tabelle 24: Vergleich Metaentitätstypen Ablauforganisation.....	217
Tabelle 25: Vergleich Metaentitätstypen Aufbauorganisation.....	218
Tabelle 26: Vergleich Metaentitätstypen Softwarekomponenten	219
Tabelle 27: Metamodellbasierter Vergleich mit ARIS.....	236
Tabelle 28: Metamodellbasierter Vergleich mit MEMO	238

Tabelle 29: Metamodellbasierter Vergleich mit SOM	240
Tabelle 30: Merkmalbasierte Evaluierung	243

Kurzfassung

Die Entwicklung einer Unternehmensarchitektur ist eine fundamentale Voraussetzung für die Bereitstellung von Informationssystemen, die mit den organisatorischen und strategischen Richtlinien des Unternehmens konsistent sind. Eine für das gesamte Unternehmen definierte und zentral gepflegte Architektur trägt dazu bei, redundante und inkonsistente Informationen über das Unternehmen zu beseitigen. Sie dient als zentrales Koordinationsinstrument zwischen unterschiedlichen Unternehmensbereichen für die integrierte Gestaltung von Informationssystemen, Organisationsstrukturen und Unternehmensstrategien. Sowohl in der Praxis als auch in der Wissenschaft fehlt bisher ein allgemein akzeptiertes Verständnis der wesentlichen Ebenen, Elemente und Beziehungen einer Unternehmensarchitektur. Vor allem existiert für diese keine umfassende, (semi-)formale Spezifikation. Ziel der vorliegenden Arbeit ist daher die Entwicklung eines Ansatzes zur Modellierung der Unternehmensarchitektur und dessen Abbildung in einem kommerziellen Metamodellierungswerkzeug zur Sicherstellung der Anwendbarkeit. Im Mittelpunkt steht dabei zum einen die Definition eines ebenen- und sichtenübergreifenden Metamodells, das die grundlegenden Elemente eines Unternehmens sowie deren Beziehungen festlegt. Zum anderen wird ein Software-Prototyp mit Hilfe des Metamodellierungswerkzeuges erstellt. Damit sollen die Konsistenz und Umsetzbarkeit des entwickelten Metamodells gezeigt werden. Darüber hinaus erfolgt eine multiperspektivische Evaluation des Prototyps und des zugrunde liegenden Metamodells in Form einer Fallstudie sowie einer natürlichsprachlichen, merkmalsbasierten und metamodellbasierten Evaluierung. Den Forschungsrahmen der Dissertation bilden das Business Engineering sowie die Prinzipien des Methoden-Engineering. Der entwickelte Ansatz umfasst die drei Gestaltungsebenen „Geschäftsstrategie“, „Organisation“ und „Informationssystem“. Der Dissertation liegen die konstruktiven Forschungsansätze „Deduktion“, „Modellierung“ sowie „Entwicklung und Test eines Prototyps“ zugrunde. Sie lässt sich auch dem Design-Science-Paradigma zuordnen, welches im Umfeld des „Information Systems Research“, dem angelsächsischen Pendant der Wirtschaftsinformatik, anerkannt ist. Ausgangspunkt für die Entwicklung des Ansatzes bilden eine bereits bestehende Methode, eine umfangreiche Literaturanalyse sowie ein detaillierter Vergleich existierender Ansätze.

Schlüsselwörter: Unternehmensarchitektur, Modellierung, Metamodellierung, Business Engineering, Methoden-Engineering, Werkzeugunterstützung, Softwareprototyp, Metamodellierungsplattform

Abstract

The design of information systems as well as the redesign of corporate strategies and organizational structures are both challenging tasks. They need to be coordinated in order to provide information systems that are consistent with the strategic and organizational guidelines of the enterprise. To describe the structure of information systems, both from a technical and a business view, the design of information systems is increasingly integrated with adequate methods on other levels of abstraction and discussed as a broad or holistic design concept called “Enterprise Architecture”. The goal of enterprise architecture is to represent the whole structure of an enterprise and its information systems in order to support the future development and design of the enterprise. In recent years, both in science as well as in practice, various approaches for modeling the enterprise architecture have been developed on different levels of abstraction. These different levels reflect the well known cultural differences between business and information technology professionals. Holistic approaches use a framework or a hierarchy of design layers to structure the different views on an enterprise. These approaches are often rather abstract, do not consider all design layers, or provide various views on the enterprise without defining specific modeling languages. Many of these approaches have less elaborated meta models, which integrate the models on different layers. Furthermore, some of them are too much focused on domain-specific concepts (e.g. software development or business process management). This dissertation develops an approach for the design of enterprise architectures, which provides models on all relevant design layers and gives a holistic view on all key artefacts and their relationships within an enterprise that are relevant for strategic decisions in the overlapping area between business and IT (e.g. services, performance indicators, business processes, organizational units, applications, software components). The applicability of the proposed approach and its meta model is shown by means of a software prototype that is tested in practice. The research model is based on the Business Engineering (BE) approach and the guidelines of Method Engineering (ME). It incorporates the three BE design layers business strategy, organization, and information system. The underlying constructive research methodologies are deduction, modeling, as well as development and test of a prototype. These are considered valid research approaches of Information Systems Research (ISR) in the German-speaking community. Furthermore, this dissertation can also be assigned to design science, a generally accepted research approach in the Anglo-American ISR community. The starting point for the development of the meta model and prototype is an existing method and a comprehensive analysis of related work in the field of enterprise modeling.

Key Words: Enterprise Architecture, Modeling, Metamodeling, Business Engineering, Method Engineering, Tool Construction, Software Prototype, Metamodeling Platform

1 Einführung

1.1 Problemstellung und Handlungsbedarf

Die Planung, Einführung und der Betrieb von Informationssystemen stellen eine grosse Herausforderung dar. Gerade Unternehmen mit überwiegend immateriellen Produkten, wie beispielsweise Finanzdienstleister, sind bei der Entwicklung, dem Vertrieb und der Abwicklung ihrer Produkte bzw. Leistungen stark an informationsverarbeitende Systeme gebunden.¹ Zudem müssen sich Unternehmen des Informationszeitalters dessen bewusst sein, dass neue Informationstechnologien ein „Enabler“ für neue Geschäftsmodelle bzw. -lösungen sein können.² Strategische und organisatorische Veränderungen müssen deshalb die Informationssysteme mit einbeziehen. Umgekehrt müssen aber auch bei der Einführung neuer Informationstechnologien strategische und organisatorische Aspekte berücksichtigt und gegebenenfalls verändert werden.

Der Entwurf von Informationssystemen sowie die Umgestaltung von Unternehmensstrategien und Organisationsstrukturen sind für sich genommen bereits herausfordernde Aufgaben. Sie bedürfen zudem einer Koordination, um Informationssysteme bereitstellen zu können, die mit den strategischen und organisatorischen Richtlinien des Unternehmens konsistent sind.³ Um die Struktur von Informationssystemen sowohl aus technischer als auch aus fachlicher Sicht beschreiben zu können, wird die Gestaltung der Informationssystem-Architektur zunehmend mit entsprechenden Methoden auf anderen Betrachtungsebenen integriert und als umfassendes Gestaltungskonzept „Unternehmensarchitektur“ („Enterprise Architecture“) diskutiert.⁴ Das Ziel einer Unternehmensarchitektur ist es, die Strukturen des Unternehmens und seiner Informationssysteme ganzheitlich abzubilden und damit die zukünftige Entwicklung und Gestaltung des Unternehmens zu unterstützen.⁵

In den letzten Jahren sind sowohl in der Wissenschaft als auch in der Praxis zahlreiche Ansätze zur Modellierung der Unternehmensarchitektur auf unterschiedlichen Abstraktionsebenen entwickelt worden.⁶ Diese unterschiedlichen Abstraktionsebenen und Sichten auf ein Unternehmen reflektieren die unterschiedlichen Fachkenntnisse und Interessenslagen zwischen Mitarbeitern aus den Fachabteilungen und Mitarbeitern aus dem IT-Bereich.⁷

¹ Vgl. *Winter* (2005), S. 575.

² Vgl. *Baumöl et al.* (2005), *Kalakota/Robinson* (2001) und *Orlikowski/Barley* (2001).

³ Vgl. *Frank* (2002), S. 1.

⁴ Vgl. *Schwinn/Winter* (2005), S. 587.

⁵ Vgl. *Krcmar* (2003), S. 39f.

⁶ Vgl. z.B. *OMG* (2003b), *Firesmith et al.* (1998), *Teubner* (1999), *Frese* (1995), *Heinrich* (2000).

⁷ Vgl. *Frank* (2002), S. 2.

Ansätze zur Gestaltung der Anwendungssysteme sind z.B. das Software Engineering⁸, Information Systems Engineering⁹ und Information Management¹⁰. Aufgrund ihrer Mikrosicht sind diese Ansätze nur bedingt für eine umfassende Abbildung der Unternehmensarchitektur geeignet. Sie berücksichtigen meist nicht oder nur unzulänglich die Modellierung der Unternehmensstrategie und der Geschäftsprozesse.¹¹ Generelle Modellierungssprachen wie beispielsweise UML¹² oder OML¹³ ermöglichen im Prinzip auch die Abbildung von Geschäftsabläufen und Organisationsstrukturen. Sie sind aber primär für die Entwicklung von Softwaresystemen entworfen worden und bieten daher keine Konzepte und grafischen Repräsentationen, die für alle Bereiche der Unternehmensmodellierung geeignet sind. Aspekte der Unternehmensorganisation sowie der strategischen Unternehmensführung werden nur in geringem Umfang berücksichtigt. Zudem bietet beispielsweise die UML eine Vielzahl unterschiedlicher Modelle, deren Zusammenhänge oftmals nicht klar definiert sind.¹⁴ Sie sind daher für Mitarbeiter aus den Fachbereichen nur schwer verständlich.

Ansätze aus dem Bereich der Organisationstheorie¹⁵ (z.B. Business Process Reengineering) und der strategischen Unternehmensführung¹⁶ (z.B. „Market based View“, „Resource based View“, „Value based View“) ermöglichen dagegen zwar die Gestaltung von Unternehmensstrategien und Organisationsstrukturen. Sie basieren meist auf einfachen, grafischen Modellen, mit denen grundlegende Sachverhalte und Beziehungen leicht kommunizierbar und auf hohem abstraktem Niveau dargestellt werden können. Strategie- und Organisationsmodelle basieren aber in der Regel auf unterschiedlichen Konzepten und haben keine Gemeinsamkeiten mit den konzeptionellen Modellen der Softwareentwicklung.

Umfassendere Ansätze¹⁷ (z.B. ARIS, SOM, Zachman) verwenden einen Bezugsrahmen oder eine Hierarchie von Modellierungsebenen, um die verschiedenen Perspektiven auf das Unternehmen zu strukturieren. Diese sind allerdings meist auf einem eher abstrakten Niveau angesetzt, berücksichtigen nicht alle relevanten Modellierungsebenen und/oder stellen eine Vielzahl unterschiedlicher Sichten auf das Unternehmen zur Verfügung (wie z.B. Daten, Funktionen, Personen und Standorte), ohne konkrete Modellierungssprachen und -techniken zu spezi-

⁸ Vgl. z.B. *Balzert* (2000).

⁹ Vgl. z.B. *van Hee* (1994).

¹⁰ Vgl. z.B. *Heinrich* (2002c), *Teubner* (1999).

¹¹ Vgl. z.B. *Jacobson et al.* (1994).

¹² Vgl. *OMG* (2003b).

¹³ Vgl. *Firesmith et al.* (1998).

¹⁴ Vgl. hierzu z.B. *Jonkers et al.* (2004), S. 262.

¹⁵ Vgl. z.B. *Frese* (1995), *Hammer/Champy* (1993), *Hess/Brecht* (1996) und *Teubner* (1999), S. 125-140.

¹⁶ Vgl. z.B. *Porter* (1999), *Rockart* (1979), *Grant* (1998), *Heinrich* (2000), S. 28-38, *Hahn* (1997), S. 157-161 und *Müller-Stewens/Lechner* (2003).

¹⁷ Vgl. z.B. *Ferstl/Sinz* (1995), *Krcmar* (1990), *Österle/Winter* (2003), *Scheer* (1998), *Opengroup* (2002), *FEA* (2001) und *Zachman* (1987).

fizieren. Einige dieser Ansätze stellen vor allem die Gestaltung der Geschäftsprozesse in den Mittelpunkt und/oder fokussieren zu stark auf softwareentwicklungsrelevante Aspekte.

Des Weiteren stehen bei vielen Ansätzen die einzelnen, auf unterschiedlichen Modellierungsebenen erstellten Modelle unabhängig nebeneinander und sind nicht durchgängig miteinander integriert, z.B. durch ein umfassendes, ebenenübergreifendes Metamodell. Ein solches Metamodell ist allerdings eine wesentliche Voraussetzung für die systematische Integration und damit Konsistenz der einzelnen Teilsichten. Bisher existieren keine Konzepte, die die Zusammenhänge zwischen den verschiedenen Modellierungsebenen definieren.¹⁸ Die Definition dieser Zusammenhänge würde die Konsistenz zwischen Modellen auf unterschiedlichen Ebenen gewährleisten und die Durchführung von Impact-Analysen ermöglichen. Damit wäre auch eine wichtige Grundlage für die systematische, modellbasierte Lösung von Problemstellungen des IT-Business-Alignment¹⁹ geschaffen.

Für die praktische Anwendung eines bestimmten Ansatzes ist entscheidend, dass entsprechende unterstützende (Software-)Werkzeuge verfügbar sind, die die Modellierung, Speicherung, Darstellung und Weiterentwicklung von Modellen der Unternehmensarchitektur ermöglichen.²⁰ Da die Methoden zur Gestaltung der Unternehmensarchitektur immer umfassender werden, wird zunehmend auch nach Werkzeugen verlangt, welche die durchgängige Abbildung und Analyse von Geschäftsstrategien, Organisationsstrukturen, Geschäftsprozessen und Applikationen eines Unternehmens unterstützen.²¹ Im Vordergrund steht dabei vor allem die Abbildung der Gesamtzusammenhänge zwischen den einzelnen Gestaltungsebenen, die in den meisten Fällen ein nur schwer zu erkennendes und zu verstehendes System darstellen. Werden diese Zusammenhänge nicht erkannt und verstanden, so ist die Änderung des Systems immer mit einer gewissen Unsicherheit behaftet, da die Auswirkungen nicht abgeschätzt werden können.²²

Viele Unternehmen verwenden bereits einfache Visualisierungswerkzeuge, wie z.B. Microsoft Visio oder Powerpoint, um ihre Architektur zu dokumentieren.²³ Allerdings kann dadurch die Konsistenz und Redundanzfreiheit in den verschiedenen Dokumenten nur schwer überprüft und eingehalten werden. Wenn beispielsweise ein Geschäftsprozess oder eine Applikation in mehreren Modellen auftaucht, dann müssen bei einer Änderung des Geschäftsprozesses bzw. der Applikation alle Modelle aktualisiert werden. Dies birgt die Gefahr von Inkonsistenzen, da die Änderungen in allen Modellen bei diesen Werkzeugen manuell erfolgen müssen. Darüber hinaus können mit diesen Werkzeugen keine Impact-Analysen durchgeführt werden.

¹⁸ Vgl. hierzu z.B. *Jonkers et al.* (2004), S. 264.

¹⁹ Vgl. *Henderson/Venkatraman* (1993).

²⁰ Vgl. *Frank* (2002), S.4.

²¹ Vgl. *Schekkerman* (2004), S. 201.

²² Vgl. *James* (2005a), S. 2.

²³ Vgl. *IFEAD* (2004), S. 26.

1.2 Ziele und Adressaten der Arbeit

Ausgehend von den im vorhergehenden Abschnitt beschriebenen Herausforderungen werden im Folgenden die Forschungsfrage sowie die Ziele und Adressaten des Dissertationsvorhabens definiert.

Im Kompetenzzentrum Bankenarchitekturen im Informationszeitalter (CC BAI)²⁴ wurde gemeinsam mit Partnerunternehmen aus der Finanzindustrie entsprechend den Grundsätzen des Business Engineering²⁵ und dem Methoden-Engineering²⁶ eine Methode zur Gestaltung der Unternehmensarchitektur erarbeitet (BAI-Methode). Da diese bereits einen weitgehend umfassenden Ansatz darstellt, der alle relevanten Gestaltungsebenen eines Unternehmens berücksichtigt sowie auf allen Ebenen Metamodelle und Vorgehensweisen definiert, dient sie als Ausgangsbasis für die vorliegende Arbeit.

Eine weitere Grundlage bildet der in dieser Arbeit durchgeführte, detaillierte Vergleich zwischen bereits existierenden Ansätzen der Unternehmensmodellierung. Die dabei identifizierten Konzepte sowie Stärken und Schwächen sollen anschliessend in die Spezifizierung des in der vorliegenden Arbeit präsentierten Ansatzes und dessen Implementierung einfließen.

Ziel ist es, die BAI-Methode im Sinne der im Grundlagenteil identifizierten Anforderungen sowie der Stärken und Schwächen bereits existierender Ansätze genauer zu definieren und zu erweitern sowie in einem Softwareprototyp abzubilden. Dabei sind insbesondere alle Gestaltungsebenen einer Organisation bzw. eines Unternehmens einzubeziehen und eine möglichst breite Darstellung der Sachverhalte mit einem geringen Fokus auf softwareentwicklungsrelevante Aspekte zu erreichen. Als Ergebnis soll ein ebenenübergreifendes Metamodell zur konsistenten Abbildung aller Artefakte und deren Zusammenhänge vorliegen, die für strategische Entscheidungen im Schnittbereich von Fachbereich und IT relevant sind (z.B. Leistungssystem, Zielsystem, Prozesse, Organisationseinheiten und Applikationen). Da es nicht möglich ist, alle Artefakte in einem Totalmodell abzubilden und ausreichend zu pflegen, wird ein sinnvoller Aggregationsgrad angestrebt.²⁷

Der in dieser Arbeit entwickelte Ansatz soll bereits existierende Ansätze, die auf einen bestimmten Anwendungsbereich fokussiert sind und einen hohen Detaillierungsgrad besitzen, nicht ersetzen, sondern deren Konzepte in aggregierter Form abbilden und miteinander integrieren. Der Fokus liegt dabei auf der Modellierung des Ist-Zustandes der Unternehmensarchitektur und nicht auf deren Transformation. Da allerdings für zukünftige Transformationspro-

²⁴ Vgl. *Leist/Winter* (2002).

²⁵ Vgl. *Österle/Winter* (2003).

²⁶ Vgl. *Heym* (1993) und *Brinkkemper* (1996).

²⁷ Viele Unternehmen sind in der Vergangenheit bereits während der Modellierung eines unternehmensweiten Datenmodells (Totalmodells) an der hohen Komplexität dieser Aufgabe gescheitert. In anderen Fällen wurden die erstellten Modelle nicht mehr gepflegt oder nicht konsequent implementiert. Vgl. hierzu auch *Frie/Auth* (2001), S. 5.

jekte konsistente Informationen über die aktuellen Strukturen des Unternehmens vorliegen sollten, stellt die Entwicklung einer einheitlichen Terminologie in Form eines Metamodells eine wichtige Grundlage dafür dar. Gestaltungsrichtlinien und Handlungsanweisungen für die Transformation der Unternehmensarchitektur sind nicht Teil dieser Arbeit und müssen in nachfolgenden Arbeiten entwickelt werden.

Um die Konsistenz, Umsetzbarkeit und Anwendbarkeit des entwickelten Ansatzes zu illustrieren, wird dieser in einem kommerziellen Metamodellierungswerkzeug abgebildet. Der daraus entstehende (Software-)Prototyp²⁸ wird zuletzt anhand eines Fallbeispiels in der Praxis getestet. Darüber hinaus erfolgt eine merkmalsbasierte und metamodellbasierte Evaluierung der in dem Ansatz verwendeten Konzepte. Im Hinblick auf einen iterativen Entwicklungsprozess, die Orientierung an den Anforderungen der Anwender sowie die Notwendigkeit zur Gewinnung von Managementunterstützung wird die Erstellung eines Prototyps als wichtige Grundlage für die Akzeptanz des Ansatzes in der Praxis gesehen, da dieser einerseits die Möglichkeit einer schnelleren Überprüfung von Entwurfsentscheidungen bietet sowie andererseits die Kommunikation zwischen (Methoden-)Entwicklern, Fachbereichen und dem Management erleichtert.

Anhand der dargestellten Herausforderungen im Umfeld der Unternehmensmodellierung lässt sich die Forschungsfrage des Dissertationsvorhabens wie folgt formulieren:

Wie kann die BAI-Methode zu einem Ansatz für die Modellierung der Unternehmensarchitektur erweitert und mit Hilfe eines Metamodellierungswerkzeuges implementiert werden?

Im Einzelnen verfolgt die Arbeit somit folgende Gestaltungsziele:

- Metamodelle: Für die drei Gestaltungsebenen „Geschäftsstrategie“, „Organisation“ und „Informationssystem“ sind jeweils einzelne Metamodelle zu spezifizieren und deren Darstellungsmöglichkeiten zu beschreiben.
- Ebenenübergreifendes Metamodell: Zwischen den Metamodellen der drei Ebenen sind die Zusammenhänge klar zu definieren, so dass als Ergebnis ein konsistentes, ebenenübergreifendes Metamodell vorliegt.
- Konstruktion des Prototyps: Die Metamodelle sollen mit Hilfe eines Metamodellierungswerkzeuges prototypisch abgebildet werden. Dabei ist auf die notwendigen Anpassungen des Werkzeuges einzugehen. Darüber hinaus ist die Darstellung der einzelnen Metamodellobjekte zu spezifizieren.

²⁸ Ein Prototyp kann als eine spezielle Ausprägung eines ablauffähigen Softwaresystems, welches ausgewählte Aspekte des Zielsystems im Anwendungsbereich realisiert, definiert werden. Er dient als Kommunikationsbasis für alle beteiligten Gruppen (Benutzer, Entwickler und Wissenschaftler) und vermittelt experimentelle und praktische Erfahrungen für die Auswahl zwischen Gestaltungsalternativen. Des Weiteren dient er in der Wissenschaft vor allem dazu, Ansätze und Konzepte auf ihre Umsetzbarkeit zu testen. Vgl. *Rechenberg/Pomberger* (1999), S. 779.

- Test des Prototyps: Anhand eines Fallbeispiels wird der entwickelte Prototyp beschrieben und getestet.

Die Arbeit richtet sich an Personen, die sich in Theorie und Praxis mit der Entwicklung von Ansätzen und unterstützenden Werkzeugen für die Modellierung von Unternehmensarchitekturen auseinandersetzen. Für die Praxis können folgende primäre Nutzenpotenziale genannt werden:

- die Bereitstellung eines umfassenden Ansatzes zur Gestaltung und Abbildung der Unternehmensarchitektur sowie eine Übersicht und Bewertung bereits existierender Ansätze und aktuell verfügbarer Werkzeuge,
- die Bereitstellung eines Metamodells, das in der Praxis insbesondere als Basis einer einheitlichen Kommunikation verwendet werden kann,
- die Bereitstellung eines Softwareprototyps für den Test und die Evaluierung des entwickelten Ansatzes im eigenen Unternehmensumfeld.

Darüber hinaus bietet es Herstellern von Werkzeugen für die Unternehmensmodellierung die Möglichkeit, ihre Werkzeuge besser an die Anforderungen eines umfassenden, ebenenübergreifenden Ansatzes anzupassen.

Für Wissenschaftler, die sich mit den Themen Unternehmensmodellierung, Metamodellierung und deren Werkzeugunterstützung beschäftigen, bietet die Arbeit folgende primäre Nutzenpotenziale:

- Herleitung und Diskussion eines Ansatzes sowie eines umfassenden Metamodells zur Gestaltung der Unternehmensarchitektur,
- Beschreibung, wie anhand eines Prototyps ein Ansatz validiert werden kann,
- Verwendung des Prototyps in der Lehre, um die theoretischen Ansätze und Ideen der Unternehmensmodellierung anschaulich und praktisch vermitteln zu können.

Weitere Nutzenpotenziale bestehen für Wissenschaftler und Praxisvertreter, die andere Ansätze in dem für die Umsetzung verwendeten Werkzeug abbilden möchten.

1.3 Forschungsmethodik

Die vorliegende Arbeit ist der Wirtschaftsinformatik zuzuordnen. Nach der Wissenschaftlichen Kommission Wirtschaftsinformatik sind Informations- und Kommunikationssysteme (IKS) in Wirtschaft und Verwaltung Gegenstand der Wirtschaftsinformatik.²⁹ IKS werden dabei als soziotechnische Systeme aufgefasst, die menschliche und maschinelle Komponenten (Teilsysteme) als Aufgabenträger umfassen, die voneinander abhängig sind, ineinander grei-

²⁹ Vgl. auch im Folgenden *WKWI* (1994) und *Braun et al.* (2004).

fen und/oder zusammenwirken. Im Mittelpunkt der Untersuchungen der Wirtschaftsinformatik steht die Unterstützung bei der Erfüllung betrieblicher Aufgaben. IKS werden in Organisationen implementiert, um die Effektivität und die Effizienz der Prozesse innerhalb dieser Organisation zu verbessern.³⁰ Die Forschung muss dabei die Zusammenhänge zwischen Geschäftsstrategie, Organisationsstruktur und IKS berücksichtigen. Diese Zusammenhänge gewinnen zunehmend an Bedeutung, da die Informationstechnologie als „Enabler“ neuer Geschäftsstrategien und Organisationsstrukturen gesehen wird.³¹

Ihre Ursprünge hat die Wirtschaftsinformatik sowohl in den Wirtschaftswissenschaften als auch in der Informatik.³² Während die Wirtschaftswissenschaften primär durch empirische Forschungsansätze geprägt sind, verwendet die Informatik vor allem formalwissenschaftliche Ansätze zur Konstruktion technischer Systeme. Aufgrund dieser unterschiedlichen forschungsmethodischen Ausrichtung der Mutterdisziplinen kommt in der Wirtschaftsinformatik eine Vielzahl unterschiedlicher Forschungsmethoden zum Einsatz.³³

König et al. ermittelten im Jahr 1996 im Zuge einer Delphi-Studie³⁴, welche grundlegenden Forschungsmethoden die Wirtschaftsinformatik in den nächsten Jahren verwenden soll, um ihre Wettbewerbsposition gegenüber der Betriebswirtschaftslehre und der Informatik zu behaupten. Als wesentliche konstruktive Methoden wurden „Entwicklung und Test eines Prototyps“, „Simulation“, „Modellierung“, „Deduktion“, „Learning by Doing“ und „Kreativitätstechniken“ ermittelt.

Die vorliegende Arbeit entstand am Institut für Wirtschaftsinformatik der Universität St. Gallen (IWI-HSG) als Resultat angewandter, konstruktiver Forschung. Die Forschung am IWI-HSG ist in Form so genannter Kompetenzzentren organisiert. Zusammen mit Unternehmensvertretern werden dabei in Workshops und bilateralen Projekten Lösungsansätze zu übergeordneten Forschungsthemen erarbeitet. Die Ergebnisse der Forschungsarbeit bestehen aus Artefakten, wie z.B. Methoden und Modellen.

Das Dissertationsvorhaben baut auf bereits am IWI-HSG erarbeiteten Ergebnissen auf. Die im Kompetenzzentrum BAI³⁵ entwickelte Methode und deren Modelle bilden die Grundlage für das Dissertationsprojekt. Diese Methode wird basierend auf einem Vergleich existierender Ansätze genauer spezifiziert und erweitert sowie in Form eines Software-Prototyps abgebildet. Anhand eines Fallbeispiels wird der Prototyp anschliessend getestet. Dem Dissertationsvorhaben können somit die konstruktiven Forschungsansätze „Deduktion“, „Modellierung“ sowie „Entwicklung und Test eines Prototyps“ zugrunde gelegt werden.

³⁰ Vgl. *Hevner et al.* (2004), S. 75.

³¹ Vgl. z.B. *Kalakota/Robinson* (2001), *Orlikowski/Barley* (2001) und *Österle/Winter* (2003).

³² Vgl. *Becker et al.* (2003), S. 3.

³³ Vgl. *Becker et al.* (2003), S. 3.

³⁴ Vgl. *König et al.* (1996) und *Braun et al.* (2004).

³⁵ Vgl. *Leist/Winter* (2002).

Auch im Umfeld des „Information Systems Research (ISR)“, dem angelsächsischen Pendant der Wirtschaftsinformatik, wird der Konstruktion von Artefakten eine zunehmende Bedeutung beigemessen, wie das Design-Science-Paradigma zeigt.³⁶ Das Dissertationsvorhaben lässt sich somit auch dem Design-Science-Paradigma zuordnen, welches im Folgenden genauer dargestellt wird (vgl. Abbildung 1).³⁷

Das Design-Science-Paradigma ist wie folgt definiert: „The design-science paradigm is fundamentally a problem-solving paradigm. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished.“³⁸ Ziel des Paradigmas ist die Lösung bestimmter organisatorischer Probleme durch die Konstruktion innovativer Artefakte. Artefakte können üblicherweise Konstrukte (z.B. Begriffe und Symbole), Modelle (z.B. Abstraktionen und Abbildungen), Methoden (z.B. Algorithmen und Vorgehensweisen) oder Instanzen (z.B. implementierte oder prototypische Systeme) sein. Die Artefakte werden in strukturierter Form repräsentiert, z.B. in Form von Software, formaler Logik, mathematischen Formeln oder natürlichsprachlichen Beschreibungen.

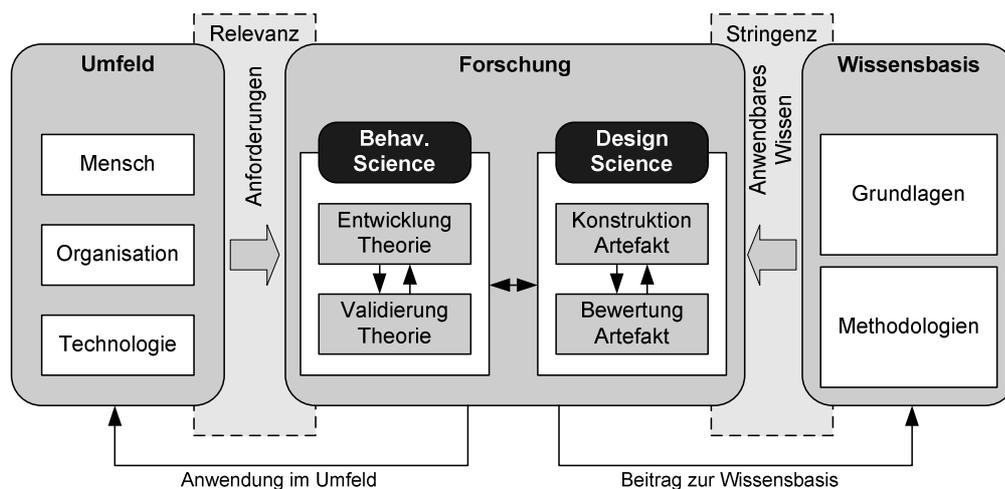


Abbildung 1: Information Systems Research Framework³⁹

Im Gegensatz zum Design-Science-Paradigma steht beim Behavioral-Science-Paradigma die Entwicklung und Validierung von Theorien im Mittelpunkt. In der Regel werden bereits existierende Artefakte betrachtet, die in einem bestimmten organisatorischen Umfeld eingesetzt werden. Theorien versuchen bestimmte Phänomene zu erklären, die in Verbindung mit dem

³⁶ Vgl. hierzu *Hevner et al.* (2004) und die im Februar 2006 durchgeführte Konferenz zum Thema „Design Science Research in Information Systems and Technology“ (<http://ncl.cgu.edu/designconference/>).

³⁷ Vgl. auch im Folgenden *Hevner et al.* (2004).

³⁸ Vgl. *Hevner et al.* (2004), S. 76.

³⁹ Vgl. *Wortmann* (2005), S. 6 und *Hevner et al.* (2004), S. 80.

Einsatz eines bestimmten Artefakts auftreten.⁴⁰ Das von HEVNER ET AL. vorgeschlagene „Information Systems Research Framework“ soll dazu dienen, die Informationssystemforschung zu verstehen, durchzuführen und zu evaluieren, indem das Design-Science- und Behavioral-Science-Paradigma miteinander kombiniert werden. Es besteht aus den folgenden Elementen:⁴¹

- **Umfeld:** Das Umfeld legt den Gegenstandsbereich der Forschung fest. Es besteht aus den Dimensionen Mensch, Organisation und Technologie.
- **Anforderungen:** Das Umfeld beschreibt Ziele, Aufgaben, Probleme und Möglichkeiten und definiert damit Anforderungen an die Forschung aus Sicht der betroffenen Menschen innerhalb der Organisation. Durch die Ausrichtung der Forschung an den Anforderungen des Umfeldes wird die Relevanz der Forschung garantiert.
- **Forschung:** Die Informationssystemforschung wird mit Hilfe von zwei sich ergänzenden Ansätzen durchgeführt. Die konstruktionsorientierte Forschung (Design Science) konstruiert und beurteilt Artefakte, die die aus dem Umfeld abgeleiteten Anforderungen sowie Ergebnisse der behavioristischen Forschung umsetzen. Die behavioristische Forschung untersucht dagegen die Phänomene, die durch den Einsatz von zuvor konstruierten Artefakten entstehen, und entwickelt und validiert daraus entsprechende Theorien.
- **Wissensbasis:** Die Wissensbasis stellt dem Forscher Grundlagen und Methodologien zur Verfügung, die von früheren Arbeiten der Informationssystemforschung oder von Arbeiten aus Referenzdisziplinen stammen. Die Grundlagen bestehen beispielsweise aus Theorien, Bezugsrahmen, Methoden und Modellen, die für die Konstruktion von Artefakten verwendet werden können. Methodologien geben Richtlinien für die Validierung von Theorien und Artefakten vor.
- **Anwendbares Wissen:** In einem konkreten Forschungsprozess verwendet der Forscher Teile der Wissensbasis. Durch die konsequente, kontextbezogene Anwendung von Teilen der Wissensbasis wird die Stringenz des Forschungsprozesses gewährleistet.

Die Ergebnisse der konstruktionsorientierten sowie der behavioristischen Forschungsarbeit können anhand der Erfüllung der Anforderungen eines bestimmten Umfeldes bzw. anhand des Beitrags zur Wissensbasis beurteilt werden.

Hevner et al. haben basierend auf ihrem Bezugsrahmen sieben Richtlinien aufgestellt, an denen sich der Forscher bei einem bestimmten Forschungsprojekt orientieren kann, um erfolgreich und effektiv Design-Science-Forschung zu betreiben. Für eine vollständige Design-Science-Forschungsarbeit sollte jede der sieben Richtlinien zumindest ansatzweise adressiert werden. Tabelle 1 zeigt, wie die vorliegende Arbeit die einzelnen Richtlinien adressiert.

⁴⁰ Vgl. Hevner et al. (2004), S. 77.

⁴¹ Vgl. Hevner et al. (2004), S. 79-80.

Richtlinien für die Erfüllung des Design-Science-Paradigma		
Richtlinie	Beschreibung	Adressierung in der vorliegenden Arbeit
1. Konstruktion Artefakt	Die DS-Forschung verlangt die Entwicklung eines anwendbaren Artefakts in Form einer Methode, eines Modells oder einer Instanz.	Die konstruierten Artefakte sind der entwickelte Ansatz zur Modellierung der Unternehmensarchitektur sowie der implementierte Softwareprototyp.
2. Problemlösung und Relevanz	Das Ziel der DS-Forschung ist die Entwicklung technologiebasierter Lösungen für wichtige und relevante Probleme.	Die Relevanz ergibt sich aus der Bedeutung (Ziele und treibende Kräfte) der Unternehmensarchitektur sowie den Defiziten bestehender Ansätze und Werkzeuge.
3. Bewertung des Artefakts	Die Anwendung, Effektivität und Qualität des Artefakts müssen stringent und nachvollziehbar dargelegt werden.	Das Metamodell wird in einem Prototyp implementiert und anhand eines Fallbeispiels getestet. Darüber hinaus erfolgt eine metamodellbasierte sowie natürlichsprachliche und merkmalsbasierte Evaluation.
4. Forschungsbeitrag	Die DS-Forschung muss klar nachvollziehbare Ergebnisse liefern.	Der Forschungsbeitrag ist sichergestellt, wenn die Defizite bestehender Ansätze und Werkzeuge adressiert werden konnten.
5. Stringenz	Die DS-Forschung verlangt bei der Konstruktion und Bewertung eines Artefakts die Einhaltung forschungsmethodischer Grundsätze.	Es werden die konstruktiven Methoden Deduktion, Modellierung sowie Entwicklung und Test eines Prototyps angewendet.
6. Suchprozess	Die DS-Forschung ist ein Suchprozess. Die endgültige Problemlösung wird durch einen iterativen Prozess erreicht.	Die Ergebnisse des Tests und der Evaluation sowie das Feedback bei den Workshops fließen in einem iterativen Prozess in die Verbesserung des Prototyps und Metamodells ein.
7. Kommunikation der Ergebnisse	Die Ergebnisse müssen sowohl einem technologieorientierten als auch einem managementorientierten Publikum vermittelt werden.	Die Ergebnisse werden in der Forschungsgemeinschaft veröffentlicht sowie vor Vertretern der Partnerunternehmen präsentiert.

Tabelle 1: Richtlinien für die Erfüllung des Design-Science-Paradigma⁴²

1.4 Aufbau der Arbeit

In der Einführung wurden auf der Basis der Problemstellung und des identifizierten Handlungsbedarfs die Zielsetzungen abgeleitet, die Adressaten festgelegt und die zugrunde liegende Forschungsmethodik erläutert. Nach der Einführung befasst sich das folgende Kapitel zwei mit den konzeptionellen Grundlagen der Dissertation. Hierzu werden zunächst relevante Begrifflichkeiten aus den Bereichen Methoden, Modelle und Unternehmensarchitektur erläutert. Anschliessend folgt eine Beschreibung der wesentlichen Ziele und treibenden Kräfte der Unternehmensarchitektur. Darauf aufbauend werden Anforderungen an den zu entwickelnden Ansatz sowie die Werkzeugunterstützung formuliert. Im darauf folgenden Kapitel drei werden die für das Dissertationsvorhaben relevanten Entwicklungen dargestellt. Zum einen wird ein Überblick über Methoden zur Unternehmensmodellierung, die primär aus der Wissenschaft stammen, gegeben. Zum anderen erfolgt eine Darstellung von Bezugsrahmen (Frameworks), die in der Praxis am häufigsten zum Einsatz kommen und die von den meisten aktuell am Markt verfügbaren Werkzeugen unterstützt werden. Im vierten Kapitel wird der eigene An-

⁴² In Anlehnung an Hevner *et al.* (2004), S. 83.

satz zur Modellierung der Unternehmensarchitektur beschrieben. Dabei liegt der Fokus vor allem auf der Definition eines Metamodells mit den wesentlichen Schlüsselementen einer Unternehmensarchitektur sowie deren Zusammenhänge. Die Beschreibung der Abbildung dieses Metamodells mit Hilfe einer kommerziellen Metamodellierungsplattform folgt im Kapitel fünf. Der daraus entstandene Software-Prototyp wird im Hinblick auf dessen wesentliche Komponenten und Funktionalitäten vorgestellt. Zum Ende des Hauptteils wird eine multiperspektivische Evaluation des in dieser Arbeit vorgeschlagenen Ansatzes in Form einer Fallstudie sowie einer natürlichsprachlichen, merkmalsbasierten und metamodellbasierten Evaluation durchgeführt. Abschliessend erfolgen eine Zusammenfassung der wesentlichen Ergebnisse dieser Arbeit und deren kritische Würdigung sowie ein Ausblick auf zukünftige Forschungs- und Entwicklungsmöglichkeiten. Die folgende Abbildung 2 visualisiert den Aufbau der Arbeit.

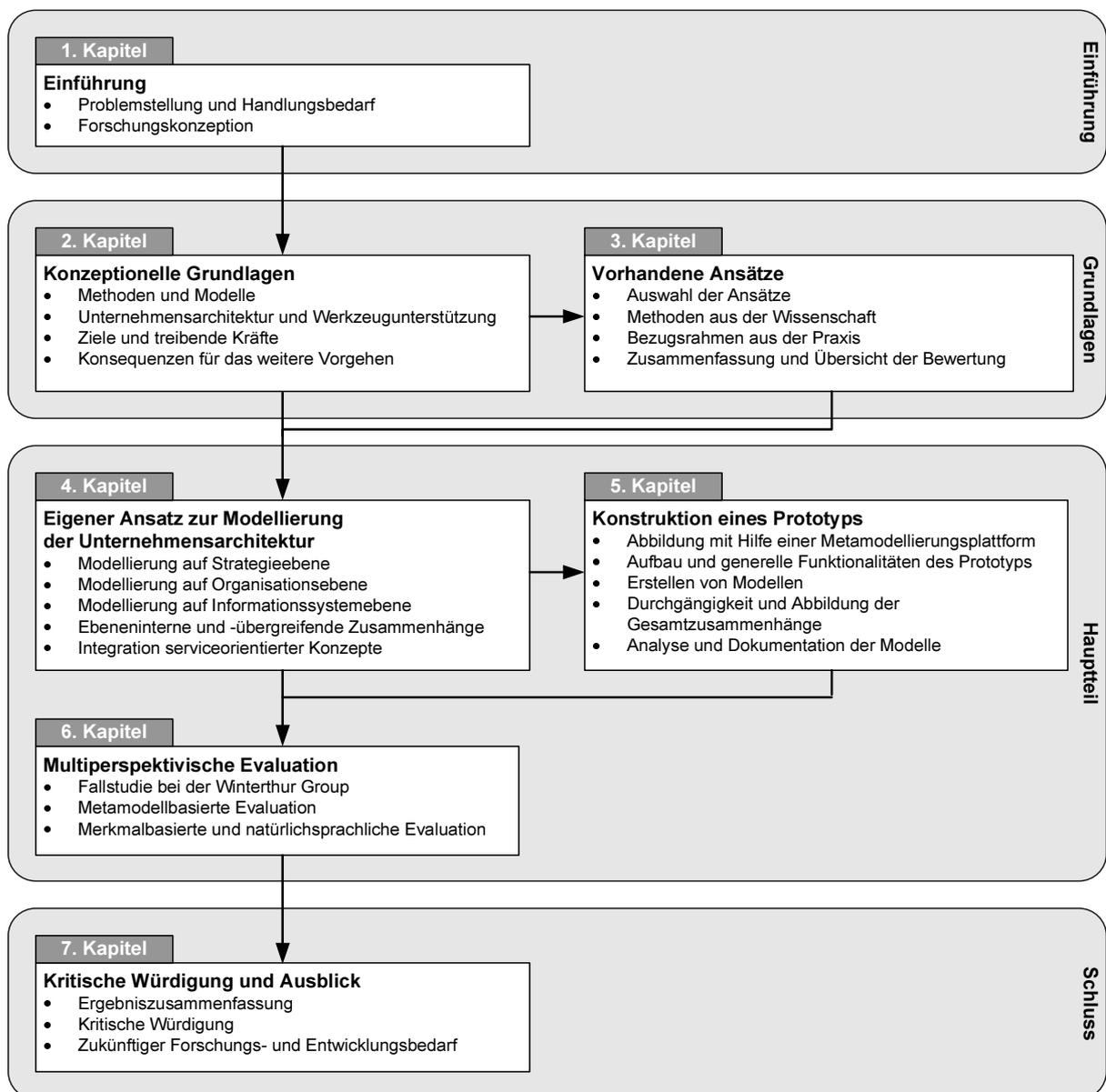


Abbildung 2: Aufbau der Arbeit

2 Konzeptionelle Grundlagen der Unternehmensmodellierung

In diesem Kapitel werden die begrifflichen Grundlagen der vorliegenden Arbeit gelegt. Für die relevanten Themenbereiche Methoden, Modelle, Unternehmensarchitektur und Werkzeugunterstützung werden die in den späteren Kapiteln verwendeten Begriffe erläutert. Anschliessend folgt eine Beschreibung der wesentlichen Ziele und treibenden Kräfte der Unternehmensarchitektur. Darauf aufbauend werden zuletzt Anforderungen an den zu entwickelnden Ansatz und die Werkzeugunterstützung abgeleitet.

2.1 Methoden

Das „Information Systems Research (ISR)“ (angelsächsisches Pendant der Wirtschaftsinformatik) wurde bisher vor allem durch einen behavioristischen Forschungsansatz dominiert. Nach HEVNER wird in Zukunft aber auch das „Design Science (DS)“ als komplementärer Forschungsansatz zunehmend an Bedeutung gewinnen.⁴³ Während vor allem der Einsatz von Informationssystemen als auch Konstrukte und Modelle sehr ausführlich und umfangreich in der IS-Forschung diskutiert wurden, wurde die Konstruktion von Methoden nur selten adressiert.

Grundsätzlich sollten alle wissenschaftlichen Methoden eine bestimmte Forschungsfrage, die Validität der Ergebnisse und die aktuelle Diskurswelt reflektieren.⁴⁴ In Bezug auf den DS-Ansatz wurden zahlreiche epistemologische Aspekte diskutiert.⁴⁵ Nach GREIFFENBERG können Methoden als Theorien der DS-Forschung betrachtet werden, sofern sie entsprechend konstruiert und validiert wurden.⁴⁶

In den folgenden beiden Abschnitten werden gängige Methodenverständnisse zusammengefasst und deren übergreifende Charakteristika sowie Elemente identifiziert. Diese werden später zur Beschreibung und Beurteilung bereits existierender Methoden sowie für die Konstruktion des eigenen Methoden-Ansatzes verwendet.

2.1.1 Method Engineering

Die Gestaltung von Organisationsstrukturen erfordert, ebenso wie die Entwicklung von Softwaresystemen, ein ingenieurmässiges Vorgehen, um systematisch geplant und nachvollzogen werden zu können.⁴⁷ Sie verlangt somit eine Konstruktionssystematik. Die Anwendung von

⁴³ Vgl. *Hevner et al.* (2004) sowie Abschnitt 1.3.

⁴⁴ Vgl. *Lührs* (2001) und *Yin* (2002).

⁴⁵ Vgl. z.B. *Ferstl/Sinz* (1998), *Greiffenberg* (2003) und *Schütte et al.* (1999).

⁴⁶ Vgl. *Greiffenberg* (2003).

⁴⁷ Vgl. *Braun et al.* (2005), S. 1297.

Methoden bildet den Ausgangspunkt ingenieurmässigen Vorgehens.⁴⁸ Wer methodisch vorgeht, kann die Mittel und Vorgehensweisen, die er zur Erreichung bestimmter Ziele gewählt hat, begründen.⁴⁹ Es herrscht weitestgehend Einigkeit darüber, dass der Einsatz von Methoden die Grundlage für ein ingenieurmässiges Vorgehen darstellt.⁵⁰ Dennoch existieren für den Methodenbegriff eine Vielzahl unterschiedlicher Definitionen und Interpretationen. Führt man die unterschiedlichen Verständnisse in ihrem Kern zusammen, kann eine Methode anhand der folgenden wesentlichen Eigenschaften charakterisiert werden:

- Zielorientierung/Problemlösung als Ziel: Methoden sind stets zielorientiert. Sie geben Vorschriften des Vorgehens bzw. Handelns, um festgelegte Ziele zu erreichen oder Probleme zu lösen.⁵¹
- Systematik/Planmässigkeit: Wenn Methoden Handlungsvorschriften bzw. Anleitungen zur Lösung von Problemen oder zur Erreichung von Zielen liefern sollen, dann müssen sie eine systematische Struktur aufweisen, damit sich konkrete Arbeitsschritte bzw. Aufgaben zur Erreichung der Ziele ableiten lassen.⁵²
- Prinzipienorientierung: Viele Definitionen verweisen auf die Relation der Methode zu Prinzipien, wie grundsätzliche Gestaltungskriterien und/oder Strategien (z.B. Top-Down- oder Bottom-Up-Ansatz).⁵³
- Nachvollziehbarkeit: Von einigen Autoren wird die intersubjektive Nachvollziehbarkeit von Methoden gefordert.⁵⁴

BRINKKEMPER betrachtet die Entwicklung von Informationssystemen als ein Phänomen, das starke Parallelen zu den Ingenieurwissenschaften aufweist.⁵⁵ Er schlägt deshalb einen Forschungsrahmen für Methoden und Werkzeuge für die Entwicklung von Informationssystemen vor, genannt Methoden-Engineering (ME). Er definiert ME als eine Ingenieurwissenschaft, die sich mit dem Entwurf, der Konstruktion und Anpassung von Methoden, Techniken und Werkzeugen für die Entwicklung von Informationssystemen beschäftigt.⁵⁶ Seine Arbeit konzentriert sich überwiegend auf situationsbezogene Methoden, die er als Methoden definiert, die für ein bestimmtes Projekt bzw. ein bestimmtes unternehmensindividuelles Umfeld konfiguriert sind. Die Konstruktion einer situationsbezogenen Methode erfordert standardisierte

⁴⁸ Vgl. Greiffenberg (2003), S. 955.

⁴⁹ Vgl. Lorenz (1995).

⁵⁰ Vgl. Braun et al. (2005), S. 1297.

⁵¹ Vgl. Balzert (2000), Becker et al. (2001), Hesse et al. (1992), Stahlknecht/Hasenkamp (1999), Teubner (1999) und Zelewski (1999).

⁵² Vgl. Becker et al. (2001), Greiffenberg (2003), Gutzwiller (1994), Heym (1993), Stahlknecht/Hasenkamp (1999), Zelewski (1999).

⁵³ Vgl. Becker et al. (2001), Stahlknecht/Hasenkamp (1999) und Teubner (1999).

⁵⁴ Vgl. Balzert (2000), Hesse et al. (1992) und Zelewski (1999).

⁵⁵ Vgl. Brinkkemper (1996), S. 276.

⁵⁶ Vgl. Brinkkemper (1996), S. 276.

Bausteine sowie Richtlinien, um die Bausteine richtig zusammensetzen zu können. Diese Richtlinien werden durch so genannte Metamethoden definiert.⁵⁷ Zudem sollte ein Konfigurationsprozess festgelegt werden, der die Zusammensetzung der einzelnen Methodenbausteine zu einer situationsbezogenen Methode anleitet. Da alle Projekte unterschiedlich sind, können sie nicht durch eine Standardmethode unterstützt werden.⁵⁸ Es stellt sich somit die Frage, wie den Entwicklern von Informationssystemen eine adäquate methodische Vorgehensweise und eine entsprechende Werkzeugunterstützung zur Verfügung gestellt werden können.

2.1.2 Methodenbestandteile

Das Methoden-Engineering wendet die Prinzipien von Methoden, bzw. den ingenieurmässigen Entwurf von Informationssystemen, auf den Entwurf von Methoden an.⁵⁹ Es schafft somit eine zusätzliche Abstraktionsebene und bildet die Grundlage für die Entwicklung und Beschreibung von Methoden. GUTZWILLER und HEYM haben zahlreiche Ansätze des Methoden-Engineering analysiert und daraus allgemeingültige Bestandteile der Methodenbeschreibung sowie deren Beziehungen zueinander abgeleitet (vgl. Abbildung 3).⁶⁰ Eine Methode umfasst demnach die Elemente „Aktivität“, „Rolle“, „Ergebnisdokument“, „Metamodell“ und „Technik“.

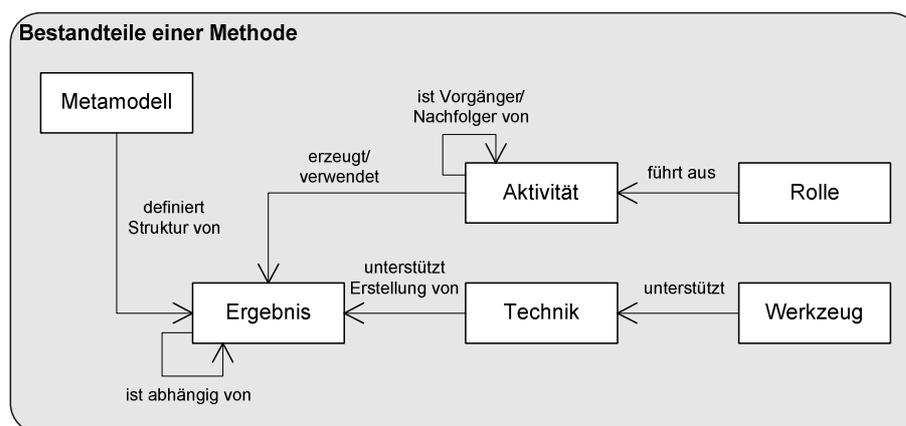


Abbildung 3: Bestandteile einer Methode⁶¹

Durch eine Methode wird ein Vorgehen in Form von Aktivitäten festgelegt. Die Spezifikation des Ablaufs dieser Aktivitäten hinsichtlich ihrer Reihenfolge und zulässiger Überlappungen wird als Vorgehensmodell bezeichnet. Aktivitäten erzeugen und/oder verwenden ein oder mehrere Ergebnisse. Ergebnisse werden in zuvor durch Metamodelle spezifizierten Ergebnisdokumenten hinterlegt. Die Aktivitäten werden bestimmten Rollen (z.B. Personen, Stellen

⁵⁷ Vgl. Brinkkemper (1996), S. 277.

⁵⁸ Vgl. Brinkkemper (1996), S. 278.

⁵⁹ Vgl. Brenner (1995), S. 11.

⁶⁰ Vgl. Gutzwiller (1994) und Heym (1993).

⁶¹ Vgl. Gutzwiller (1994), S. 13.

oder Organisationseinheiten) zugeordnet. Die jeweilige Vorschrift zur Erstellung (und damit zur Dokumentation) der Ergebnisse wird als (Modellierungs-)Technik bezeichnet. Im Gegensatz zum Vorgehensmodell, das das Vorgehen im Grossen definiert, beschreiben Techniken das Vorgehen im Kleinen.⁶² Aktivitäten spezifizieren, wann welche Ergebnisse produziert werden.⁶³ Techniken liefern dagegen die detaillierten Handlungsanweisungen zur Erstellung der Ergebnisse.⁶⁴ Software-Werkzeuge können verwendet werden, um die Anwendung einer Technik zu unterstützen. Die in den verschiedenen Ergebnisdokumenten repräsentierten Ergebnisse und deren Zusammenhänge können im Metamodell der Methode abgebildet werden. Eine Methode lässt sich somit durch ihr Vorgehensmodell, das Metamodell (d.h. die Menge aller Ergebnisspezifikationen), das Rollenmodell sowie die verwendeten Techniken beschreiben.

2.2 Modelle

Die Abbildung natürlicher oder künstlicher Originale in Form von Modellen ist sowohl in der Wissenschaft als auch in der Praxis weit verbreitet. Auch bei der Gestaltung bzw. Modellierung von Informationssystemen und Unternehmensstrukturen werden die Ergebnisse in Form von Modellen repräsentiert. Sie sind somit für das Dissertationsvorhaben von Interesse und werden daher in den folgenden Abschnitten näher erläutert. Dabei wird insbesondere auf Begriffe aus den Bereichen der Modellierung, Metamodellierung und Referenzmodellierung eingegangen.

2.2.1 Modell und Modellierung

In der Literatur finden sich zahlreiche Definitionen zum Modellbegriff.⁶⁵ Ein domänenübergreifender und von breiten Kreisen der Forschung⁶⁶ akzeptierter Modellbegriff wird von STACHOWIAK vorgeschlagen. Nach STACHOWIAK besitzt ein Modell drei wesentliche Merkmale:⁶⁷

- **Abbildung:** Ein Modell ist ein Abbild oder eine Repräsentation eines realen oder gedachten Sachverhaltes.
- **Verkürzung:** Ein Modell erfasst nicht alle Attribute des entsprechenden Originals.

⁶² Vgl. *Gutzwiller* (1994), S. 14.

⁶³ Vgl. *Gutzwiller* (1994), S. 14.

⁶⁴ Vgl. *Gutzwiller* (1994), S. 14 und *Winter* (2003d), S. 88.

⁶⁵ Vgl. z.B. *Alpar et al.* (2005), S. 20, *Ferstl/Sinz* (1998), S. 18, *Schütte* (1998), S. 45, *Teubner* (1999), S. 14, *Winter* (2003d), S.89.

⁶⁶ Vgl. z.B. *Lehner* (1995), S. 27, *Schelp* (2000), S. 15-17, *Schütte* (1998), S. 41-45 und *Teubner* (1999), S. 14.

⁶⁷ Vgl. *Stachowiak* (1973), S. 131f.

- Pragmatismus: Ein Modell sollte sich am Nützlichen orientieren. Es wird innerhalb einer bestimmten Zeitspanne und zu einem bestimmten Zweck für ein Original eingesetzt.

Die Modellierungstechnik schreibt vor, wie ein Modell zu konstruieren ist. Sie besteht aus einer Modellierungssprache und einer Vorgehensweise (vgl. Abbildung 4). Von der Vorgehensweise (Vorgehen im Kleinen⁶⁸) als Vorschrift zur Erstellung eines Modells ist die Sprache zu unterscheiden, mit der das Modell beschrieben wird.⁶⁹ Die Sprache legt die Elemente fest, die bei der Erstellung eines Modells verwendet werden können. Sie wird durch ihre Syntax, Semantik und Notation beschrieben. Die Notation beschreibt die Darstellung der Elemente der Modellierungssprache (Symbole) und visualisiert somit ihre Syntax. Die Syntax definiert die Grammatik der Elemente und die Semantik deren Bedeutung. Die Darstellung der Elemente bzw. Symbole sollte so gewählt sein, dass sie die Bedeutung des Elements (Semantik) der Sprache möglichst intuitiv wiedergeben. Bestimmte Sprachen oder Notationen sind in der Regel an bestimmte Vorgehensweisen gebunden und umgekehrt. Die Vorgehensweise definiert die Schritte, die durchgeführt werden müssen, um mit der Sprache bestimmte Modelle zu erstellen.

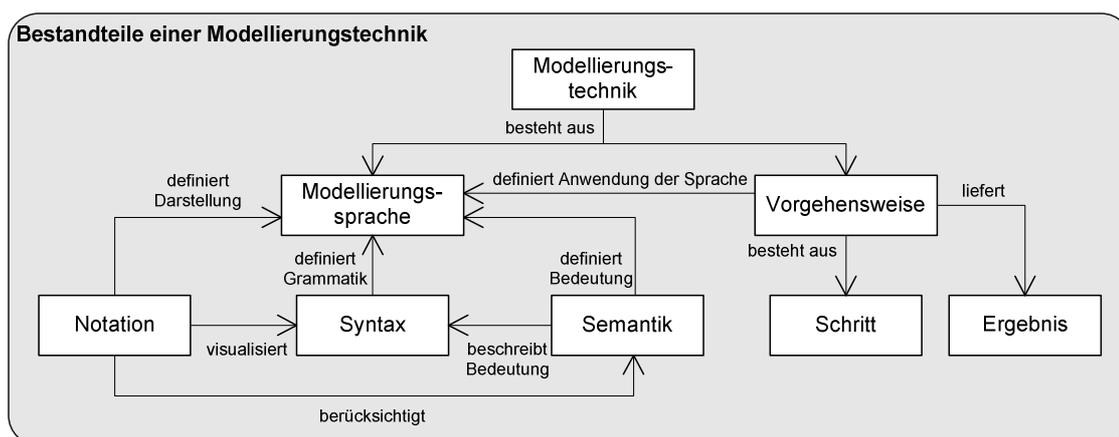


Abbildung 4: Bestandteile einer Modellierungstechnik⁷⁰

Modelle entstehen als Ergebnisse der Modellierung mit einer bestimmten Technik.⁷¹ Für das Dissertationsvorhaben wird die folgende Begriffsdefinition verwendet: „Als Modellierung wird der Vorgang der Konstruktion eines Abbilds realer oder gedachter Sachverhalte verstanden, welcher auf der Grundlage der Wahrnehmung dieser Sachverhalte durch den/die Modellierer/in erfolgt und durch den jeweiligen Modellierungszweck beeinflusst wird.“⁷² Nach

⁶⁸ Im Unterschied zur Vorgehensweise der Methode (vgl. Abschnitt 2.1.2).

⁶⁹ Vgl. Winter (2003d), S. 89.

⁷⁰ Vgl. Karagiannis/Kühn (2002), S. 184.

⁷¹ Vgl. Schütte (1998), S. 40.

⁷² Vgl. Winter (2003d), S. 89.

LEIST lassen sich beispielsweise die unterschiedlichen in der Literatur genannten Zwecke der Modellierung zu folgenden Gruppen zusammenfassen:⁷³

- **Schulungszweck:** Der Schulungszweck beinhaltet sowohl das Erlernen der Modellierungstechnik als auch das einfache Vermitteln organisatorischer Abläufe oder Funktionen eines Informationssystems an dessen Nutzer.
- **Kommunikationsbasis:** Ergebnisdokumente einer Modellierungstechnik können innerhalb/ausserhalb eines Unternehmens zur Kommunikation unterschiedlichster Sachverhalte dienen. Zudem dienen sie der Kommunikation zwischen verschiedenen an einem Projekt beteiligten Rollen, indem relevante Objekte, ihre Beziehungen sowie ihre Attribute festgelegt werden.
- **Analysezweck:** Ergebnisse der Modellierung können zum Analysieren von Sachverhalten herangezogen werden.
- **Gestaltungs- und Entwicklungszweck:** Ergebnisse der Modellierung können als Grundlage für die Entwicklung und Gestaltung beispielsweise von Informationssystemen oder Organisationsstrukturen dienen.

Trotz unterschiedlicher Wahrnehmungen und Modellierungszwecke können Modelle anhand von zwei wesentlichen Dimensionen strukturiert werden:⁷⁴

- **Generalisierung vs. Spezialisierung:** Spezielle Modelle bilden Sachverhalte in einem bestimmten Anwendungskontext ab, beispielsweise in einer bestimmten Branche oder in einem bestimmten Unternehmen. Generalisierte Modelle abstrahieren dagegen von einem bestimmten Anwendungskontext (*Generalisierungsgrad*).
- **Aggregation vs. Dekomposition:** Architekturmodelle, wie beispielsweise Prozessarchitekturmodelle (Prozesslandkarten), bilden Zusammenhänge zwischen Komponenten im Groben ab. Ein Detailmodell (z.B. Ablaufplanungsmodell eines bestimmten Prozesses) bildet dagegen eine ganz bestimmte Komponente eines Architekturmodells ab (*Aggregationsgrad*).

In der Realität existieren eine Vielzahl unterschiedlicher Originale bzw. Sachverhalte, die in Modellen abgebildet werden können. Für das Dissertationsvorhaben sind insbesondere Modelle von Unternehmen von Interesse. Grundlage der Modellierung ist somit die betriebliche Diskurswelt.

Innerhalb der betrieblichen Diskurswelt stellt der Grad der Abstraktion eine weitere Dimension zur Strukturierung der unterschiedlichen Modelle dar.⁷⁵ Eine Organisation bzw. ein Unter-

⁷³ Vgl. Leist (2002), S. 8-9.

⁷⁴ Vgl. Winter (2003d), S. 90.

⁷⁵ Vgl. z.B. Anaya/Ortiz (2005), S. 26.

nehmen kann mit unterschiedlichen Abstraktionsgraden betrachtet bzw. in unterschiedliche Abstraktionsebenen⁷⁶ eingeteilt werden. Häufig wird davon ausgegangen, dass Mitarbeiter aus den Fachabteilungen einen abstrakteren, gesamthafteren Blick auf das Unternehmen sowie dessen Strukturen und deren Strategie haben als Mitarbeiter aus dem IT-Bereich aufgrund der Informationen, die ihnen zur Verfügung stehen und mit denen sie arbeiten. Ein Geschäftsprozessverantwortlicher betrachtet das Unternehmen beispielsweise als eine Menge von Prozessen, die bestimmte Leistungen erzeugen und verwenden. Ein IT-Verantwortlicher betrachtet dagegen das Unternehmen als eine Menge von Informationssystemen, die bestimmte Funktionalitäten zur Verfügung stellen. Innerhalb jeder dieser Abstraktionsebenen gibt es Modelle mit unterschiedlichen Generalisierungs- und Aggregationsgraden (vgl. Abbildung 5).

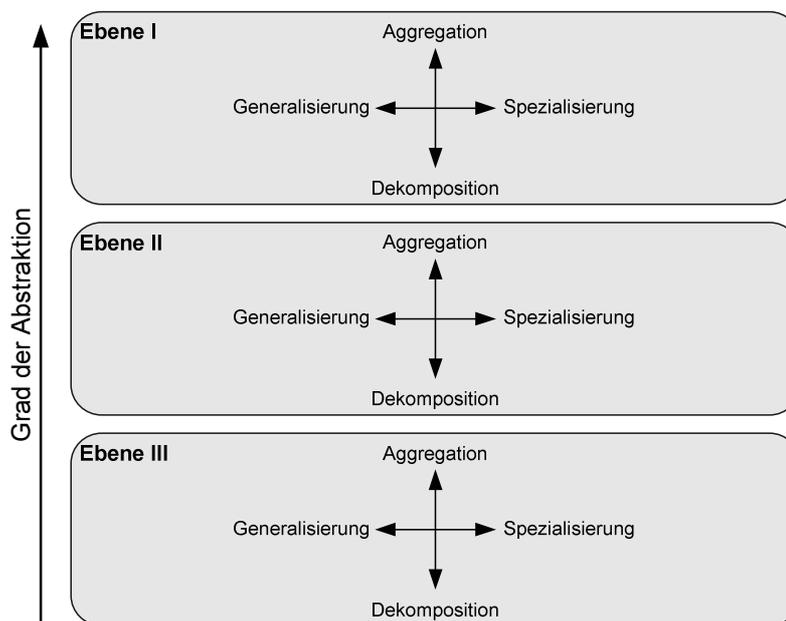


Abbildung 5: Strukturierung von Modellen

2.2.2 Referenzmodell und Referenzmodellierung

Die Modellierung kann durch Referenzmodelle unterstützt werden. Der Begriff Referenz kennzeichnet üblicherweise einen Verweis oder einen Bezug, dem eine Empfehlung zugesprochen wird, so dass die Referenz den Best Practice oder Common Practice markiert.⁷⁷ Referenzmodelle entstehen als Ergebnisse der Referenzmodellierung, welche häufig auch als Referenzmodellkonstruktion bezeichnet wird.⁷⁸ Referenzmodellierung kann beschrieben werden als die Menge aller Handlungen, welche die Konstruktion und Anwendung wieder ver-

⁷⁶ Die Begriffe Abstraktions-, Betrachtungs-, Modellierungs-, Architektur- und Gestaltungsebene werden in dieser Arbeit synonym verwendet. Oftmals wird auch aus Gründen der Einfachheit lediglich von einer „Ebene“ gesprochen.

⁷⁷ Vgl. vom Brocke (2003a), S. 31.

⁷⁸ Vgl. z.B. Becker/Delfmann (2004), S. 1.

wendbarer Modelle (Referenzmodelle) beabsichtigen.⁷⁹ Analog zur Definition eines Modells wird ein Referenzmodell als Ergebnis einer Konstruktion beschrieben, das die Repräsentation eines Originals als Empfehlung deklariert.⁸⁰ Ein Referenzmodell gilt somit als Empfehlung für einen bestimmten Anwendungskontext. Referenzmodelle repräsentieren Abbildungen von allgemein gültigen Strukturen (z.B. Prozessen und Daten), die für eine Gruppe von Unternehmen (innerhalb eines Sektors oder einer Branche) oder für eine betriebswirtschaftliche Funktion Gültigkeit haben sollen.⁸¹ Durch die Verwendung von Referenzmodellen bei der Konstruktion von unternehmensspezifischen Modellen können Kosteneinsparungen erzielt werden, da das in den Referenzmodellen enthaltene Know-how wieder verwendet werden kann.⁸² Sie beinhalten ausserdem in vielen Fällen den aktuellen Best Practice bzw. Common Practice für einen bestimmten betriebswirtschaftlichen Anwendungskontext, so dass der Anwender zwar nicht von der besten, aber zumindest von einer guten Lösung ausgehen kann.⁸³

Der Grundgedanke der Referenzmodellierung ist demnach – analog zur Konstruktion von Methoden bzw. zum Methoden-Engineering (vgl. Abschnitt 2.1.1) – die Konstruktion unternehmensspezifischer Modelle auf Basis vorgefertigter Modelle bzw. Modellbausteine. Das Referenzmodell stellt dabei ein Modell für einen bestimmten Typ von Unternehmen dar (z.B. Unternehmen der Versicherungsbranche), das für den Einsatz in einem bestimmten unternehmensindividuellen Umfeld (Versicherung XY), in einem bestimmten Anwendungsbereich (z.B. Schadenbearbeitung), angepasst und wieder verwendet werden kann (vgl. Abbildung 6). Dadurch soll eine Steigerung der Effektivität und Effizienz von Modellierungsprozessen erreicht werden.

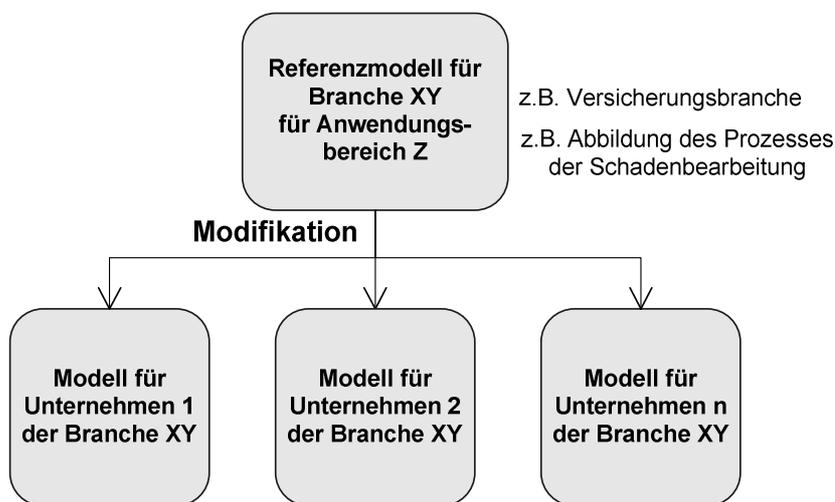


Abbildung 6: Referenzmodellierung

⁷⁹ Vgl. *Fettke/Loos* (2004b), S. 331.

⁸⁰ Vgl. *Schütte* (1998), S. 69.

⁸¹ Vgl. *Becker* (2004a), S. 325.

⁸² Vgl. *Becker* (2004b), S. 325.

⁸³ Im Bereich der Referenzmodellierung kann zwischen Common-Practice- und Best-Practice-Referenzmodellen unterschieden werden. Vgl. hierzu *Becker et al.* (2002a).

Sowohl die Referenzmodellierung als auch die in Abschnitt 2.1 beschriebene Methodenentwicklung können in die zwei Vorgänge Konstruktion und Anwendung zerlegt werden.⁸⁴ Das Ziel der Konstruktion ist die Entwicklung eines Referenzmodells bzw. einer Methode, das/die für eine Klasse von Unternehmen (z.B. Unternehmen einer bestimmten Branche) Gültigkeit hat und das/die in unterschiedlichen Situationen eingesetzt werden kann. Die Anwendung des konstruierten Referenzmodells bzw. der konstruierten Methode erfolgt im Anschluss an den Konstruktionsprozess. Dazu wird das Referenzmodell bzw. die Methode für den Einsatz im jeweiligen unternehmensspezifischen Umfeld durch den Anwender manuell modifiziert.

2.2.3 Metamodell und Metamodellierung

Bei der Modellbildung wird aus einem Problembereich, dem Original, durch Abbildung und Verkürzung mit einer gegebenen Pragmatik ein Modell für einen bestimmten Zweck gebildet. Wenn Modelle und Modellbildung selbst zum Gegenstand der Modellierung werden, spricht man von Metamodellierung.⁸⁵ Metamodelle definieren Konstruktionsregeln für die Erstellung von Modellsystemen, indem sie die verfügbaren Arten von Bausteinen (Metaobjekte, Objekttypen, Klassen, Metaentitätstypen), die Beziehungen zwischen den Bausteinen (Meta-Beziehungen, Beziehungstypen) sowie Konsistenzbedingungen für die Verwendung von Bausteinen und Beziehungen spezifizieren.⁸⁶

Metamodelle sind ein wesentlicher Bestandteil einer Methode (vgl. Abbildung 3). Sie definieren die Ergebnisse (Instanzen bzw. Objekte und Beziehungen), welche bei der Anwendung einer bestimmten Technik erzeugt werden. Die erzeugten Ergebnisse werden meist in Form von Modellen dargestellt. Ein Metamodell bezeichnet somit ein Modell, welches eine Menge gleichartiger, mit derselben Technik erstellter Modelle repräsentiert (Typ). Da in den verschiedenen Ergebnissen bzw. Modellen Informationen dokumentiert werden, wird häufig auch von einem Informationsmodell einer Methode anstelle von einem Metamodell gesprochen.

Das in Abbildung 3 dargestellte Modell ist ebenfalls ein Metamodell. Es handelt sich dabei um das Metamodell des Methoden-Engineering⁸⁷, da es die wesentlichen konstituierenden Elemente einer Methode definiert. Es stellt somit den „Bauplan“ für Methoden bereit. Ausgehend von der Aktivität wird die Methode mit ihren Elementen „Rolle“, „Ergebnis“, „Technik“, „Werkzeug“ und „Metamodell“ aufgebaut.⁸⁸ Durch das Metamodell wird sichergestellt, dass Methoden mit den gleichen Komponenten, wenn auch unterschiedlicher Ausprägung,

⁸⁴ Vgl. im Folgenden *Fettke/Loos* (2002), S. 10 und *Wortmann* (2005), S. 100.

⁸⁵ Vgl. *Strahringer* (1998), S. 1.

⁸⁶ Vgl. *Sinz* (1999b), S. 1035.

⁸⁷ Vgl. *Heym* (1993).

⁸⁸ Vgl. *Gutzwiller* (1994), S.11-17, *Winter* (2003d), S. 88.

entwickelt und so einerseits vergleichbar und andererseits nachvollziehbar werden. Darüber hinaus wird sichergestellt, dass keine Komponente, wie z.B. das Ergebnis bzw. das Ergebnisdokument, unberücksichtigt bleibt.⁸⁹

Modelle werden mit Hilfe von Modellierungstechniken erstellt. Aus Sicht der Metamodellierung beinhalten Modellierungstechniken eine Modellierungssprache sowie eine Vorgehensweise (vgl. Abschnitt 2.2.1 und die dortige Abbildung 4). Modellierungssprachen werden durch Metamodelle definiert. Die Metamodelle werden ebenfalls mit Hilfe einer Modellierungssprache erstellt. Diese wird als Metamodellierungssprache bezeichnet. Das Meta-Metamodell definiert wiederum die Metamodellierungssprache. Dadurch entsteht eine Hierarchie von Modellen, Metamodellen etc. (vgl. Abbildung 7).

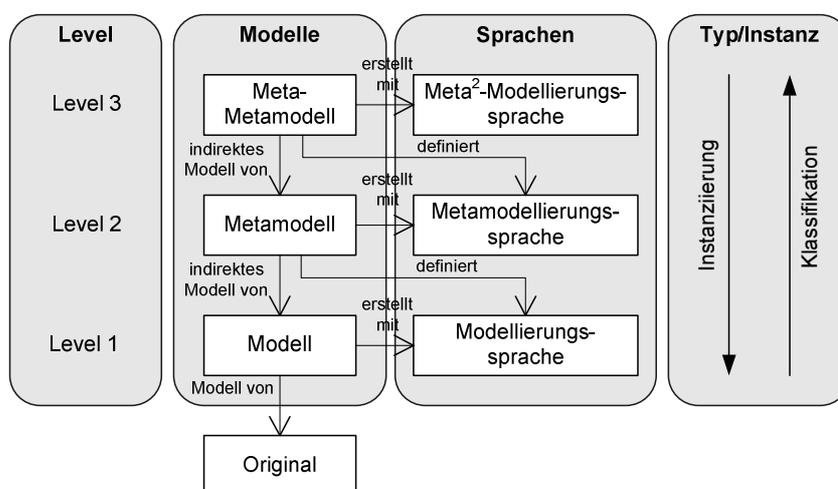


Abbildung 7: Modellierungshierarchie⁹⁰

2.2.4 Verwendete Meta-Modellierungssprachen

Für die Darstellung der Metamodelle und der Vorgehensmodelle der in Kapitel 3 und 4 beschriebenen Ansätze wird die Unified Modeling Language (UML) verwendet, da diese sich als führender Notationsstandard in Praxis und Wissenschaft etabliert hat.⁹¹

Die UML ist eine durch die Object Management Group (OMG) standardisierte graphische Sprache zur Beschreibung objektorientierter Modelle.⁹² Die UML ermöglicht die Modellierung verschiedenartiger Systeme, wie beispielsweise Software-Systeme oder Geschäftssysteme. Sie wurde durch Vertreter verschiedener Firmen entwickelt und standardisiert.

UML-Aktivitätsdiagramme dienen der Darstellung des Kontroll- und Informationsflusses bei Software-Systemen und können beispielsweise auch für die Modellierung von Geschäftspro-

⁸⁹ Vgl. Baumöl et al. (2005), S. 4.

⁹⁰ In Anlehnung an Karagiannis/Kühn (2002), S. 184.

⁹¹ Vgl. z.B. Balzert (2000), S. V.

⁹² Vgl. OMG (2003b) und Jeckle et al. (2004a).

zessen oder Workflows eines Unternehmens verwendet werden.⁹³ Mit Hilfe von UML-Klassendiagrammen kann die Struktur eines Systems dargestellt werden. Sie eignen sich beispielsweise für die Abbildung der Beziehungen zwischen einzelnen Mitarbeitern, Organisationseinheiten, Geschäftsobjekten und aussenstehenden Parteien eines Unternehmens.

2.2.4.1 Notation für Metamodelle

Wie bereits in Abschnitt 2.1 und 2.2.3 beschrieben, sind Metamodelle ein wesentlicher Bestandteil einer Methode. Sie definieren die Ergebnisse (Modelle bestehend aus Objekten und Beziehungen), welche bei der Anwendung einer bestimmten Technik erzeugt werden. Für die Darstellung aller Metamodelle der im Folgenden beschriebenen Methoden sowie des entwickelten Ansatzes werden UML-Klassendiagramme verwendet. Ein Klassendiagramm bietet die Möglichkeit, die Struktur eines Systems darzustellen. Es zeigt dessen wesentliche statische Eigenschaften sowie ihre Beziehungen zueinander. Das zentrale Konzept sind Klassen. Eine Klasse wird in der UML 2 als eine Sammlung von Exemplaren definiert, die über gemeinsame Eigenschaften, Einschränkungen und Semantik verfügen. Jede Klasse stellt demnach einen abstrahierten Sammelbegriff für eine Menge gleichartiger Dinge dar.⁹⁴ Für die Darstellung der Metamodelle werden Modellierungskonstrukte verwendet, die mit der in Tabelle 2 beschriebenen Bedeutung und der in UML verwendeten Notation belegt sind.

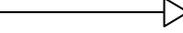
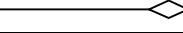
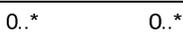
Metamodell		UML-Klassendiagramm	
Modellierungs-konstrukt	Beschreibung	UML-Konstrukt	Notation (Symbol)
Metaentitätstyp	Ein Metaentitätstyp beschreibt eine Sammlung von Objekten, die über gemeinsame Eigenschaften und eine gemeinsame Semantik verfügen.	Klasse	
Beziehungstyp	Ein Beziehungstyp beschreibt eine Menge semantisch gleichartiger Beziehungen zwischen zwei Metaentitätstypen.	Assoziation	
		Generalisierung	
		Komposition	
Eigenschaft	Eigenschaften beschreiben bestimmte Charakteristika eines Metaentitätstyps.	Klassenattribut	
Kardinalität	Jeder Beziehungstyp wird durch die Angabe von Kardinalitäten spezifiziert.	Multiplizität	

Tabelle 2: Modellierungskonstrukte für das Metamodell

⁹³ Vgl. hierzu z.B. *Oestereich et al.* (2004).

⁹⁴ Vgl. *Jeckle et al.* (2004b), S. 32.

2.2.4.2 Notation für Vorgehensmodelle

Die Vorgehensmodelle beschreiben Aktivitäten und deren Reihenfolge. Aktivitäten erzeugen bzw. verwenden Ergebnisse in Form von Ergebnisdokumenten. Dieser Zusammenhang wird durch ein- und ausgehende Ergebnisdokumente in den Vorgehensmodellen kenntlich gemacht. Die Ergebnisdokumente stellen eine Verbindung zu den Metamodellen her, da sie von diesen spezifiziert werden.

Für die Darstellung der Vorgehensmodelle gibt es eine Vielzahl möglicher Modellierungssprachen. Vor allem wegen ihrer weiten Verbreitung und ihrer Ausdrucksmächtigkeit werden sehr häufig die ereignisgesteuerten Prozessketten (EPK)⁹⁵ oder UML-Aktivitätsdiagramme⁹⁶ eingesetzt. Mit der Anwendung der EPK würde eine Modellierungssprache gewählt, die formal spezifiziert ist⁹⁷ und die über eine ausreichende semantische Ausdrucksmächtigkeit verfügt. Jedoch zwingt das Metamodell der EPK zu einer abwechselnden Folge von Aktivitäten und Ereignissen, die den Ablauf unnötig verlängert und damit zu einem unter Umständen unverständlicheren und unübersichtlicheren Diagramm führt.⁹⁸ Da auf den Zwang der Abbildung solcher Ereignisse zugunsten einer übersichtlicheren Darstellung verzichtet wird, werden im Folgenden UML-Aktivitätsdiagramme zur Modellierung der Aktivitäten der Vorgehensmodelle verwendet. UML-Aktivitätsdiagramme sind ebenfalls im Gesamtkontext der UML formal spezifiziert.⁹⁹ Da die Übertragung des Gesamtkontextes zu aufwendig erscheint, wird im Folgenden lediglich der relevante Ausschnitt betrachtet. Hierzu werden in Tabelle 3 die in Vorgehensmodellen enthaltenen Modellierungskonstrukte und deren Beziehungen sowie deren Repräsentation innerhalb der UML beschrieben.

Vorgehensmodell		UML-Aktivitätsdiagramm	
Modellierungs-konstrukt	Beschreibung	UML-Konstrukt	Notation (Symbol)
Aktivität	Durchführung einer atomaren Aktivität innerhalb des Vorgehensmodells	Aktionszustand	
Aktivitätshierarchie	Durchführung einer Aktivität, die aus Sub-Aktivitäten bzw. einem weiteren Vorgehensmodell besteht	Unteraktions-zustand	
Kontrollfluss	Übergang zwischen einer Aktivität und einer nachfolgenden Aktivität	Transition	
Objektfluss	Erzeugung oder Verwendung eines Ergebnisdokuments	Objektfluss	

⁹⁵ Vgl. z.B. *Becker et al.* (2002b).

⁹⁶ Vgl. z.B. *Jeckle et al.* (2004b), S. 199-266.

⁹⁷ Vgl. *Scheer* (2001), S. 170-171.

⁹⁸ Vgl. *Leist* (2004), S. 142.

⁹⁹ Vgl. *OMG* (2003b) und *Jeckle et al.* (2004b), S. 199-266.

Input/Output	Ergebnisdokumenttyp, der von einer Aktivität verwendet bzw. erzeugt wird	Objektzustand	
Start	Beginn des Vorgehensmodells	Anfangszustand	
Ende	Abschluss des Vorgehensmodells	Endzustand	

Tabelle 3: Modellierungsstrukturen für das Vorgehensmodell

2.3 Modellierung der Unternehmensarchitektur

Der Begriff Architektur wird in der Literatur sehr unterschiedlich verwendet.¹⁰⁰ Allen Definitionen liegt aber zugrunde, dass die Architektur mit einer Struktur in Verbindung gebracht wird, deren Komponenten und Beziehungen im Vordergrund stehen. Der IEEE-Standard-1471 definiert den Begriff Architektur beispielsweise wie folgt: „Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding its design and evolution“.¹⁰¹

In der vorliegenden Arbeit wird der Begriff Architektur als Konzept verstanden, welches nicht nur einzelne Komponenten und Beziehungen strukturiert abbildet, sondern auch deren Konstruktionsregeln festlegt.¹⁰²

Um die Struktur von Informationssystemen sowohl aus technischer als auch aus fachlicher Sicht beschreiben zu können, wird die Gestaltung der Informationssystemarchitektur zunehmend mit entsprechenden Methoden und Modellen auf anderen Betrachtungs- bzw. Abstraktionsebenen integriert und als umfassendes Gestaltungskonzept „Unternehmensarchitektur“ („Enterprise Architecture“) diskutiert.¹⁰³ Das Ziel einer Unternehmensarchitektur ist es, die Strukturen des Unternehmens und dessen Informationssysteme ganzheitlich abzubilden und damit die zukünftige Entwicklung und Gestaltung des Unternehmens zu unterstützen.¹⁰⁴ Die Architektur eines Unternehmens besteht aus verschiedenen Sichten auf dieses Unternehmen, wobei jede Sicht wiederum verschiedene Modelle enthält, die Ausschnitte des Unternehmens für einen bestimmten Zweck repräsentieren.¹⁰⁵ Dabei stehen vor allem die Identifikation der Gesamtzusammenhänge zwischen unterschiedlichen Bereichen innerhalb des Unternehmens sowie deren komplexitätsreduzierende Darstellung im Vordergrund. Im folgenden Abschnitt wird auf diese Gesamtzusammenhänge genauer eingegangen.

¹⁰⁰ Vgl. z.B. IEEE (2000), Heinrich (1993), S. 327, Krcmar (1990), S. 396, Scheer (1998), S.1 und Zachman (1987), S. 277.

¹⁰¹ IEEE (2000).

¹⁰² Vgl. Sinz (1999b), S. 1035 und IEEE (2000).

¹⁰³ Vgl. Schwinn/Winter (2005), S. 587.

¹⁰⁴ Vgl. Krcmar (2003), S. 39f. und Opengroup (2002).

¹⁰⁵ Vgl. Leist (2002), S. 7.

2.3.1 Darstellung von Gesamtzusammenhängen in Unternehmungen

Unternehmen können als offene Systeme hoher Komplexität in einer ebenfalls komplexen Umwelt definiert werden, deren Gestaltung und Lenkung ein umfassendes, integrierendes Denken erfordert.¹⁰⁶ Die komplexen Strukturen eines Unternehmens lassen sich aus den unterschiedlichsten Perspektiven und für die unterschiedlichsten Zwecke modellieren.¹⁰⁷ Um die Komplexität der Strukturen zu reduzieren, wird bei den meisten bereits existierenden Ansätzen¹⁰⁸ eine Hierarchie von Modellierungsebenen¹⁰⁹ gebildet. Die Modelle der verschiedenen Ebenen unterscheiden sich üblicherweise hinsichtlich des Aggregationsgrades, des Generalisierungsgrades, des Abstraktionsgrades sowie der jeweiligen Ziele der Modellbildung (vgl. hierzu Abschnitt 2.2.1). Viele Ansätze folgen dem hierarchischen Systemansatz nach MESAROVIC, demzufolge die Informationssysteme entsprechend den fachlichen Anforderungen gestaltet werden und damit die Ergebnisse der Modellierung einer Gestaltungsebene die Freiheitsgrade bei der Modellierung der nachfolgenden Ebenen einschränken.¹¹⁰

Architekturmodelle unterstützen Gestaltungs- und Entwicklungsprozesse. Die Unternehmensarchitektur sollte daher alle relevanten Elemente eines Unternehmens in aggregierter Form abbilden. Eine Einteilung in Architekturebenen und Sichten ist dabei sehr hilfreich, um unterschiedliche Aspekte in unterschiedlichen Modellen darzustellen, solange die Konsistenz durch die Definition eines ebenen- und sichtenübergreifenden Metamodells gewährleistet ist.

Eine Analyse der Literatur zum Thema Unternehmensarchitektur hat gezeigt, dass folgende Architekturebenen differenziert werden können:¹¹¹

- Geschäftsarchitektur als Gesamtzusammenhang der Leistungsverflechtung zwischen Organisationen in einem Wertschöpfungsnetzwerk
- Prozessarchitektur als Gesamtzusammenhang der Leistungsentwicklung, Leistungserstellung und des Leistungsvertriebs innerhalb einer Organisation
- Applikations- bzw. Integrationsarchitektur als Gesamtzusammenhang der informatischen Verflechtung von Applikationen innerhalb einer Organisation
- Softwarearchitektur als Gesamtzusammenhang der funktionalen Verflechtung zwischen Softwareartefakten einschliesslich Datenstrukturen innerhalb des Informationssystems

¹⁰⁶ Vgl. Ulrich (2001), S. 118.

¹⁰⁷ Vgl. Winter (2003d), S. 81.

¹⁰⁸ Vgl. z.B. Ferstl/Sinz (1995), Frank (2002), Krcmar (1990) und Scheer (1998).

¹⁰⁹ Die Begriffe Gestaltungs-, Modellierungs-, Abstraktions- und Architekturebene werden in dieser Arbeit synonym verwendet.

¹¹⁰ Vgl. Mesarovic et al. (1970).

¹¹¹ Die Ergebnisse der Analyse finden sich in Winter (2003d), S. 92-93, Winter/Fischer (2006), S. 6-7 und Hafner (2005), S. 32-34.

- Technologie- bzw. Infrastrukturarchitektur als Gesamtzusammenhang der eingesetzten IT-Komponenten, wie beispielsweise Server, Workstations, Netzwerkkomponenten etc.

Die zuvor genannten Ebenen finden sich nicht in allen Ansätzen. So definiert die Architektur integrierter Informationssysteme (ARIS)¹¹² beispielsweise nicht explizit eine Strategieebene. Das Zachman-Framework¹¹³ enthält zwar ein so genanntes Geschäftsmodell (Business Model). Dieses beschreibt allerdings die geschäftsorientierten Artefakte wie Wertschöpfungsketten, Ziele, Organisationseinheiten und Geschäftsprozesse nicht mit einem ausreichenden Detaillierungsgrad.

Zur weiteren Strukturierung und Komplexitätsreduktion werden in der Regel innerhalb jeder Architekturebene verschiedene Sichten gebildet.¹¹⁴ Diese reflektieren oftmals die verschiedenen Anspruchsgruppen in einem Unternehmen. Für jede Anspruchsgruppe ist in der Regel nicht die gesamte Unternehmensarchitektur, sondern lediglich ein bestimmter Ausschnitt davon relevant.¹¹⁵ Typische Sichten innerhalb der Ebenen der Prozess- und Applikationsarchitektur sind beispielsweise die Organisations- und Prozesssicht, die Funktions- und Kommunikationssicht sowie die Informations- bzw. Datensicht.¹¹⁶ Neben diesen primär ebeneninternen Sichten werden aber auch häufig für bestimmte Fragestellungen ebenenübergreifende Sichten definiert. So unterteilen ZACHMAN/SOWA beispielsweise in ihrem Architektur-Framework alle Ebenen nach den Sichten Data, Function, Network, People, Time und Motivation, die der Beantwortung verschiedener Fragestellungen dienen sollen (Was? Wie? Wo? Wer? Wann? Warum?).¹¹⁷

Innerhalb der betrieblichen Diskurswelt lassen sich die verschiedenen Modelle, wie bereits in 2.2.1 dargestellt, nach unterschiedlichen Abstraktionsebenen, Aggregations- und Generalisierungsgraden sowie ebenenübergreifenden Sichten strukturieren. Die Unternehmensarchitektur umfasst im Verständnis der vorliegenden Arbeit lediglich diejenigen Modelle, die alle aggregierten Artefakte und deren Beziehungen auf allen Abstraktionsebenen eines Unternehmens bzw. einer Organisation abbilden (vgl. Abbildung 8).¹¹⁸ Die detaillierte Zerlegung dieser Artefakte erfolgt dementsprechend in spezifischen Architekturen, wie beispielsweise Prozessarchitekturen, Datenarchitekturen oder Software- und Technologiearchitekturen. Für diese liegen oftmals bereits entsprechende Modellierungskonzepte vor. Allerdings existieren bisher keine Konzepte, die die Zusammenhänge zwischen diesen Architekturebenen definieren.¹¹⁹ Die Definition dieser Zusammenhänge ermöglicht die Gewährleistung der Konsistenz zwi-

¹¹² Vgl. *Scheer* (1998).

¹¹³ Vgl. *Zachman/Sowa* (1992).

¹¹⁴ Vgl. *Sinz* (1999a), S. 1036.

¹¹⁵ Vgl. *Lankhorst* (2005), S. 55.

¹¹⁶ Vgl. hierzu auch *Hafner* (2005), S. 36.

¹¹⁷ Vgl. *Zachman/Sowa* (1992), S. 600f.

¹¹⁸ Vgl. hierzu *Winter/Fischer* (2006), S. 4.

¹¹⁹ Vgl. hierzu z.B. *Jonkers et al.* (2004), S. 264.

schen Modellen auf unterschiedlichen Architekturebenen und bildet somit beispielsweise auch eine wichtige Grundlage für die systematische, modellbasierte Lösung von Problemstellungen des IT-Business-Alignment¹²⁰, Service Management¹²¹ oder Compliance Management¹²².

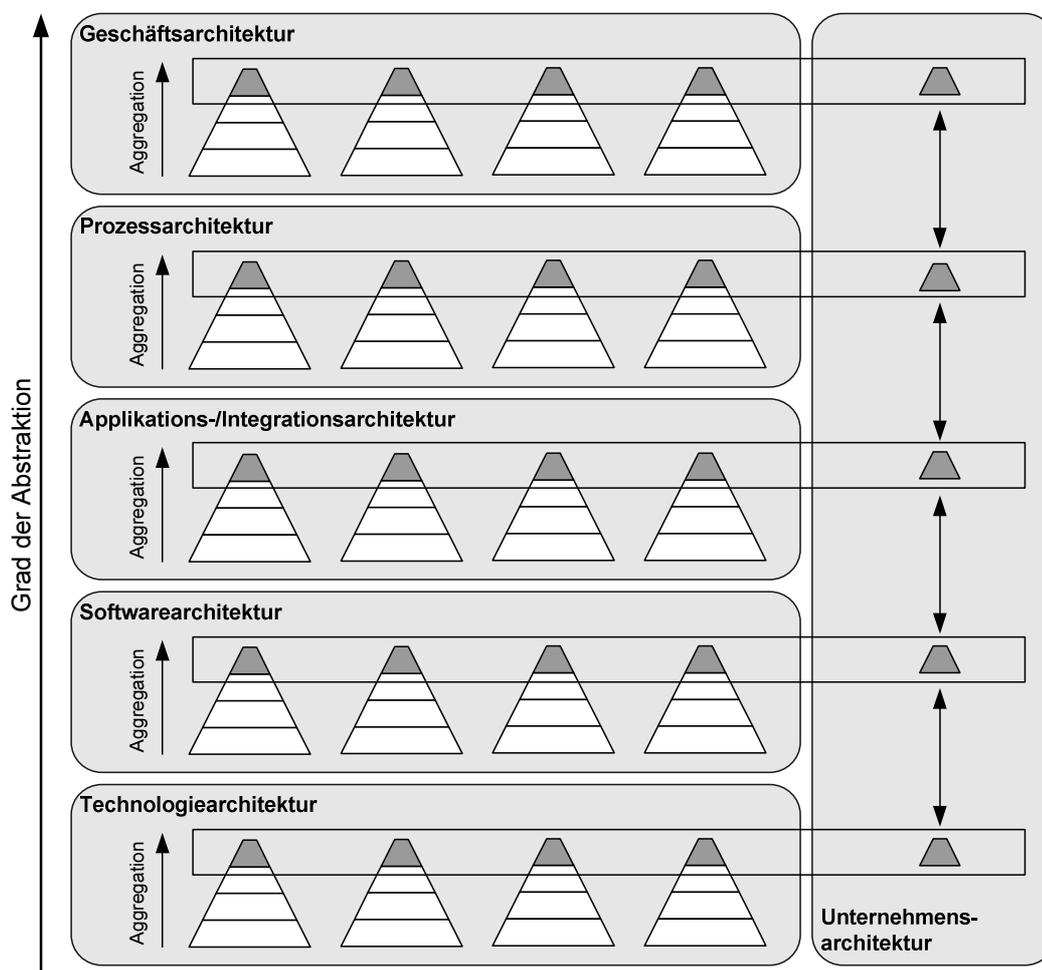


Abbildung 8: Unternehmensarchitektur¹²³

2.3.2 Business Engineering

Für die Entwicklung einer Unternehmensarchitektur ist, analog zur Konstruktion von Methoden und Referenzmodellen, ein systematischer, umfassender Ansatz erforderlich, um die Konsistenz der erzeugten Ergebnisse gewährleisten zu können. Der Begriff „Business Engineering“ (BE)¹²⁴ bezeichnet einen Ansatz zur ingenieurmässigen, methoden- und modellbasierten Konstruktion von Unternehmensmodellen. Das Business Engineering bildet daher den Forschungsrahmen des Dissertationsvorhabens. Im Folgenden werden die

¹²⁰ Zum Thema IT-Business-Alignment vgl. z.B. *Henderson/Venkatraman* (1993).

¹²¹ Zum Thema IT-Service-Management vgl. z.B. *Hochstein/Hunziker* (2003).

¹²² Für eine Erläuterung zum Thema Compliance Management vgl. z.B. *Geissler* (2004).

¹²³ Vgl. *Winter/Fischer* (2006), S. 4.

¹²⁴ Vgl. *Österle/Winter* (2003).

grundlegende Ausrichtung, die Gestaltungsebenen und Inhalte sowie das vereinfachte Metamodell des Business Engineering vorgestellt.

Der BE-Ansatz verfolgt das Ziel, Unternehmen bei der Transformation in das Informationszeitalter zu unterstützen. Transformation bedeutet dabei, vorhandene Unternehmen zu restrukturieren oder neue Unternehmen zu schaffen. Die Transformation wird meist durch IT-Innovationen ausgelöst.¹²⁵ Um diese systematisch umzusetzen, sind entsprechende Vorgehensweisen, Methoden und Werkzeuge notwendig. Das in Abschnitt 2.1 beschriebene Methoden-Engineering stellt deshalb ein wichtiges Element des Business Engineering dar, da es das ingenieurmässige Vorgehen bei der Transformation eines Unternehmens sichert.

Der BE-Ansatz ist ein interdisziplinärer Ansatz, der die vielfältigen Aspekte der Transformation nach unterschiedlichen Ebenen gliedert (vgl. Abbildung 9). Die wesentlichen Dimensionen des Ansatzes sind die fachliche Dimension, bestehend aus einer Strategie-, Organisations- und Informationssystemebene, sowie die politisch-kulturelle Dimension, die Themengebiete wie Motivation und Führung, Verhalten, Kommunikation und Machtverhältnisse abdeckt.

Nach ÖSTERLE ET AL. sind verteilte Strukturen nicht nur ein Phänomen von Informationssystemen und Softwarekomponenten.¹²⁶ Auch Geschäftsprozesse erstrecken sich zunehmend über die Grenzen einzelner Geschäftseinheiten oder sogar Unternehmen hinweg. Zudem basieren neue Geschäftsmodelle immer häufiger auf Kooperationen oder Kooptionen. Nach WINTER verfolgt der Business Engineering-Ansatz deshalb das Ziel, die konsistente Spezifikation von vernetzten Strukturen auf allen Architekturebenen eines Unternehmens zu unterstützen.¹²⁷

Auf Strategieebene werden die Rolle des Unternehmens im Wertschöpfungsnetzwerk, die relevanten Kundenprozesse und -segmente sowie die wesentlichen Leistungen des Unternehmens festgelegt. Des Weiteren wird das Zielsystem des Unternehmens in Form von kritischen Erfolgsfaktoren und Kennzahlen beschrieben. Modellierungstechniken für die Gestaltung von Geschäftsnetzwerken und -strategien sowie entsprechende Notationen finden sich beispielsweise in Arbeiten von WINTER und HEINRICH.¹²⁸

Auf Organisationsebene werden die zur Umsetzung der Strategien notwendigen Geschäftsprozesse und ihr Zusammenwirken beschrieben. Die Geschäftsprozesse erstrecken sich in der Regel über die Grenzen eines einzelnen Unternehmens bzw. einer einzelnen Geschäftseinheit hinweg. Für jeden Geschäftsprozess werden zudem die zu erbringenden Prozessleistungen spezifiziert, die zu deren Erbringung notwendigen Aktivitäten einschliesslich deren Abfolgen festgelegt und Performance-Indikatoren für die Führung der Prozesse sowie Verantwortlich-

¹²⁵ Vgl. *Österle/Winter* (2003), S. 4.

¹²⁶ Vgl. *Österle et al.* (2001).

¹²⁷ Vgl. *Winter* (2003d).

¹²⁸ Vgl. z.B. *Winter* (2003b), *Heinrich/Winter* (2004) und *Heinrich* (2002b).

keiten bzw. Organisationsstrukturen definiert. Ausserdem werden auf dieser Ebene Informationsobjekte und -flüsse in Form abstrakter Informationsmodelle abgebildet. Techniken für die Modellierung von Geschäftsprozessen und Organisationsstrukturen finden sich beispielsweise in Arbeiten von ÖSTERLE, BRECHT und IMG.¹²⁹

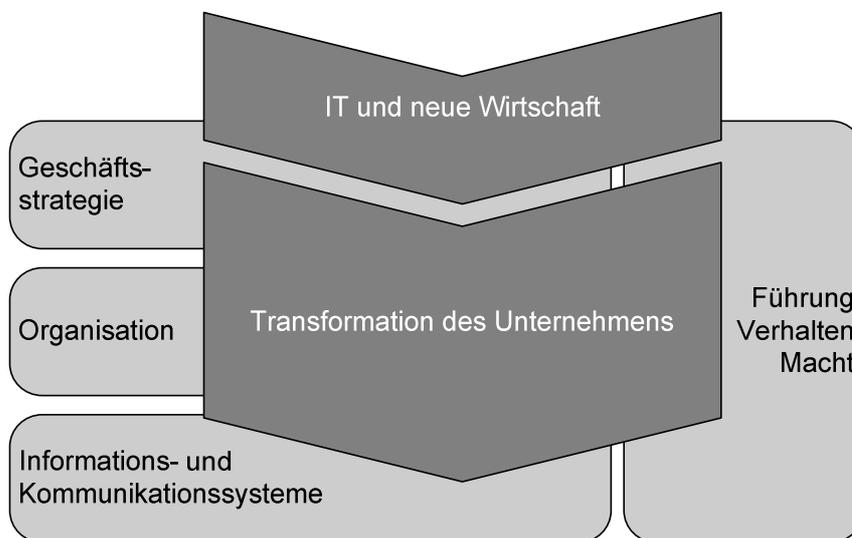


Abbildung 9: Business-Engineering-Landkarte¹³⁰

Die Informationssystemebene wird weiter unterteilt in die Ebenen Applikationen sowie Softwarekomponenten und Datenstrukturen. Das Gestaltungsziel auf der Applikationsebene besteht darin, Komponenten der Informationssysteme mit den Geschäftsanforderungen zu verbinden, indem entsprechende Applikationsstrukturen und Integrationssysteme (z.B. Data-Warehouse-Systeme, Operational-Data-Store-Systeme oder Enterprise-Application-Integration-Systeme) entworfen werden. Applikationen repräsentieren konzeptionelle Konstrukte, die zur Strukturierung der Informationsflüsse, Geschäftsprozessunterstützung und Verantwortlichkeiten auf Unternehmens- bzw. Geschäftseinheitsebene verwendet werden.¹³¹ Ihre Strukturen werden aus der Analyse von Aktivitäten, Informationsbedürfnissen und Verantwortlichkeiten abgeleitet. Ausserdem werden Informations- und Kontrollflüsse zwischen Applikationen spezifiziert, die auf unterschiedlichen Plattformen laufen oder in unterschiedlichen Unternehmen bzw. Geschäftseinheiten eingesetzt werden. Ein umfassender Ansatz zur Modellierung der Unternehmensarchitektur sollte vernetzte und verteilte Strukturen sowie die durch diese Strukturen resultierenden Herausforderungen an das Management reflektieren. Ein Modell für die Abbildung der Applikationsarchitektur wird beispielsweise von WINTER vorgeschlagen.¹³²

¹²⁹ Vgl. z.B. Österle (1995), IMG (1997) und Brecht (2002).

¹³⁰ Vgl. Österle/Winter (2003), S. 12.

¹³¹ Vgl. Braun/Winter (2005b), S. 67.

¹³² Vgl. Winter (2003a).

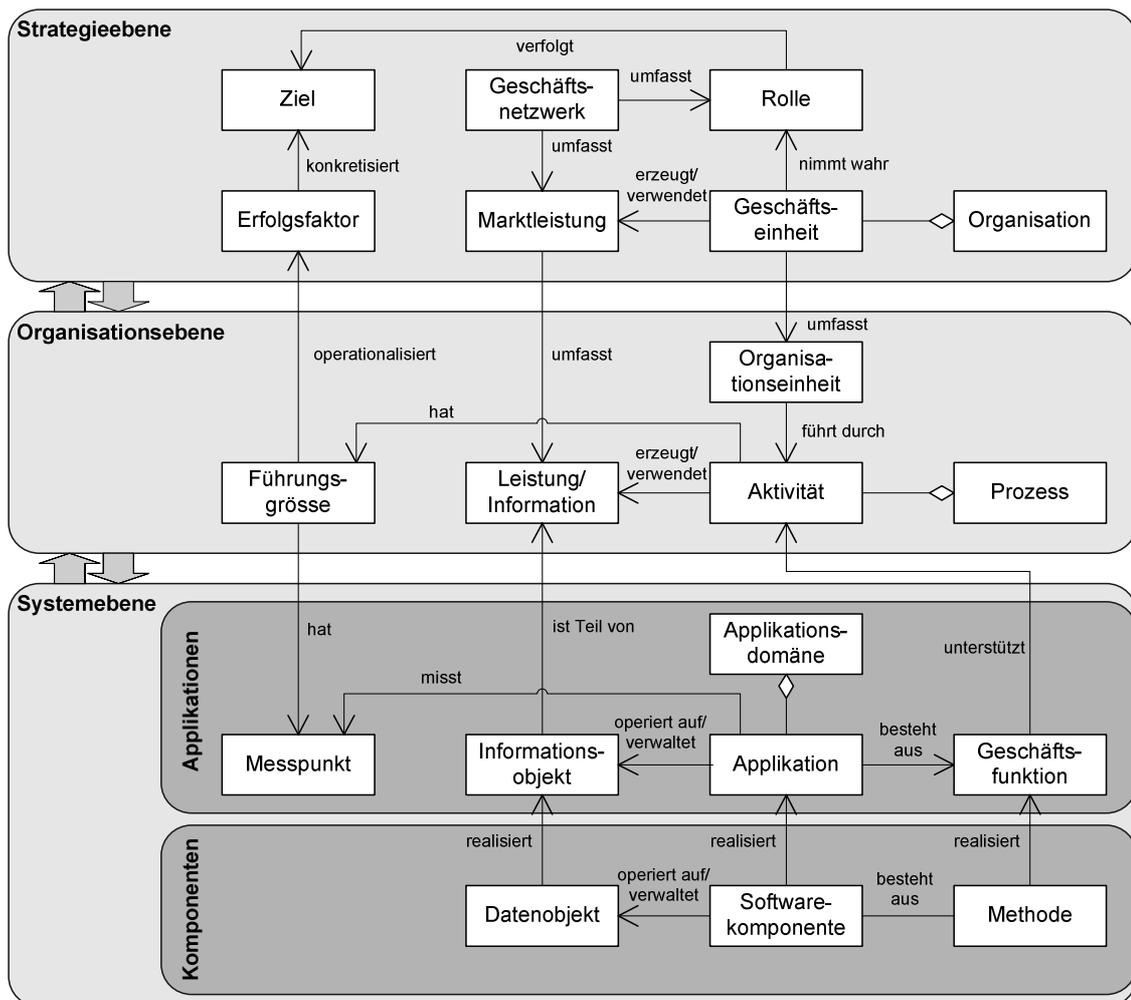


Abbildung 10: Vereinfachtes Metamodell des BE¹³³

Das primäre Ziel auf der Softwarekomponenten- und Datenstrukturebene ist die optimale (Wieder-)Verwendung von Softwareartefakten und entsprechenden Datenstrukturen. Gestaltungsobjekte für die Modellierung der Softwarekomponenten- und Datenstrukturebene werden beispielsweise von SCHWINN¹³⁴ vorgeschlagen. Er stellt vor allem diejenigen Gestaltungsobjekte in den Vordergrund, die zur Durchführung eines Applikationsintegrationsprojekts notwendig sind. Abbildung 10 zeigt zur Veranschaulichung eine vereinfachte Version des ebenenübergreifenden Metamodells, das die Konsistenz zwischen den einzelnen Ebenen gewährleistet.

Mit Hilfe des BE-Ansatzes sind in den letzten Jahren einige Methoden entstanden, die auf verschiedene Branchen oder spezielle Problemstellungen ausgerichtet sind. Einige dieser Methoden wurden für die Anwendung in der Praxis in einem Methodenset mit dem Namen PROMET (Prozessmethode)¹³⁵ zusammengefasst und weiterentwickelt. Eine weitere Methode

¹³³ Vgl. Braun/Winter (2005b), S. 66 und Wortmann (2005), S. 14.

¹³⁴ Vgl. Schwinn (2005).

¹³⁵ Vgl. IMG (1997) und <http://www.promet-web.com>.

ist die BAI-Methode¹³⁶, welche im Kompetenzzentrum Bankenarchitekturen im Informationszeitalter (CC BAI)¹³⁷ entwickelt wurde und als Grundlage für das Dissertationsprojekt dient. Die Methoden von PROMET als auch die BAI-Methode sind einheitlich nach den Grundsätzen des Methoden-Engineering¹³⁸ strukturiert.

2.3.3 Methoden und Konsistenz

Nach der Beschreibung der Gestaltungsebenen des Business Engineering wird in diesem Abschnitt darauf eingegangen, warum ein systematischer, umfassender Ansatz, basierend auf einem ebenen- und sichtenübergreifenden Metamodell, für die Gestaltung von Organisationsstrukturen wichtig ist.¹³⁹

Traditionell verfolgt der BE-Ansatz ein Top-Down-Vorgehen (Top-Down-Alignment), d.h. eine Veränderung der Geschäftsstrategie erfordert eine Umgestaltung der Geschäftsprozesse sowie Organisationsstrukturen und dies wiederum eine Anpassung der unterstützenden Informationssystemstrukturen.¹⁴⁰ Entsprechend den Vorgehensweisen auf den einzelnen Gestaltungsebenen werden unterschiedliche Ergebnisdokumente erzeugt. Liegt diesen Ergebnisdokumenten ein ebenenübergreifendes Metamodell zugrunde, so kann die Konsistenz der gesamten Spezifikation der Unternehmensarchitektur gewährleistet werden. Die Konstruktion und Anwendung entsprechender Methoden ist deshalb für die Generierung konsistenter Ergebnisse bei der Abbildung der Unternehmensstrukturen unerlässlich. In der Praxis ist es oftmals schwierig, alle Abhängigkeiten zu identifizieren und in einem Metamodell zu erfassen. Dies liegt unter anderem darin begründet, dass die Schlüsselemente auf den unterschiedlichen Gestaltungsebenen unterschiedliche Lebenszyklen besitzen und im Verantwortungsbereich von unterschiedlichen Unternehmensbereichen bzw. Organisationseinheiten liegen. Die meisten existierenden Ansätze zur Gestaltung und Abbildung von Unternehmensstrukturen schlagen Modellierungssprachen für bestimmte Ebenen oder Sichten vor, die allerdings nicht durch ein gemeinsames zugrunde liegendes Metamodell miteinander verbunden sind. Dadurch entstehen oftmals inkonsistente Modelle.

Ein ebenenübergreifendes Metamodell, das die Abhängigkeiten zwischen den einzelnen Modellen definiert, ist nicht nur für ein Top-Down-Vorgehen, sondern auch für einen Bottom-Up-Ansatz (Bottom-Up-Alignment) eine wichtige Grundlage.¹⁴¹ Dadurch kann besser analysiert und abgeschätzt werden, welche Auswirkungen die Veränderung von Schlüsselementen einer Ebene auf die Elemente der übergeordneten Ebenen hat (vgl. Abbildung 11). Typi-

¹³⁶ Vgl. *Choinowski et al.* (2003).

¹³⁷ Vgl. *Leist/Winter* (2002).

¹³⁸ Vgl. *Gutzwiller* (1994) und *Heym* (1993).

¹³⁹ Vgl. im Folgenden auch *Winter* (2005).

¹⁴⁰ Vgl. *Österle* (1995), S. 24.

¹⁴¹ Nach *ÖSTERLE/BLESSING* erfordert die Ebenenbildung im Business Engineering keine ausschliessliche Top-Down-Vorgehensweise. Vgl. *Österle/Blessing* (2003), S. 77.

sche Fragen, die dadurch beantwortet werden können, sind beispielsweise: Welche Auswirkungen hat die Veränderung der technischen Architektur auf die Struktur der Applikationen? Wie wirkt sich die Implementierung neuer Applikationen auf die Organisationsstrukturen und Abläufe aus? Welche Auswirkungen haben organisatorische Veränderungen auf die Gestaltung der Vertriebskanäle und Partnerbeziehungen? Erfahrungen mit Veränderungen der Organisationsstrukturen und Geschäftsmodelle, die durch neue Technologien induziert wurden, wie beispielsweise E-Commerce oder mobile Technologien, bestätigen die Relevanz dieses Bottom-Up-Ansatzes in der Praxis.

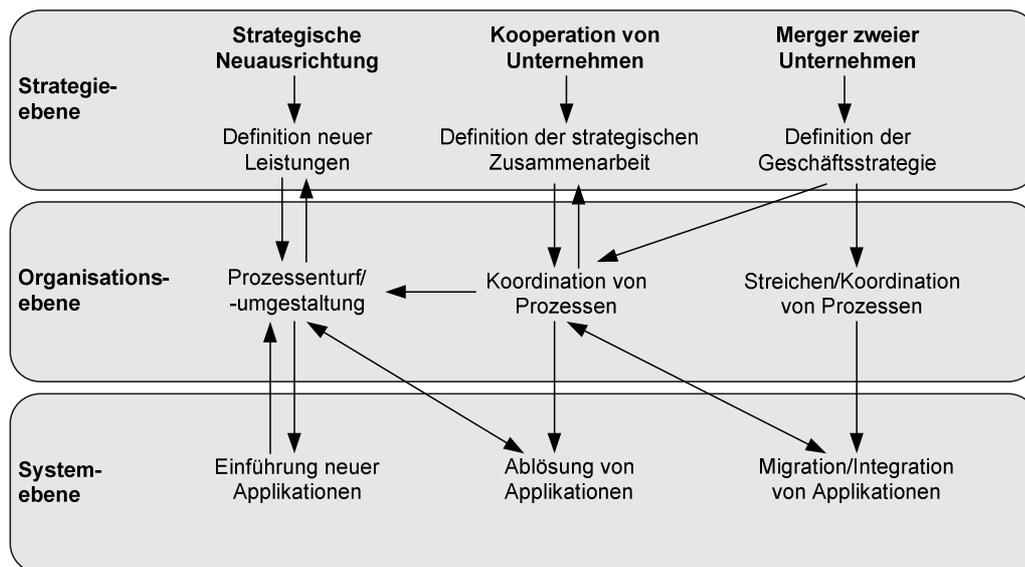


Abbildung 11: Beispiel für Wechselwirkungen zwischen den Ebenen¹⁴²

2.3.4 Die BAI-Methode

Die BAI-Methode unterstützt die systematische Gestaltung von Unternehmensarchitekturen, insbesondere im Bankensektor. Der aktuelle Entwicklungsstand der BAI-Methode ist das Ergebnis des Kompetenzzentrums Bankenarchitekturen im Informationszeitalter (CC BAI).¹⁴³ Im CC BAI wurden gemeinsam mit den Partnerunternehmen Architekturkonzepte für den Vertriebsbereich im Retail Banking erarbeitet. Auf der Grundlage von Good Practices wurden unternehmensspezifische Modelle sowie Referenzmodelle für die Geschäfts-, Organisations- und Informationssystemebene identifiziert. Zusätzlich wurde ein Vorgehensmodell für die konsistente Gestaltung von Geschäftsmodellen, Prozessen und Applikationen aufgestellt.

Die BAI-Methode wurde entsprechend den Grundsätzen des Methoden-Engineering (vgl. Abschnitt 2.1.2) konstruiert. Sie bietet auf jeder Gestaltungsebene Vorgehensmodelle, Aktivitäten, Techniken sowie Metamodelle für die Spezifikation der Ergebnisdokumente. Um die

¹⁴² Vgl. in Anlehnung an Vogler (2003), S. 21.

¹⁴³ Vgl. Leist/Winter (2002) und Choinowski et al. (2003).

Zusammenhänge zwischen den drei Ebenen abzubilden, besitzt die Methode ansatzweise ein ebenenübergreifendes Metamodell.

Grundsätzlich beschreibt die BAI-Methode ein Top-Down-Vorgehen. Zunächst wird auf Strategieebene das Geschäftsnetzwerk festgelegt, das Leistungsmodell entwickelt, die Kundenprozessanalyse durchgeführt sowie das Zielsystem definiert. Anschliessend folgen auf Organisationsebene der Entwurf der Prozesslandkarte, der Prozessvision und der Prozessführung sowie die Entwicklung der Ablauf- und Aufbauorganisation. Auf Applikationsebene wird letztendlich anhand der Vorgaben auf Organisationsebene die Applikationsarchitektur abgeleitet. Beim Übergang von einer Ebene zur nächsten werden jeweils diejenigen Elemente herausgestellt, die in der darunter liegenden Ebene Verwendung finden bzw. einfließen.¹⁴⁴

Der in der vorliegenden Arbeit präsentierte Ansatz zur Modellierung der Unternehmensarchitektur enthält die wesentlichen Konzepte der BAI-Methode. Diese werden daher in Kapitel 4 detaillierter beschrieben.

Trotz der ursprünglichen Fokussierung der BAI-Methode auf die Finanzbranche lassen sich deren Konzepte weitgehend auch auf andere Branchen übertragen. Diese analog zur Referenzmodellierung als Allgemeingültigkeit bezeichnete Eigenschaft einer Methode kann zwar nur eingeschränkt erzielt und überprüft werden.¹⁴⁵ Dennoch wird an den in dieser Arbeit auf Basis der BAI-Methode entwickelten Ansatz ein branchenübergreifender Allgemeingültigkeitsanspruch gestellt. Dieser Allgemeingültigkeitsanspruch muss durch den Einsatz des entwickelten Prototyps und des Metamodells in zahlreichen Unternehmen verschiedener Branchen überprüft und damit auf eine breite Fallstudienbasis gestellt werden.

2.4 Werkzeugkonstruktion und -unterstützung

Aufgrund der sich schnell und ständig ändernden Geschäftsanforderungen wird die Entwicklung geeigneter Informationssysteme zunehmend komplexer. Es hat sich gezeigt, dass Software-Werkzeuge, deren zugrunde liegende Metamodelle nicht fest vorgegeben sind, sondern flexibel angepasst werden können, den Umgang mit dieser Komplexität vereinfachen.¹⁴⁶ Das Methoden-Engineering (vgl. Abschnitt 2.1.1) erfordert folglich ebenfalls entsprechende Software-Werkzeuge, um die Modellierungstechniken, Modellierungssprachen und Vorgehensmodelle einer bestimmten Methode einrichten, umsetzen und flexibel anpassen zu können. Anpassbare und erweiterbare Metamodelle sowie wieder verwendbare Mechanismen sind eine wichtige Voraussetzung für die effiziente Konstruktion, Anwendung und Modifikation von Methoden und Modellen.

¹⁴⁴ Vgl. *Choinowski et al.* (2003), S. 35.

¹⁴⁵ Vgl. hierzu z.B. *vom Brocke* (2003b), S. 31f. und *Wortmann* (2005), S. 100.

¹⁴⁶ Vgl. hierzu z.B. *Karagiannis/Kühn* (2002).

Methoden für die Gestaltung von Unternehmensarchitekturen umfassen Vorgehensweisen, Metamodelle und Modellierungstechniken auf den drei Ebenen Geschäftsstrategie, Organisation und Informationssystem (vgl. Abschnitt 2.1). In der Vergangenheit konzentrierte sich die werkzeugunterstützte Modellierung auf Teilaspekte der gesamten Unternehmensarchitektur, z.B. die Prozessmodellierung mit dem ARIS Toolset. Ausserdem basierten diese Werkzeuge vor allem auf unveränderlichen, nicht anpassbaren Metamodellen.

Viele wissenschaftliche Arbeiten haben sich auf die Entwicklung von Software-Werkzeugen für die Erfassung von Methodenwissen¹⁴⁷ als auch auf die Konstruktion von anpassbaren Metamodellierungswerkzeugen¹⁴⁸ konzentriert. Metamodellierungswerkzeuge, oftmals auch als Meta-CASE-Tools¹⁴⁹ bezeichnet, bieten die Möglichkeit, Modellierungswerkzeuge für die Verwendung einer bestimmten Methode zu konfigurieren. Sie können formale Beschreibungen von Methoden interpretieren und daraus geeignete Modellierungswerkzeuge generieren, die bestimmte Methoden unterstützen.¹⁵⁰ Diese Metamodellierungswerkzeuge konnten sich allerdings trotz ihrer Relevanz in der Praxis bisher nicht durchsetzen. Ein Grund dafür ist vermutlich deren höhere Komplexität.

In den letzten Jahren versuchen nun aber zunehmend Hersteller von Modellierungswerkzeugen für bestimmte Anwendungsbereiche, wie z.B. das Business Process Management oder das Architekturmanagement¹⁵¹, Metamodellierungskonzepte in ihre Produkte zu integrieren, um diese einfacher an die verwendete Methode und das unternehmensindividuelle Umfeld anpassen zu können. Diese werden in der vorliegenden Arbeit nicht als Meta-CASE-Tools, sondern als Metamodellierungsplattformen bezeichnet.¹⁵² Metamodellierungsplattformen ermöglichen im Verständnis der vorliegenden Arbeit¹⁵³

- die problemspezifische Definition von Methodenelementen, wie z.B. von Metamodellen,
- die Definition von Mechanismen, die auf Modelle und deren zugrunde liegende Metamodelle angewendet werden können,
- die Definition von Vorgehensmodellen, die das Vorgehen bei der Anwendung der Metamodelle und der entsprechenden Mechanismen beschreiben.

Die folgende Abbildung 12 zeigt die Anwendung des Metamodellierungskonzeptes auf die Werkzeugkonstruktion. Spezifische Modellierungswerkzeuge (Konfigurationen) können von

¹⁴⁷ Vgl. z.B. *Heym* (1993).

¹⁴⁸ Vgl. z.B. *Kelly/Smolander* (1996) und *Junginger et al.* (2000).

¹⁴⁹ CASE steht für Computer Aided Software Engineering.

¹⁵⁰ Vgl. hierzu *Nuseibeh et al.* (1996), S. 267.

¹⁵¹ Vgl. z.B. *Junginger et al.* (2000) und *James* (2004b).

¹⁵² Vgl. z.B. *Karagiannis/Kühn* (2002).

¹⁵³ Vgl. *Karagiannis/Kühn* (2002), S. 186f.

der Metaebene der Metamodellierungsplattform durch Anpassung des zugrunde liegenden Meta-Metamodells abgeleitet werden.¹⁵⁴

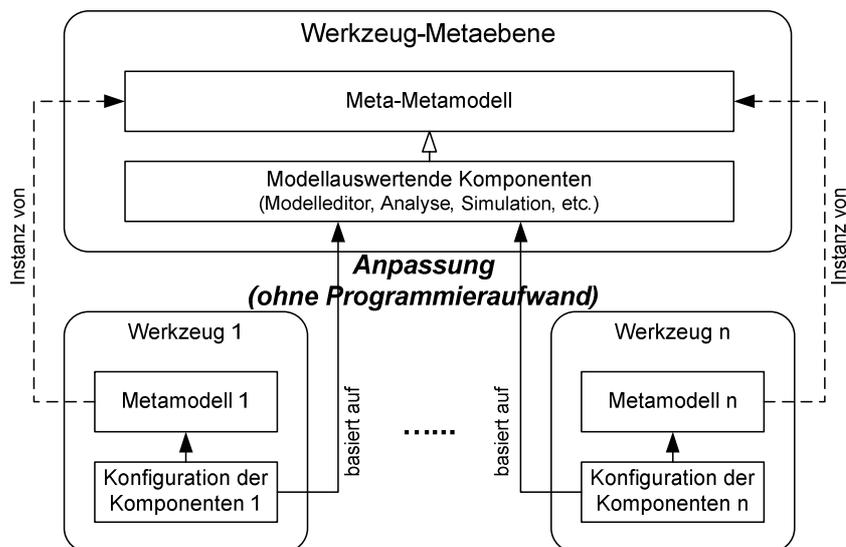


Abbildung 12: Metamodellierungskonzepte für die Werkzeugkonstruktion¹⁵⁵

Es gibt zahlreiche Gründe für die Verwendung von Modellierungswerkzeugen:¹⁵⁶ Sie bieten zum einen Unterstützung bei der Standardisierung von Semantik und Syntax der verwendeten Modellierungssprachen und -techniken einer Methode. Darüber hinaus gewährleisten sie die Korrektheit und Konsistenz der erstellten Modelle durch automatisierte Konsistenzprüfungen und die Anwendung von Gestaltungsprinzipien. In der Regel können auch Impact-Analysen und quantitative Analysen durchgeführt werden. Zudem können Modellierungswerkzeuge den Modellierer bei der Verwendung bzw. Wiederverwendung der erstellten Modelle unterstützen.

Die meisten neueren Ansätze zur Gestaltung von Unternehmensarchitekturen fokussieren auf die Abbildung von Beziehungen zwischen aggregierten Artefakten unterschiedlichen Typs. Vorrangiges Ziel dabei ist vor allem die Identifizierung von Inkonsistenzen zwischen den einzelnen Artefakten sowie von Richtlinien für die konsistente, gesamthafte Abbildung von Unternehmen und insbesondere deren Informationssystemen. Getrieben durch Aspekte wie Anwendbarkeit und betriebswirtschaftlicher Nutzen wird die Unternehmensarchitektur zunehmend als Konzept betrachtet, das die Analyse, Optimierung und Abstimmung von Geschäftsstrategien, Organisationsstrukturen, Geschäftsprozessen, Informationsflüssen sowie Informationssystemen unterstützt.¹⁵⁷ Damit die Unternehmensarchitektur diese aktive Rolle übernehmen kann, ist es erforderlich, dass hierfür geeignete Modellierungswerkzeuge zur

¹⁵⁴ Vgl. Junginger et al. (2000), S. 393.

¹⁵⁵ Vgl. Junginger et al. (2000), S. 393.

¹⁵⁶ Vgl. im Folgenden auch Lankhorst (2005), S. 248.

¹⁵⁷ Vgl. Schekkerman (2004), S. 201.

Verfügung stehen. Diese müssen nicht nur die Dokumentation von Artefakten und die Identifizierung von Inkonsistenzen ermöglichen. Ein geeignetes Werkzeug muss zudem die Entwicklung, die Speicherung, die Kommunikation, die Analyse sowie die Modifikation und Erweiterung von allen relevanten Bestandteilen der Unternehmensarchitektur unterstützen.

Ebenso wie die Methoden befinden sich auch die Werkzeuge zur Unterstützung der Unternehmensmodellierung noch in einem frühen Entwicklungsstadium.¹⁵⁸ Von grosser Bedeutung ist dabei vor allem die Definition und Abbildung von Abhängigkeiten zwischen Modellen auf unterschiedlichen Architekturebenen. Aufgrund der Komplexität von Geschäftsstrukturen in der Praxis können diese Abhängigkeiten oft nur schwer identifiziert und verstanden werden. Für den Erfolg einer Unternehmensarchitektur ist es aber entscheidend, dass solche Abhängigkeiten identifiziert und die Auswirkungen von Änderungen an den Elementen analysiert und abgeschätzt werden können.

Auf dem Markt existieren bereits zahlreiche Anbieter für Werkzeuge zur Unternehmensmodellierung.¹⁵⁹ Deren Produkte unterstützen allerdings oftmals nur Teilbereiche der Unternehmensarchitektur oder bieten begrenzt Schnittstellen zu anderen Werkzeugen. Dies lässt sich damit begründen, dass viele der bisher als Werkzeuge für die Modellierung von Unternehmensarchitekturen angebotenen Lösungen ursprünglich als anwendungsspezifische Werkzeuge, beispielsweise für die Softwareentwicklung oder die Geschäftsprozessmodellierung, entwickelt wurden. Die geringe Interoperabilität zwischen den Werkzeugen hat sowohl technische als auch konzeptionelle Gründe. Die aktuell auf dem Markt verfügbaren Werkzeuge können nach TER DOERST/LANKHORST in folgende Kategorien eingeteilt werden (vgl. Abbildung 13):¹⁶⁰

- Unternehmensarchitektur: Dabei handelt es sich um speziell für die Unternehmensmodellierung entwickelte Werkzeuge, die unterschiedliche Modellierungskonzepte für unterschiedliche Architekturebenen umfassen.
- Softwareentwicklung: Diese Werkzeuge stammen ursprünglich aus der Softwareentwicklung und integrieren zunehmend Techniken für die Modellierung von Geschäftsprozessen und Unternehmensarchitekturen.
- Geschäftsprozessmanagement: Analog zu den Softwareentwicklungswerkzeugen integrieren diese Werkzeuge zunehmend Modellierungstechniken der Softwareentwicklung (z.B. UML) und abstraktere Modellierungskonzepte für die Unternehmensmodellierung.
- Repositorys und IT-Management: Diese Werkzeuge werden um Modellierungs- und Analysefunktionalitäten erweitert, die auch für die Unternehmensmodellierung relevant sind.

¹⁵⁸ Vgl. *Schekkerman* (2004), S. 201.

¹⁵⁹ Für eine Übersicht aktuell auf dem Markt verfügbarer Werkzeuge zur Unternehmensmodellierung vgl. *IFEAD* (2006).

¹⁶⁰ Vgl. *ter Doest/Lankhorst* (2004), S. 2.

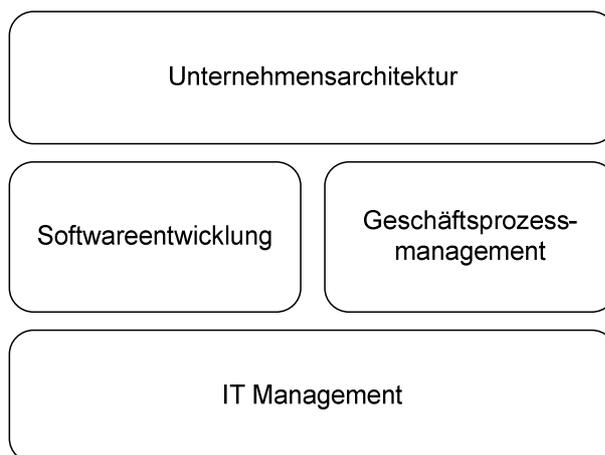


Abbildung 13: Kategorisierung der Werkzeugunterstützung¹⁶¹

In der Praxis werden oftmals Grafik-Tools wie Microsoft Visio oder Microsoft Powerpoint verwendet, um die Artefakte der Unternehmensarchitektur zu dokumentieren.¹⁶² Allerdings werden durch diese Werkzeuge die Prüfung der Konsistenz zwischen den einzelnen Modellen und die Wiederverwendung von Modellen bzw. Modellkomponenten nicht unterstützt. Für den Erfolg eines Unternehmensarchitektur-Projektes ist daher entscheidend, dass von Beginn an professionelle, speziell für die Unternehmensmodellierung entwickelte Werkzeuge zum Einsatz kommen. Professionelle Modellierungswerkzeuge speichern alle Artefakte der Unternehmensarchitektur in einer Datenbank (Repository) und stellen Mechanismen für die Wiederverwendung und die Darstellung in unterschiedlichen Formen zur Verfügung.¹⁶³ Entsprechend der in dieser Arbeit angeführten Charakterisierung der Unternehmensarchitektur (vgl. Abschnitt 2.3) sollten Werkzeuge für die Unternehmensmodellierung daher auf einer Datenbank basieren,

- die Informationen über (a) Produkte/Services, (b) Ziele, Erfolgsfaktoren, Performance-Indikatoren und Messgrößen, (c) Geschäftsnetzwerke, Rollen und Leistungsflüsse innerhalb des Netzwerkes, (d) Aktivitäten, Organisationseinheiten und Kontrollflüsse, (e) Informationsobjekte und Informationsflüsse, (f) Applikationen, Funktionen und Verantwortlichkeiten sowie (g) Softwarekomponenten und eventuell auch (h) IT-Plattformen speichert,
- der ein verständliches Metamodell zugrunde liegt, das diese Artefakte zueinander in Beziehung stellt,
- die die Darstellung der Informationen in unterschiedlichen Formen, z.B. grafisch oder textuell, ermöglicht und

¹⁶¹ Vgl. *ter Doest/Lankhorst* (2004), S. 3.

¹⁶² Vgl. hierzu z.B. *IFEAD* (2005), S. 29.

¹⁶³ Vgl. *Schekkerman* (2004), S. 201.

- die die Auswertung und Analyse der gespeicherten Informationen unterstützt.¹⁶⁴

Da die Artefakte der Unternehmensarchitektur lediglich in einer verständlichen, aggregierten Form abgebildet werden, ist es erforderlich, dass das Werkzeug Schnittstellen zu den operativen Systemen besitzt. Operative Systeme, die Detailinformationen über Artefakte einer bestimmten Anwendungsdomäne verwalten, sind beispielsweise Application-Management-Systeme, Product-Data-Management-Systeme oder Performance-Management-Systeme.¹⁶⁵

2.5 Ziele und treibende Kräfte der Unternehmensarchitektur

Eines der primären Ziele und damit auch eine der wesentlichen treibenden Kräfte der Unternehmensarchitektur stellt die Erreichung des schon seit längerer Zeit in der Praxis wie in der Wissenschaft diskutierten IT-Business-Alignment dar.¹⁶⁶ Das IT-Business-Alignment wird allgemein als ein wichtiges Koordinationsinstrument gesehen, um organisatorische Strukturen und Informationssysteme möglichst effektiv aufeinander abzustimmen.¹⁶⁷

HENDERSON/VENKATRAMAN haben bereits vor einiger Zeit ein grundlegendes IT-Business-Alignment-Modell aufgestellt.¹⁶⁸ Der Grundgedanke dieses Modells ist die Ausrichtung der Informationstechnologie an den strategischen Zielen eines Unternehmens. Die Autoren verfolgen mit diesem Modell das Ziel, einen Ansatz für die Ausrichtung der Informationstechnologie an den betriebswirtschaftlichen Anforderungen und Zielen des Unternehmens zu bieten, um den Nutzen von IT-Investitionen zu steigern bzw. zu optimieren.

IT-Business-Alignment bedeutet allerdings nicht nur, dass die Informationstechnologie und die Geschäftsstrategie eines Unternehmens aufeinander abzustimmen sind. Vielmehr kommt es im Hinblick auf die Integration auch darauf an, die eingesetzten Technologien, die Organisationsstrukturen und Abläufe sowie die benötigten Qualifikationen und Ressourcen zu koordinieren.¹⁶⁹

Das IT-Business-Alignment wird nach Ansicht einiger Autoren nicht durch die Optimierung einzelner Komponenten oder Bereiche, sondern durch die bereichsübergreifende Koordination aller organisatorischen Schlüsselemente eines Unternehmens erreicht.¹⁷⁰ LANKHORST ET AL. sehen beispielsweise primär in der Definition der Beziehungen und Abhängigkeiten zwischen Schlüsselementen und nicht in der detaillierten Spezifikation aller Einzelkomponenten eine

¹⁶⁴ Vgl. hierzu auch *Braun/Winter* (2005b), S. 68.

¹⁶⁵ Vgl. hierzu auch *Winter/Fischer* (2006), S. 8-9.

¹⁶⁶ Vgl. z.B. *Aziz et al.* (2005), *IFEAD* (2005), *Lankhorst* (2005) und *Jonkers et al.* (2004).

¹⁶⁷ Vgl. z.B. *Henderson/Venkatraman* (1993), *Luftman et al.* (1993), *Luftman et al.* (1999) und *Sledgianowski/Luftman* (2005).

¹⁶⁸ Vgl. *Henderson/Venkatraman* (1993).

¹⁶⁹ Vgl. *Henderson/Thomas* (1992), S. 72.

¹⁷⁰ Vgl. z.B. *Nadler et al.* (1992), *Lankhorst* (2005), S. 6 und *Schekkerman* (2004), S. 30.

wichtige Voraussetzung für das erfolgreiche IT-Business-Alignment.¹⁷¹ Jede Organisation sollte eine klare Vorstellung über ihre Strukturen, Produkte, Prozesse, Technologien etc. und deren Zusammenhänge haben sowie über die Beziehungen zu ihrer Umwelt, wie beispielsweise Kunden, Lieferanten und Konkurrenten. Gerade grosse Organisationen mit komplexen Strukturen sind auf die strukturierte Gestaltung und Dokumentation ihrer Architektur angewiesen, da eine Veränderung der Geschäftsstrategie und Unternehmensziele erhebliche Auswirkungen auf alle Bereiche des Unternehmens hat, wie beispielsweise auf die Organisationsstrukturen, Geschäftsprozesse, Informationssysteme sowie die technische Infrastruktur.¹⁷² Ein passendes Beispiel zur Veranschaulichung dieses Sachverhaltes ist die Einführung eines neuen Produktes bzw. einer neuen Leistung. Diese erfordert die Änderung bestehender oder die Definition neuer Geschäftsprozesse, eventuell die Einstellung zusätzlicher Mitarbeiter, die Änderung oder Neueinführung unterstützender Applikationen sowie den Ausbau der technischen Infrastruktur aufgrund der zunehmenden Belastung durch die neuen Applikationen. Diese Auswirkungen auf die Unternehmensstrukturen sollten idealerweise vor der Einführung eines neuen Produktes erörtert und abgeschätzt werden können. Darüber hinaus sind bei einer solchen Entscheidung viele unterschiedliche Personen sowohl innerhalb als auch ausserhalb des Unternehmens, von der Unternehmensführung bis zum Softwareentwickler, mit einzubeziehen. Jede in die Entscheidungsfindung eingebundene Person benötigt spezifische Informationen in einer ihr zugänglichen und verständlichen Form, um die Auswirkungen einer solch weit reichenden Entscheidung beurteilen zu können. Es ist allerdings sehr schwierig, einen Gesamtüberblick über die Veränderungen und deren Auswirkungen zu erhalten und Entscheidungsträger sowie Prozessgestalter und Softwareentwickler mit den für sie relevanten Informationen zu versorgen.

Die Definition und Dokumentation einer Unternehmensarchitektur stellt daher ein wichtiges Instrument dar, um die Kommunikation und Diskussion zwischen den verantwortlichen Personen über die wesentlichen Elemente sowie deren Beziehungen und Funktionsweisen innerhalb des Unternehmens zu ermöglichen. Sie stellt konsistente Informationen über das Unternehmen an einem zentralen Ort für Mitarbeiter aus unterschiedlichen Geschäftsbereichen zur Verfügung. Dadurch trägt sie entscheidend dazu bei, dass Mitarbeiter aus unterschiedlichen Bereichen eine gemeinsame Sicht auf das Unternehmen haben und eine gemeinsame Sprache und ein gemeinsames Verständnis über das Unternehmen und dessen Strukturen aufbauen können. Mit Hilfe einer dokumentierten Unternehmensarchitektur können Führungskräfte zudem besser beurteilen, wie gut die Ressourcen für die Erstellung von Leistungen bzw. Produkten eingesetzt werden und welche Technologien benötigt werden, um die Kundenprozesse besser unterstützen zu können. Sie erleichtert somit auch die Reaktion auf Veränderungen und das Treffen der richtigen Entscheidungen. Zudem reduziert eine für das gesamte Unterneh-

¹⁷¹ Vgl. *Lankhorst* (2005), S. 6-7.

¹⁷² Vgl. *Jonkers et al.* (2004), S. 257.

men definierte Architektur redundante und inkonsistente Informationen über das Unternehmen und trägt somit dazu bei, dass der Nutzen zukünftiger IT-Investitionen wesentlich besser beurteilt werden kann.¹⁷³

Die internen Treiber der Unternehmensarchitektur lassen sich entsprechend den obigen Ausführungen wie folgt zusammenfassen:¹⁷⁴

- Änderungen der Strategie eines Unternehmens können besser beurteilt und umgesetzt werden. (*Strategieumsetzung*)
- Auswirkungen von Veränderungen an der Unternehmensstruktur können besser abgeschätzt werden. (*Veränderung/Auswirkung*)
- Die Kommunikation zwischen Mitarbeitern aus unterschiedlichen Unternehmensbereichen kann verbessert werden. (*Kommunikationsbasis/IT-Business-Alignment*)
- Die Abstimmung zwischen den Geschäftsanforderungen und der Informationstechnologie führt zu niedrigeren Kosten und einer Steigerung des Nutzens von IT-Investitionen. (*Kosten/Nutzen*)
- Die Qualität der erzeugten und angebotenen Produkte bzw. Leistungen kann gesteigert werden. (*Qualität*)
- Die Einführung neuer Leistungen bzw. Produkte kann reduziert werden. (*Time-to-Market*)
- Die zuvor genannten Punkte können letztendlich eine gesteigerte Kundenzufriedenheit bewirken. (*Kundenzufriedenheit*)

Neben den internen Treibern existieren auch externe Einflussfaktoren, die die systematische Gestaltung und Dokumentation der Unternehmensarchitektur erfordern. In einer zunehmend vernetzten Welt kann ein Unternehmen sich nicht mehr nur auf die eigenen Strukturen und Prozesse konzentrieren. Verteilte Strukturen sind nicht nur ein Phänomen von Informationssystemen und Softwarekomponenten.¹⁷⁵ Auch Geschäftsprozesse erstrecken sich zunehmend über die Grenzen einzelner Geschäftseinheiten oder sogar Unternehmen hinweg. Zudem basieren neue Geschäftsmodelle immer häufiger auf Kooperationen oder Kooptionen. Die Unternehmensarchitektur spielt dabei eine wichtige Rolle, da sie die Beziehungen und Leistungsverflechtungen mit Kunden, Lieferanten und anderen Partnern im Geschäftsnetzwerk abbildet.

¹⁷³ Vgl. *Schekkerman* (2004), S. 14.

¹⁷⁴ Vgl. hierzu auch *Lankhorst* (2005), S. 10 und *Infosys* (2005).

¹⁷⁵ Vgl. *Österle et al.* (2001).

Ein aktuelles Beispiel für die Bedeutung einer klar definierten Unternehmensarchitektur stellt das Outsourcing von beispielsweise Geschäftsprozessen oder der IT-Infrastruktur dar.¹⁷⁶ Für den Erfolg eines jeden Outsourcing-Projektes ist es entscheidend, dass alle beteiligten Partner einen klaren Einblick in die Abläufe und Verantwortlichkeiten haben und der Leistungsaustausch sowie die Schnittstellen zwischen den Partnern klar definiert sind.¹⁷⁷

Ein weiterer wichtiger externer Treiber sind Regulationen. Externe Regulationen zwingen oftmals Unternehmen sowie staatliche Organisationen dazu, ihre Abläufe und Strukturen zu dokumentieren, um nachweisen zu können, dass sie bestimmte gesetzliche Vorgaben erfüllen. Aktuelle Beispiele für solche externen Regulationen sind der Clinger-Cohen Act¹⁷⁸, Basel II¹⁷⁹ und der Sarbanes-Oxley-Act¹⁸⁰. Die Übereinstimmung mit diesen bzw. die Einhaltung (Compliance) dieser Gesetze kann ohne eine klar dokumentierte Unternehmensarchitektur nur schwer gewährleistet werden.¹⁸¹

Die externen Treiber einer Unternehmensarchitektur lassen sich wie folgt zusammenfassen:

- Die Beziehungen zu Kunden, Lieferanten und anderen Partnern können besser gepflegt und kontrolliert werden. (*Beziehungen*)
- Die Einhaltung externer Regulationen kann besser kontrolliert werden. (*Regulationen/Compliance*)
- Unternehmensübergreifende Geschäftsprozesse lassen sich einfacher realisieren und kontrollieren. (*Vernetzung*)

Trotz der vielen zuvor genannten Faktoren, die für die Einführung einer Unternehmensarchitektur sprechen bzw. diese erfordern, bestehen bei der praktischen Umsetzung in Unternehmen noch erhebliche Probleme. Diese liegen einerseits in der Definition des adäquaten Detaillierungsgrades und andererseits im Fehlen eines einheitlichen, für Mitarbeiter aus unterschiedlichen Unternehmensbereichen verständlichen Modellierungskonzeptes begründet.¹⁸²

Eine detaillierte Dokumentation der Schlüsselemente aller Bereiche eines Unternehmens ist eine aufwendige und zeitintensive Aufgabe. Dadurch rückt das eigentliche Ziel der Unternehmensarchitektur, nämlich das Alignment, in den Hintergrund. Dies kann dazu führen, dass

¹⁷⁶ Ein Überblick sowie eine Klassifikation aktueller Phänomene des Outsourcing findet sich z.B. in Braun/Winter (2005a).

¹⁷⁷ Vgl. z.B. Masak (2005), S. 14.

¹⁷⁸ Vgl. CIO-Council (1996).

¹⁷⁹ Vgl. z.B. Bundesbank (2004).

¹⁸⁰ Vgl. Sarbanes-Oxley (2002).

¹⁸¹ DINNER/KOLBER haben in ihrer Arbeit beispielsweise einige Anforderungen von Basel II und des Sarbanes-Oxley-Act in den Bezugsrahmen von Zachman eingeordnet. Vgl. Dinner/Kolber (2005).

¹⁸² Im Rahmen eines Workshops des Kompetenzzentrums „Integration Factory“ (CC IF) gemeinsam mit Praxisvertretern aus sechs verschiedenen Unternehmen der Finanzbranche zum Thema Unternehmensarchitekturen am 28.06.2005 wurden diese beiden Punkte als wesentliche Gründe für die Probleme bei der praktischen Umsetzung einer Unternehmensarchitektur genannt.

die wesentlichen und für den Erfolg einer Unternehmensarchitektur wichtigen ebenen- und bereichsübergreifenden Zusammenhänge innerhalb des Unternehmens zu wenig berücksichtigt werden. Die detaillierte Dokumentation der einzelnen Bereiche sollte daher eher ausserhalb des Betrachtungsraums der Unternehmensarchitektur erfolgen. Der Detaillierungsgrad bei der Definition der Elemente der Unternehmensarchitektur sollte so gewählt sein, dass vor allem die Zusammenhänge zwischen den Schlüsselementen der einzelnen Ebenen und Bereiche innerhalb einer Organisation im Sinne eines Alignment möglichst gut und verständlich abgebildet werden. Zusätzlich können die aggregierten Elemente der Unternehmensarchitektur in domänenspezifischen Architekturen detaillierter abgebildet werden, was oftmals schon der Fall ist.

Des Weiteren sollten die Elemente der Unternehmensarchitektur möglichst eindeutig und verständlich beschrieben sein, damit Mitarbeiter aus unterschiedlichen Unternehmensbereichen, wie beispielsweise Softwareentwickler, Anwender und Manager, diese als gemeinsame Kommunikationsbasis verwenden können. Bisher verwendet allerdings in den meisten Unternehmen jeder Unternehmensbereich seine eigenen Beschreibungssprachen und -konventionen.¹⁸³ Es existiert bisher keine Standardsprache zur exakten Beschreibung von Unternehmensarchitekturen, die für alle Bereiche eingesetzt werden kann.¹⁸⁴ In manchen Unternehmensbereichen erfolgt die Dokumentation von Strukturen in Fliesstext oder in Form von nicht formal spezifizierten Diagrammen. In anderen werden die Strukturen dagegen sehr detailliert mit formalen oder semi-formalen Modellierungssprachen abgebildet, die für Mitarbeiter anderer Unternehmensbereiche nur schwer verständlich sind. Dies führt zu Missverständnissen und erschwert die Abstimmung zwischen Architekten, Softwareentwicklern und Mitarbeitern aus den Fachabteilungen. Aufgrund der Heterogenität der für die Abbildung der Unternehmensarchitektur verwendeten Modellierungssprachen und -techniken können die Zusammenhänge zwischen den unterschiedlichen Domänen nur schwer identifiziert werden. Für die Kommunikation zwischen unterschiedlichen Unternehmensbereichen und somit auch für die Realisierung des IT-Business-Alignment ist aber eine klare und einheitliche Sicht auf die Zusammenhänge eine grundlegende Voraussetzung.

Ein Ansatz zur Gestaltung der Unternehmensarchitektur sollte daher die Identifikation der Abhängigkeiten zwischen Schlüsselementen unterschiedlicher Bereiche bzw. Domänen adressieren. Die Modelle der Methode sollten die Schlüsselemente und deren Beziehungen innerhalb einer Domäne in aggregierter und auch für Nicht-Domänenexperten verständlicher Form abbilden. Ferner sollten die Beziehungen zwischen den Schlüsselementen der unterschiedlichen Domänen klar definiert sein, um mögliche Auswirkungen von Veränderungen in

¹⁸³ Vgl. hierzu z.B. *IFEAD* (2005), S. 30-31. Im Rahmen des zuvor genannten Workshops hat sich zudem gezeigt, dass innerhalb jedes Unternehmens eine Vielzahl von Modellierungssprachen zum Einsatz kommt, die nicht durch ein übergreifendes Konzept miteinander integriert sind. Dadurch können Inkonsistenzen zwischen den in verschiedenen Unternehmensbereichen erzeugten Ergebnissen entstehen.

¹⁸⁴ Vgl. *Lankhorst* (2005), S. 83.

einer Domäne nachvollziehen zu können. Damit die einzelnen Modelle, Elemente und Beziehungen unmissverständlich interpretiert werden können, ist die formale oder zumindest semi-formale Spezifikation der Modelle anhand eines zugrunde liegenden Metamodells notwendig. Im Hinblick auf eine Werkzeugunterstützung können so die Modelle der Methode auch erst implementiert und auf deren Grundlage anschliessend Analysen, wie beispielsweise Impact-Analysen oder Konsistenzprüfungen, durchgeführt werden.

2.6 Konsequenzen für das weitere Vorgehen

Basierend auf den zuvor beschriebenen Grundlagen sowie den Zielen und treibenden Kräften der Unternehmensarchitektur (Abschnitt 2.1 bis 2.5) können wesentliche Anforderungen an die weitere Vorgehensweise formuliert werden.¹⁸⁵ Diese sind unterteilt in Anforderungen an den zu entwickelnden Ansatz sowie in Anforderungen an die Werkzeugunterstützung.

2.6.1 Anforderungen an einen Ansatz zur Modellierung der Unternehmensarchitektur

Im Folgenden werden wesentliche Anforderungen zusammengefasst, die für das weitere Vorgehen zweckmässig sind. Ein Ansatz zur Modellierung der Unternehmensarchitektur muss folgende Anforderungen erfüllen:

- Er muss alle Schlüsselemente, die im Hinblick auf strategische Fragestellungen im Schnittbereich zwischen Fachbereich und IT-Bereich relevant sind, in aggregierter Form abbilden¹⁸⁶ (*Ganzheitlichkeit/Aggregation*).
- Eine Einteilung in Architekturebenen und Sichten ist hilfreich, um unterschiedliche Aspekte in unterschiedlichen Modellen darzustellen und somit die Komplexität beherrschbar zu machen (*Strukturierung/Komplexitätsreduktion*).
- Der Ansatz muss die Abhängigkeiten zwischen den Schlüsselementen abbilden sowie die Konsistenz zwischen den einzelnen Architekturebenen und Sichten gewährleisten (*Konsistenz*).
- Die verwendeten Konzepte müssen anschaulich und verständlich sein (*Anschaulichkeit/Verständlichkeit*). Dies hängt vor allem von der graphischen Notation ab. Die graphische Notation sollte daher eine deutliche Abbildung von Sprachkonzepten auf graphische Symbole bieten, zudem sollten die Symbole semantisch nicht überladen sein.¹⁸⁷ Bei der Modellierung betrieblicher Informationssysteme wird die Anschaulichkeit vor allem durch

¹⁸⁵ Die Ableitung der Anforderungen erfolgt in Anlehnung an *Kremer* (2004), S. 33.

¹⁸⁶ Vgl. hierzu auch die in *Winter/Fischer* (2006) als wesentlich identifizierten Elemente einer Unternehmensarchitektur.

¹⁸⁷ Vgl. z.B. *Frank/Van Laak* (2003), S. 29f.

Sprachkonzepte gefördert, die vertraute Darstellungen spezieller Sichten, wie z.B. von Geschäftsprozessen oder Organisationsstrukturen, ermöglichen.

- Die für den jeweiligen Modellierungszweck erforderlichen Aspekte des abzubildenden Sachverhaltes müssen in einer sinnvollen Detaillierung und Formalisierung dargestellt werden (*Angemessenheit/Mächtigkeit*).
- Der Ansatz muss erweiterbar sein, da sich die Vorstellungen über die abzubildenden Sachverhalte im Zeitverlauf ändern können (*Flexibilität*).
- Die verwendeten Konzepte müssen formal spezifiziert sein, damit sie unmissverständlich interpretiert werden können (*Formalisierung*). Zudem erfordert die ingenieurmässige Entwicklung von Informationssystemen eine (semi-)formale Definition der Semantik konzeptueller Modelle mittels entsprechender Metamodelle. Im Idealfall bedeutet dies, dass aus dem Metamodell alle zulässigen Modelle generiert werden können und dass für jedes Modell entschieden werden kann, ob es zulässig ist.
- Die systematische Gestaltung der Unternehmensarchitektur muss durch eine umfassende Berücksichtigung aller Bestandteile des Methoden-Engineering unterstützt werden (*Methodenelemente*).

Neben den zuvor genannten Anforderungen an einen Ansatz zur Modellierung der Unternehmensarchitektur lassen sich auch wesentliche Anforderungen an die softwaretechnische Werkzeugunterstützung ableiten.

2.6.2 Anforderungen an die Werkzeugunterstützung

Die Anforderungen an die Werkzeugunterstützung können aus unterschiedlichen Perspektiven betrachtet werden. Es kann nach den Perspektiven „Metamodellierung und Mechanismen“, „Organisation“ und „Technologie“ differenziert werden. Die Anforderungen der Perspektive „Metamodellierung und Mechanismen“ garantieren eine entsprechende Unterstützung der Modellierungstechniken und entsprechender zielgerichteter Mechanismen, wie beispielsweise der Simulation, der Prüfung von Abhängigkeiten und Konsistenz sowie der Referenzmodellierung. Vorgehensmodelle, Richtlinien und Benutzerfreundlichkeit sind die wesentlichen Kriterien der organisatorischen Perspektive. Die technologische Perspektive betrachtet Kriterien wie Anpassbarkeit, Skalierbarkeit, Multi-Client-Fähigkeit und offene Architekturen.

Die Anforderungen der Perspektive „Metamodellierung und Mechanismen“ bilden die Grundlage der Umsetzung eines Metamodellierungskonzeptes. Sie stehen daher mit allen anderen Anforderungen in Verbindung.

- Aufgrund einer Änderung der Ziele der verwendeten Methode oder einer veränderten Projektsituation muss eine einfache Erweiterbarkeit und Anpassbarkeit des zugrunde liegenden

den Metamodells ohne Programmieraufwand möglich sein.¹⁸⁸ (*Erweiterbarkeit/Anpassbarkeit*)

- Die Verbindung oder Integration von Metamodellen mit unterschiedlichen Anwendungsbereichen sollte ohne grossen Programmieraufwand möglich sein, um beispielsweise eine integrierte Prozess-, Produkt- und Informationssystem-Modellierung zu ermöglichen. (*Integration*)
- Es müssen entsprechende Mechanismen für die Überprüfung der Konsistenz zwischen den erstellten Modellen und deren Abhängigkeiten implementiert sein. (*Konsistenz*)
- Bei der Änderung des zugrunde liegenden Metamodells muss eine einfache Migration von Modellen, die noch auf einer alten Version des Metamodells basieren, möglich sein.¹⁸⁹ (*Migration*)
- Es müssen Mechanismen für die Klassifikation und Kategorisierung sowie Suchmuster für die Wiederverwendung von Modellen bereitgestellt werden. (*Wiederverwendung*)
- Eine Datenbank mit Import-/Export-Funktionalitäten für Metamodelle, Modelle und Methodenfragmente ist ein zentraler Bestandteil eines Metamodellierungswerkzeuges.¹⁹⁰ (*Import/Export*)
- Es müssen Funktionen für die Generierung unterschiedlicher Sichten auf Modelle, Objekte und Attribute zur Verfügung stehen, um dadurch generelle Mechanismen für die Modellanalyse zu ermöglichen.¹⁹¹ (*Analyse*)
- Funktionalitäten für die Automatisierung bestimmter Modellierungsaktivitäten erleichtern die Erstellung von Modellen.¹⁹² Beispiele für solche Funktionalitäten sind die Modellgenerierung basierend auf Referenzmodellen oder die Sichtengenerierung basierend auf importierten Modellinformationen. (*Automatisierung*)

In Bezug auf die Erstellung, Verwaltung und Verwendung von Methoden und deren Bestandteilen können unterschiedliche Rollen und Aktivitäten differenziert werden, die jeweils unterschiedliche Anforderungen an die Modellierungsplattform stellen.¹⁹³ Der Methodenkonstrukteur ist verantwortlich für die Einhaltung der Konsistenz der gesamten Methode. Der Sprachenentwickler definiert die Syntax, Semantik und Notation der verwendeten Modellierungssprachen. Der Prozessentwickler ist verantwortlich für die Vorgehensweise und deren Aktivitäten. Der Werkzeugentwickler bildet die Methode und deren Komponenten in dem Werk-

¹⁸⁸ Vgl. Junginger et al. (2000).

¹⁸⁹ Vgl. Kühn et al. (2004).

¹⁹⁰ Vgl. Kühn et al. (2003).

¹⁹¹ Vgl. Kühn et al. (1999).

¹⁹² Vgl. Kühn et al. (2003).

¹⁹³ Vgl. im Folgenden Karagiannis/Kühn (2002).

zeug ab, und der Anwender setzt die Methode und das Werkzeug letztendlich ein. Die organisatorische Perspektive umfasst folgende wesentlichen Anforderungen:

- Die Integration mit Projektmanagement-Werkzeugen ermöglicht die Einhaltung und Planung der von der verwendeten Methode definierten Vorgehensweisen und Ergebnisse. (*Planung*)
- Das Werkzeug muss unterschiedliche, vom jeweiligen Anwendungsbereich abhängige Versionierungskonzepte unterstützen, wie beispielsweise zeitliche Versionierung, modellbasierte Versionierung oder Check-In/Check-Out-Mechanismen. (*Versionierung*)
- Anwendungskonzepte werden durch Rollenmodelle und Verantwortlichkeiten unterstützt. Diese müssen auch in dem Rechte- bzw. Autorisierungskonzept des Werkzeuges implementiert sein. (*Rechtekonzept*)
- Eine intuitiv bedienbare Benutzeroberfläche erleichtert die Einarbeitung von Anwendern. Unterschiedliche Sichten, abhängig von den Kenntnissen der Anwender, sollten ohne Programmieraufwand flexibel generiert werden können. (*Benutzerfreundlichkeit*)
- Wenn die Methode Best Practices beinhaltet, sollte das Werkzeug das Vorgehen zur Anwendung und Anpassung dieser Best Practices unterstützen. (*Referenzmodellierung*)

Die Anforderungen der technologischen Perspektive lassen sich wie folgt zusammenfassen:¹⁹⁴

- Es sollte eine verteilte Modellierung und Ergebnisdokumentation möglich sein. Die Ergebnisdokumentation könnte beispielsweise in Form von Web-Seiten, auf die zentral im Intranet zugegriffen werden kann, erfolgen. (*Verteilung*)
- Das Werkzeug muss Schnittstellen zu den operativen Systemen anbieten, um die aggregierten Informationen der Unternehmensarchitektur mit den Detailinformationen der operativen Systeme verknüpfen zu können. (*Schnittstellen*)
- Das Werkzeug sollte für unterschiedliche Architekturen, wie Stand-alone, Client-Server und Application Hosting, konfigurierbar sein. Eine offene, serviceorientierte Architektur ermöglicht die Integration zusätzlicher Modellierungswerkzeuge, Standard-Interfaces sowie wieder verwendbare Komponenten. (*Architektur*)
- Das Werkzeug sollte plattformunabhängig einsetzbar sein. (*Plattformunabhängigkeit*)

Das folgende Kapitel 3 vergleicht bestehende Ansätze anhand der in Abschnitt 2.6.1 genannten Anforderungen an einen Ansatz zur Modellierung der Unternehmensarchitektur. Die in diesem Abschnitt zusammengefassten Anforderungen an die Werkzeugunterstützung werden bei der Wahl einer geeigneten Metamodellierungsplattform sowie der anschließenden Konstruktion des Prototyps berücksichtigt.

¹⁹⁴ Vgl. im Folgenden auch *Kühn/Bayer (2004)* und *Karagiannis/Kühn (2002)*.

3 Bisherige relevante Ansätze

Das Ziel des Dissertationsvorhabens ist die Entwicklung eines Ansatzes sowie die prototypische Konstruktion eines Werkzeuges zur Modellierung der Unternehmensarchitektur, basierend auf einer bestehenden Methode. Der Vorschlag eines neuen Ansatzes in diesem Bereich ist nur dann sinnvoll, wenn keine Methode vorhanden ist oder existierende Methoden in Bezug auf die an sie gestellten Anforderungen unvollständig sind. Auf Grundlage der in Abschnitt 2.6 abgeleiteten Kriterien werden in diesem Kapitel existierende Ansätze zur Modellierung der Unternehmensarchitektur anhand des inhaltlichen Beitrages (inhaltliche Charakteristika) sowie der definierten Methodenelemente (strukturelle Charakteristika) beurteilt.¹⁹⁵ Im Hinblick auf die Entwicklung eines eigenen ebenenübergreifenden Metamodells liegt der Fokus bei der Beschreibung der verwandten Ansätze auf dem Methodenelement „Metamodell“. Die folgende Tabelle 4 gibt einen Überblick über die angewendeten Bewertungskriterien.

Kriterium	Beschreibung
Inhaltliche Charakteristika	
Aggregation/ Ganzheitlichkeit	Abbildung aller Schlüsselemente und deren Beziehungen in aggregierter Form, die für strategische Entscheidungen im Schnittbereich zwischen Fachbereich und IT-Abteilung relevant sind
Strukturierung	Strukturierung der Schlüsselemente nach unterschiedlichen Architekturebenen und Sichten
Konsistenz	Abbildung der Abhängigkeiten zwischen den Schlüsselementen sowie Gewährleistung der Konsistenz zwischen den einzelnen Architekturebenen und Sichten
Verständlichkeit	Deutliche Abbildung von Sprachkonzepten auf grafische Symbole, mit denen der Anwender oder Betrachter der Modelle vertraut ist
Angemessenheit	Darstellung der abzubildenden Sachverhalte in einer sinnvollen Detaillierung und Formalisierung
Flexibilität	Erweiterbarkeit der Metamodelle im Hinblick auf eine mögliche Änderung der abzubildenden Sachverhalte im Zeitverlauf
Formalisierung	(Semi-)formale Definition der Semantik konzeptueller Modelle mittels entsprechender Metamodelle
Strukturelle Charakteristika	
Methodenelemente	Darstellung von Methodenbestandteilen im Sinne des Methoden-Engineering, wie z.B. Vorgehensmodelle, Metamodelle, Techniken, Rollenmodelle und Ergebnisdokumente

Tabelle 4: Bewertungskriterien

Im folgenden Abschnitt erfolgt zunächst die Auswahl der bestehenden Ansätze anhand definierter Kriterien. Diese werden anschliessend beschrieben und anhand der oben genannten

¹⁹⁵ In Anlehnung an *Kremer (2004)* und *Legner (1999)*.

Kriterien bewertet. Aus der Analyse bestehender Ansätze werden Konsequenzen für den eigenen Ansatz abgeleitet.

3.1 Auswahl der Ansätze

Die Auswahl der zu betrachtenden Ansätze erfolgt anhand von drei wesentlichen Kriterien: Ein Kriterium stellt die Anzahl der aktuell am Markt verfügbaren Software-Werkzeuge dar, die einen bestimmten Ansatz unterstützen.¹⁹⁶ Darüber hinaus wird betrachtet, wie gross die Anzahl und Verfügbarkeit von Veröffentlichungen zu einem bestimmten Ansatz sind und wie gross seine wissenschaftliche Bedeutung ist. Letztere kann beispielsweise daran gemessen werden, wie häufig dieser in Publikationen referenziert wird. Da das Dissertationsvorhaben die umfassende Modellierung der Unternehmensarchitektur fokussiert, wird ausserdem vorausgesetzt, dass der Ansatz das Unternehmen in mehrere Gestaltungsebenen und Sichten einteilt und diese konsistent abbildet.

Der Vergleich basiert auf der Auswertung von Publikationen (Bücher, Zeitschriftenartikel, Konferenzbeiträge) zu den einzelnen Ansätzen. Die Analyse wurde im Oktober 2005 abgeschlossen. Sie umfasst die in Tabelle 5 aufgeführten Ansätze.

Weitere Ansätze, die im Rahmen des Vergleichs betrachtet wurden, sind die Model Driven Architecture (MDA)¹⁹⁷, die Informationssystem-Architektur (ISA) nach KRCMAR¹⁹⁸ sowie das Treasury Enterprise Architecture Framework (TEAF) und das Department of Defense Architecture Framework (DoDAF). Diese werden allerdings aus folgenden Gründen nicht in die detaillierte Beschreibung relevanter Ansätze der vorliegenden Arbeit aufgenommen:

- MDA ist zu sehr auf softwareentwicklungsrelevante Aspekte fokussiert, wie beispielsweise die plattformunabhängige Modellierung von Softwaresystemen und die automatische Code-Generierung aus UML-Modellen. Die OMG hat zwar durch die Einführung des sogenannten Computer Independent Model (CIM) den Fokus von MDA um fachliche Aspekte erweitert.¹⁹⁹ Im traditionellen Sinne stellt dieses allerdings lediglich die Fachspezifikation eines Softwaresystems dar. Der MDA-Ansatz bietet daher zu wenig Konzepte im Sinne der Unternehmensarchitektur, wie sie in der vorliegenden Arbeit definiert ist.
- ISA bietet keine Metamodelle für die Definition der einzelnen Sichten und ist daher für die Entwicklung des eigenen Ansatzes lediglich im Hinblick auf die Einteilung in Archi-

¹⁹⁶ Eine Übersicht über gängige Ansätze und deren Werkzeugunterstützung bieten z.B. *James* (2004a) und *Schekkerman* (2004).

¹⁹⁷ Vgl. hierzu z.B. *OMG* (2003a) und *Völter* (2003).

¹⁹⁸ Vgl. *Krcmar* (1990).

¹⁹⁹ Vgl. *Masak* (2005), S. 252.

tekturebenen und Sichten interessant. Eine entsprechende Analyse von ISA wurde bereits in der Arbeit von WINTER durchgeführt.²⁰⁰

- Im Bereich der Bezugsrahmen (Frameworks) aus der Praxis wurden lediglich FEAF (Federal Enterprise Architecture Framework), TOGAF (The Open Group Architecture Framework) und das Zachman Framework ausgewählt, da diese einer Studie von IFEAD (Institute For Enterprise Architecture Developments) zufolge in der Praxis am weitesten verbreitet sind.²⁰¹
- Zudem basieren einige weitere Bezugsrahmen, wie beispielsweise DoDAF und TEAF, auf dem Zachman Framework und sind auf einander ausgerichtet.

	Ansatz	Kurzbeschreibung
Methoden aus der Wissenschaft	Architektur integrierter Informationssysteme (ARIS) Primäre Quellen: <i>Scheer</i> (1998), <i>Scheer</i> (2001), <i>Scheer/Jost</i> (2002), <i>Scheer/Thomas</i> (2005)	ARIS ist auf den Geschäftsprozess ausgerichtet und verfolgt den Ansatz, Informationssystem und Geschäftsprozesse integrativ zu betrachten.
	Semantisches Objektmodell (SOM) Primäre Quellen: <i>Ferstl/Sinz</i> (1995), <i>Ferstl/Sinz</i> (1998)	SOM wurde mit dem Ziel der geschäftsorientierten Gestaltung von Informationssystemen entwickelt.
	Multi-Perspective Enterprise Modeling (MEMO) Primäre Quellen: <i>Frank</i> (1994), <i>Frank</i> (1998b), <i>Frank</i> (2002)	MEMO ist eine Methode zur multiperspektivischen Unternehmensmodellierung. Die Methode bietet einen Bezugsrahmen, der die Strukturierung eines Unternehmens bzw. einer Organisation auf einem hohen Abstraktionsniveau ermöglicht.
Bezugsrahmen aus der Praxis	Zachman Framework Primäre Quellen: <i>Zachman</i> (1987), <i>Zachman/Sowa</i> (1992)	Das Zachman Framework stellt einen Bezugsrahmen für die Klassifizierung von Konzepten zur Verfügung, um Objekte der realen Welt mit den entsprechenden Repräsentationen in den Informationssystemen zu verbinden.
	Federal Enterprise Architecture Framework (FEAF) Primäre Quellen: <i>FEA</i> (1999), <i>FEA</i> (2001)	FEAF ist ein Bezugsrahmen für die gemeinsame Entwicklung von Prozessen und Informationsstrukturen zwischen US-staatlichen Organisationen.
	The Open Group Architecture Framework (TOGAF) Primäre Quellen: <i>Opengroup</i> (2002)	TOGAF ist ein Industriestandard für die Entwicklung von IT-Architekturen. Die aktuelle Version wurde im Sinne der Unternehmensarchitektur um weitere Ebenen erweitert.

Tabelle 5: Übersicht der analysierten Ansätze

Im folgenden Abschnitt werden die ausgewählten Ansätze analysiert und hinsichtlich der in Tabelle 4 aufgeführten Kriterien bewertet.

²⁰⁰ Vgl. dazu *Winter* (2003d).

²⁰¹ Vgl. *IFEAD* (2005), S. 28.

3.2 Diskussion ausgewählter Methoden aus der Wissenschaft

Methoden zur Modellierung der Unternehmensarchitektur, die vor allem im deutschsprachigen Raum von Bedeutung sind und im universitären Umfeld entwickelt wurden, sind das Semantische Objektmodell (SOM), Multi-Perspective Enterprise Modeling (MEMO) sowie die Architektur Integrierter Informationssysteme (ARIS). Für alle diese drei Methoden wurden Werkzeuge erstellt, es ist umfangreiche Literatur über sie vorhanden, die frei verfügbar ist, und sie werden in anderen Arbeiten häufig referenziert. Ausserdem streben alle Methoden eine umfassende Unternehmensmodellierung an. Um die Vergleichbarkeit der Methoden sicherzustellen, werden diese im Folgenden einerseits in Bezug auf die Einteilung in Modellierungsebenen und Sichten sowie andererseits anhand der in Abschnitt 2.1 genannten wesentlichen Methodenbestandteile (Metamodell, Vorgehensmodell, Techniken, Ergebnisdokumente und Rollen) beschrieben. Die Beurteilung der Methoden erfolgt zudem anhand der in Abschnitt 2.6 definierten inhaltlichen Charakteristika. Im Hinblick auf den eigenen zu entwickelnden Ansatz werden vor allem die Metamodelle der einzelnen Methoden detailliert betrachtet.

3.2.1 Architektur integrierter Informationssysteme (ARIS)

3.2.1.1 Übersicht und Architekturebenen

Die Architektur integrierter Informationssysteme (ARIS) ist auf den Geschäftsprozess ausgerichtet. Ein Geschäftsprozess wird im ARIS-Konzept als zusammengehörende Abfolge von Unternehmensverrichtungen zum Zweck einer Leistungserstellung definiert.²⁰² ARIS verfolgt den Ansatz, Informationssystem und Geschäftsprozesse integrativ zu betrachten.²⁰³ Dabei werden die folgenden fünf Sichten auf das Unternehmen unterschieden:²⁰⁴

Die **Funktionssicht** legt die Funktionen (oftmals werden synonym die Begriffe Vorgang, Aktivität, Tätigkeit oder Aufgabe verwendet) fest, die ein Unternehmen effizient ausführen muss. Sie werden meist in Verbindung mit Daten beschrieben, da sie Input-Daten zu Output-Daten transformieren. Eine Funktion unterstützt eines oder mehrere Ziele, welche hierarchisch gegliedert und netzartig untereinander verflochten sein können. Daneben wird in der Funktionssicht die Anwendungssoftware abgebildet, welche die computerunterstützten Funktionen definiert.

Die **Organisationssicht** beschreibt die Aufbauorganisation, also die Organisationseinheiten mit den zwischen ihnen bestehenden Kommunikations- und Weisungsbeziehungen. Zusätzlich wird mit dem Rollenkonzept das Anforderungsprofil einer Organisationseinheit definiert.

²⁰² Vgl. *Leist* (2002), S. 14.

²⁰³ Vgl. *Winter* (2003d), S. 93.

²⁰⁴ Vgl. im Folgenden *Scheer* (1998), S. 36, *Scheer* (2001), *Scheer/Jost* (2002) und *Scheer/Thomas* (2005).

Die **Datensicht** enthält die Beschreibung der Datenobjekte, die von Funktionen manipuliert bzw. transformiert werden, sowie Nachrichten, die Funktionen auslösen oder von Funktionen erzeugt werden.

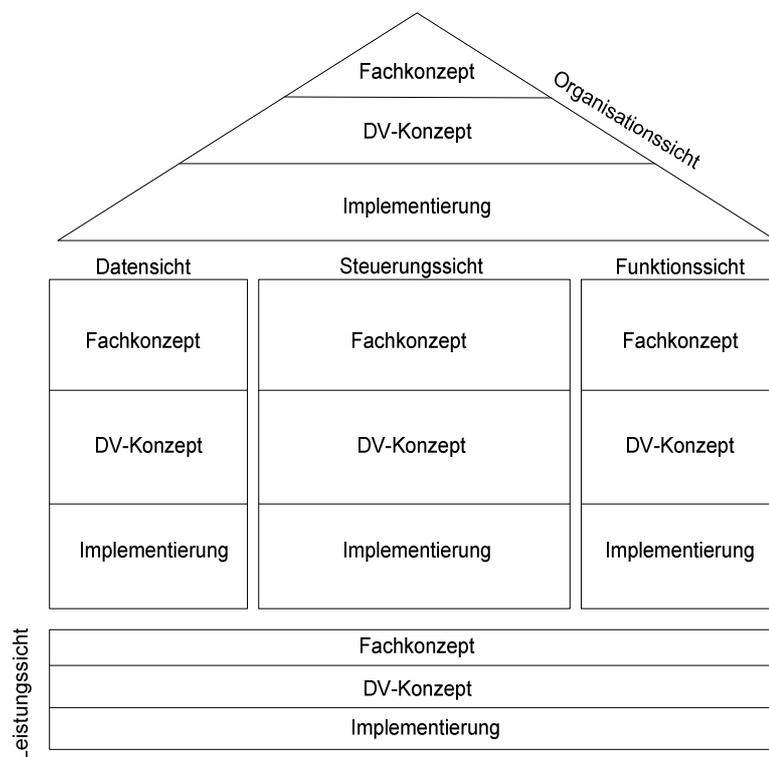


Abbildung 14: ARIS-Haus²⁰⁵

Die **Leistungssicht** umfasst alle materiellen und immateriellen Leistungen, die als Input oder Output von Funktionen festgelegt werden können.

Diese Sichten sollen möglichst unabhängig voneinander entwickelt und bearbeitet werden können, um die Komplexität beherrschbar zu machen. Um trotz der getrennten Modellierung der einzelnen Sichten die Beziehungen innerhalb des Gesamtmodells berücksichtigen zu können, wird die **Steuerungssicht** eingeführt. Diese stellt die zentrale Sicht innerhalb von ARIS dar. Sie verbindet die einzelnen Sichten miteinander und bildet somit den gesamten Geschäftsprozess ab.

Die Ebenenbildung erfolgt in ARIS erst nach der Entwicklung der Sichten (vgl. Abbildung 14). Grundlage hierfür ist ein Phasenmodell, das drei Stufen der Umsetzung vom fachlichen Konzept bis hin zur Implementierung der Informationssysteme umfasst. Auf der Ebene des **Fachkonzepts** werden betriebswirtschaftliche Konzepte abgebildet, die die fachlichen Grundlagen für die zu entwickelnden Informationssysteme beschreiben. Auf der Ebene **DV-Konzept** werden die Beschreibungen des Fachkonzepts an generelle Schnittstellen der Infor-

²⁰⁵ Vgl. Scheer (2001), S.1.

mationstechnik angepasst. Die Ebene der **Implementierung** beschreibt die programmiertechnische Umsetzung des DV-Konzepts.

Für alle Sichten bestehen Metamodelle, die zu einem integrierten Metamodell verknüpft werden können. ARIS schreibt keine bestimmten Metamodelle für die Ausgestaltung der Modellierungssprachen einzelner Ebenen vor. Exemplarisch liegen für alle Ebenen Metamodelle vor, um insbesondere für die Steuerungssicht im Sinne der Geschäftsprozessmodellierung Referenzmodelle als Konstruktionshilfen anzubieten.²⁰⁶

3.2.1.2 Metamodelle von ARIS

Im Hinblick auf die Entwicklung eines eigenen Ansatzes und Metamodells für die Modellierung der Unternehmensarchitektur werden an dieser Stelle lediglich die Metamodelle des Fachkonzepts (vgl. Abbildung 14) von ARIS beschrieben.

Die Modellierung der **strategischen Geschäftsprozesse** geht der Beschreibung der einzelnen Sichten des ARIS-Hauses (vgl. Abbildung 14) voraus. Sie umfasst das Metamodell des strategischen Geschäftsprozessmodells sowie das Metamodell des strategischen Vorgangskettendiagramms.

Für die Gestaltung der Kernprozesse werden Ansätze der strategischen Planung herangezogen, die eine Analyse der strategischen Geschäftsprozesse sowie eine Bestimmung der strategischen Soll-Konzeption beinhalten. Als Ergebnis dieser beiden Aktivitäten liegen die wesentlichen Ziele, Geschäftsfelder und die auf hoher Aggregationsstufe neu zu gestaltenden Geschäftsprozesse mit den zu beseitigenden Schwachstellen vor. Des Weiteren wird die strategische Ausrichtung bezüglich der die Geschäftsprozesse unterstützenden Informationstechnologie festgelegt.

Im Metamodell werden mehrere Ansätze der strategischen Planung integriert, wie beispielsweise das Wertschöpfungskettendiagramm nach PORTER²⁰⁷ und das Konzept der kritischen Erfolgsfaktoren nach ROCKART²⁰⁸. Zudem werden Effizienzkriterien betrachtet, Leistungsfelder differenziert und Organisationseinheiten (in Form von Organisationssegmenten dargestellt) untersucht. Abbildung 15 zeigt die wesentlichen Elemente des Metamodells des strategischen Geschäftsprozessmodells.

Die Wertschöpfungskette besteht aus mehreren Wertschöpfungsfunktionen. Eine Wertschöpfungsfunktion ist als eine mächtige Funktion definiert, die in einem Wertschöpfungsdiagramm dargestellt wird.²⁰⁹ Die Segmente der Aufbauorganisation werden durch den Metaentitätstyp „Organisationssegment“ repräsentiert. Dabei wird zwischen einer mehr horizontalen

²⁰⁶ Vgl. Schlitt (2003), S. 178.

²⁰⁷ Vgl. Porter (1999).

²⁰⁸ Vgl. Rockart (1979).

²⁰⁹ Vgl. Scheer (2001), S. 14.

Organisation (Funktionssegment) und einer mehr vertikalen Gliederung (Geschäftsprozesssegment) unterschieden.²¹⁰ Ein Leistungsfeld beschreibt wichtige Produkt- bzw. Leistungsgruppen des Unternehmens. Die Anzahl der Ausprägungen der Beziehungen zwischen Organisationssegment und Leistungsfeld sowie Organisationssegment und Wertschöpfungsfunktion zeigt, ob es sich eher um eine funktions- oder prozessorientierte Organisation handelt. Die unterschiedlichen Organisationskonzepte (funktionale Gliederung, prozessorientierte Organisation, hybride Organisation) müssen gewissen Effizienzkriterien folgen.²¹¹ Ein Organisationssegment können daher ein oder mehrere Effizienzkriterien zugeordnet werden, für das es einen bestimmten Beitrag leistet. Der abstrakte Metaentitätstyp „Effizienzkriterium“ wird in die organisatorischen Effizienzkriterien „Prozesseffizienz“, „Ressourceneffizienz“ und „Markteffizienz“ spezialisiert. Ein Geschäftsprozesssegment besteht aus mehreren Prozessen, die vom Typ „Unterstützungsprozess“ oder „Kernprozess“ sein können. Ausserdem können den Kernprozessen sowie einzelnen Wertschöpfungsfunktionen kritische Erfolgsfaktoren zugeordnet werden. Die Kernprozessvarianten spezialisieren die Kernprozesse und werden anhand der Kundengruppen segmentiert.

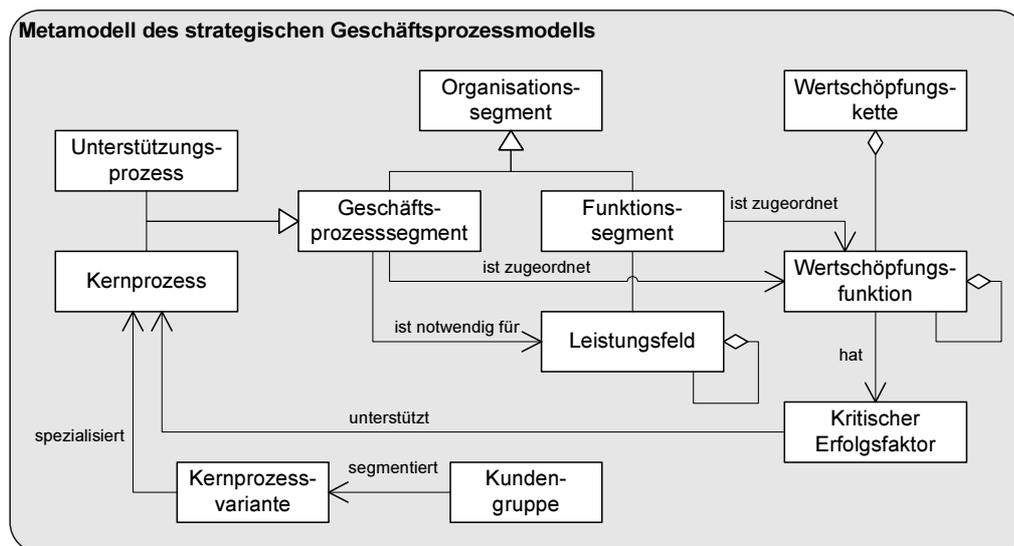


Abbildung 15: Metamodell des strategischen Geschäftsprozessmodells²¹²

Die bei der Erstellung des strategischen Geschäftsprozessmodells identifizierten Kernprozessvarianten werden mit Hilfe von strategischen Vorgangskettendiagrammen in einzelne Aufgabenschritte zerlegt.²¹³ Das Vorgangskettendiagramm unterscheidet sich nur aufgrund der Anordnung der Objekte im Diagramm sowie der eingeschränkten Anzahl an verfügbaren Metaentitätstypen von einer EPK (Ereignisgesteuerte Prozesskette). Die Kernprozesse werden lediglich so weit zerlegt, dass wesentliche strategische Mängel des gegenwärtigen Ablaufs

²¹⁰ Vgl. Scheer (2001), S. 13.

²¹¹ Vgl. Scheer (2001), S. 7.

²¹² Vgl. Scheer (2001), S. 12.

²¹³ Vgl. Scheer (2001), S. 15.

und die Vorteile des zukünftigen Ablaufs identifiziert werden können. Die Darstellung bleibt somit auf einer rein strategischen Ebene. Das Metamodell wird in Anlehnung an das von LEIST vorgeschlagene vereinfachte Metamodell der EPK dargestellt (vgl. Abbildung 16).

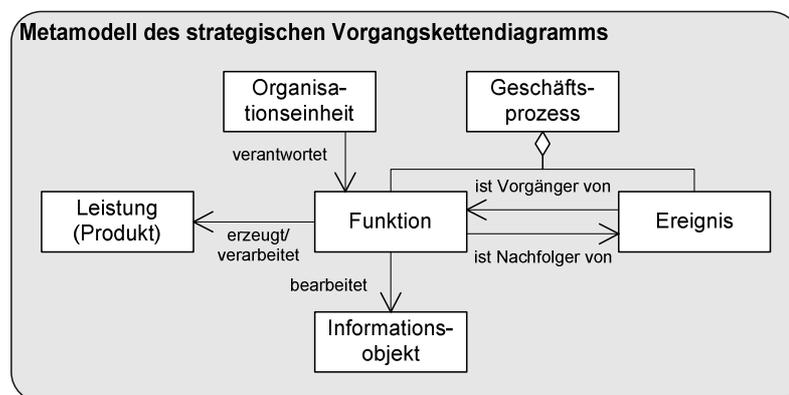


Abbildung 16: Metamodell des strategischen Vorgangskettendiagramms²¹⁴

Die **Funktionssicht** wird häufig im Zusammenhang mit Komponenten anderer Sichten modelliert, wie beispielsweise in Verbindung mit Daten, die von Funktionen transformiert werden, oder Organisationseinheiten, die für die Durchführung einer bestimmten Funktion verantwortlich sind. Der Begriff Funktion ist nicht allgemeingültig definiert und wird häufig synonym mit den Begriffen Vorgang, Tätigkeit, Aktivität oder Aufgabe verwendet. In ARIS ist eine Funktion definiert als eine Verrichtung, die zur Erfüllung einer oder mehrerer Ziele an einem Informationsobjekt (z.B. Auftrag, Kunde, Rechnung) durchgeführt wird.²¹⁵ Die Ziele können netzartig untereinander verflochten sein (n:m-Beziehung). Ein Ziel kann in mehrere detailliertere (Teil-)Ziele zerlegt werden oder auch mehreren abstrakteren Zielen zugeordnet sein. Ebenso wie die Ziele können auch die Funktionen auf unterschiedlichen Aggregationsstufen beschrieben werden. Komplexe Funktionsbündel auf der obersten Aggregationsstufe können mit Hilfe von Hierarchiediagrammen in Teilfunktionen gegliedert werden.

Für die Modellierung einer geschäftsprozessorientierten Organisation wird in ARIS eine Gliederung der Funktionen entsprechend den Geschäftsprozessen vorgeschlagen, während für eine spätere Systementwicklung eine verrichtungsorientierte Gliederung bevorzugt wird.²¹⁶ Im Metamodell der Funktionssicht werden beide Gliederungsarten der Funktionen abgebildet. Der Objekttyp „AFunktion“ (allgemeine Funktion) mit rekursiver Beziehung beschreibt Tätigkeiten unabhängig von Gliederungszusammenhängen, so dass jede Funktion nur einmal erfasst werden muss.

Abbildung 17 zeigt die wesentlichen Metamodellobjekte der Funktionssicht. Ein Geschäftsprozess unterstützt die Erreichung bestimmter Unternehmensziele. Er kann in Unter- oder

²¹⁴ In Anlehnung an Leist (2004), S. 159.

²¹⁵ Vgl. Scheer (2001), S. 22.

²¹⁶ Vgl. Scheer (2001), S. 26.

Teilprozesse zerlegt werden. Dabei können Teilprozesse in mehreren übergeordneten Prozessen vorkommen. Funktionen werden Geschäftsprozessen zugeordnet und unterstützen wie diese die Erreichung bestimmter Ziele. Zur Charakterisierung, ob eine Funktion primär informationssystem-unterstützt oder manuell bearbeitet wird, werden die beiden Subklassen „Manuelle Funktion“ und „Systemfunktion“ differenziert.²¹⁷ Der Systemfunktion wird ein entsprechendes Anwendungssystem zugeordnet. Darüber hinaus können Funktionen danach differenziert werden, ob sie primär einen material- oder informationsverarbeitenden Charakter besitzen.

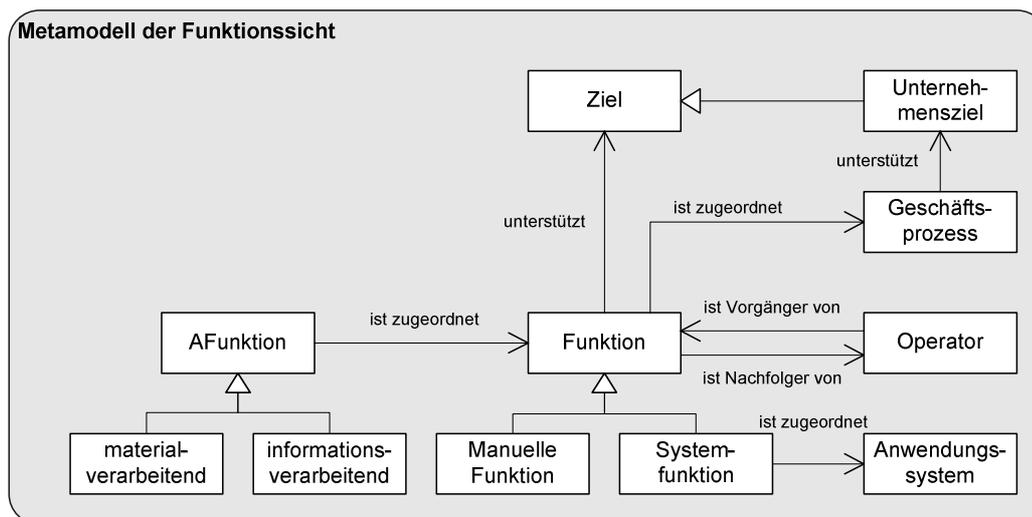


Abbildung 17: Metamodell der Funktionssicht²¹⁸

Die **Organisationssicht** beschreibt die Aufbauorganisation, also die Organisationseinheiten mit den zwischen ihnen bestehenden Kommunikations- und Weisungsbeziehungen.²¹⁹ Ferner wird mit dem Rollenkonzept das Anforderungsprofil einer Organisationseinheit definiert. Abgebildet wird das Organigramm eines Unternehmens, indem gleichartige Aufgabenkomplexe zu Organisationseinheiten zusammengefasst werden.²²⁰ Organisationseinheiten können ebenso wie Funktionen nach verrichtungsorientierten, objektorientierten und prozessorientierten Kriterien zerlegt werden. Bei einer prozessorientierten Gliederung wird das üblicherweise in einer Baumstruktur abgebildete Organigramm in eine vernetzte Struktur überführt. Zudem können Ausprägungen der Organisationseinheiten auf generischer Typebene (Unternehmensleitung, Hauptabteilung, Abteilung usw.), auf Typebene (Vertrieb, Abwicklung, Personal, usw.) oder auf Instanzebene (Vertrieb Privatkunden, Filiale XY, usw.) dargestellt werden.²²¹

²¹⁷ Vgl. Scheer (2001), S. 34.

²¹⁸ Vgl. Scheer (2001), S. 38.

²¹⁹ Vgl. Scheer (2001), S. 52.

²²⁰ Vgl. Scheer (2001), S. 53.

²²¹ Vgl. Scheer (2001), S. 53-54.

Organisationseinheiten können nach primär menschlichen und primär technischen Leistungsträgern differenziert werden. Maschinelle Leistungsträger können menschliche Leistungsträger bei der Ausführung der Aufgaben unterstützen. Des Weiteren wird eine Organisationseinheit durch den Objekttyp „Stelle“ spezialisiert. Eine Stelle bildet die kleinste Einheit einer Organisationsstruktur. Ihr Aufgabenumfang wird so definiert, dass ein Mitarbeiter diesen bewältigen kann.²²² Die geographische Verteilung von Organisationseinheiten wird durch die Beziehung zum Objekttyp „Standort“ abgebildet.

Der Objekttyp „Organisationsstruktur“ definiert die Strukturbeziehungen zwischen Organisationseinheiten. Um unterschiedliche Typen von Organisationsstrukturen modellieren zu können, wird der Objekttyp „Organisationstyp“ dem Metamodell hinzugefügt (beispielsweise zur Differenzierung von fachlicher und disziplinarischer Weisungsbefugnis).

Neben den Organisationseinheiten werden auch Mitarbeitertypen mit einer definierten Qualifikation und Kompetenz betrachtet, die unter dem Begriff Rolle zusammengefasst werden.²²³ Die Klasse „Qualifikation“ erfasst die Qualifikationskriterien, die einer Rolle zugeordnet werden können. Aus Sicht der Stelle können Anforderungen an die Qualifikation gestellt werden. Den Stellen können dann bestimmte Rollen entsprechend der Übereinstimmung der Qualifikations- und Anforderungskriterien zugeordnet werden. Die Klasse „Rolle“ wird weiter spezialisiert durch die Klasse „Benutzerklasse“. Diese kann beispielsweise für die Definition von unterschiedlichen Berechtigungen bei der Nutzung von Anwendungssystemen verwendet werden. Die folgende Abbildung 18 zeigt die zuvor beschriebenen Klassen der Organisations-sicht und deren Beziehungen.

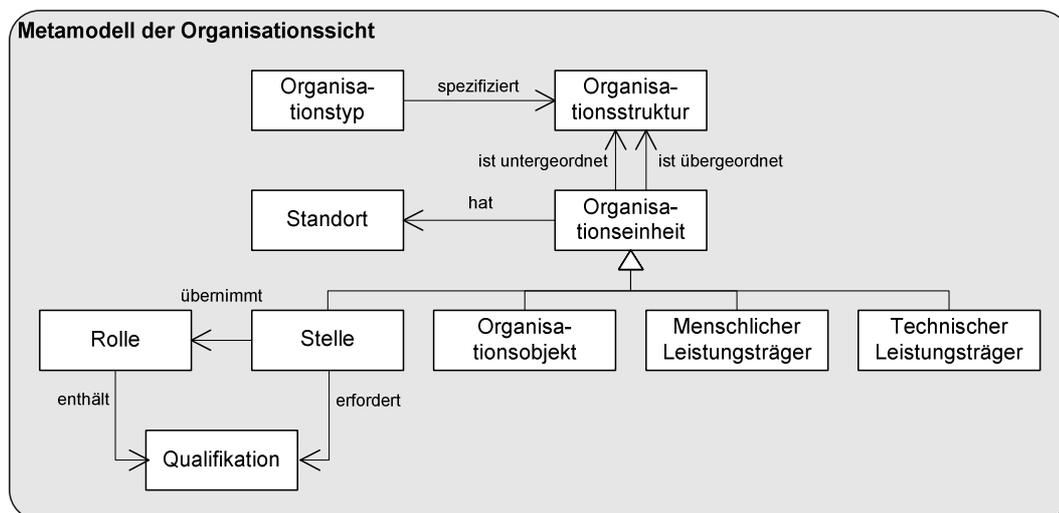


Abbildung 18: Metamodell der Organisationssicht²²⁴

²²² Vgl. Scheer (2001), S. 57.

²²³ Vgl. Scheer (2001), S. 57.

²²⁴ Vgl. Scheer (2001), S. 56.

Die **Datensicht** beschreibt die Datenobjekte, die von Funktionen manipuliert werden. Diese Datenobjekte können als Grundlage für die Definition von Klassen einer objektorientierten Modellierungsmethode, wie z.B. UML, verwendet werden. Überschneidungen mit der Leistungssicht kann es bei Datenobjekten geben, die als Informationsdienstleistungen an nachfolgende Organisationseinheiten weitergegeben werden.²²⁵ Bei der Datensicht wird eine Makro- und Mikrosicht unterschieden. Die Makrosicht beschreibt Daten auf abstrakter Ebene, die in feinere Daten zerlegt werden können, z.B. in ER-Diagramme. Aus Gründen der Anschaulichkeit werden die aggregierten Datenobjekte der Makrosicht für die Modellierung der in einem Geschäftsprozess benötigten Daten verwendet. Der Begriff Makro-Datenobjekt umfasst alle Datensammlungen der Makrosicht. Er wird durch zahlreiche Subtypen (z.B. Voice, Brief, Nachricht, Video, Akte, Datencluster und Datenmodell) spezialisiert. Diese können untereinander in Beziehung stehen. Zudem können Makro-Datenobjekte in elektronisch-alfanumerischer Form oder elektronisch als Sound, Bitmap sowie konventionell auf Papier erfasst werden.²²⁶ Elektronisch gespeicherte Datenobjekte können Anwendungssystemen zugeordnet werden, die sie speichern oder manipulieren.²²⁷

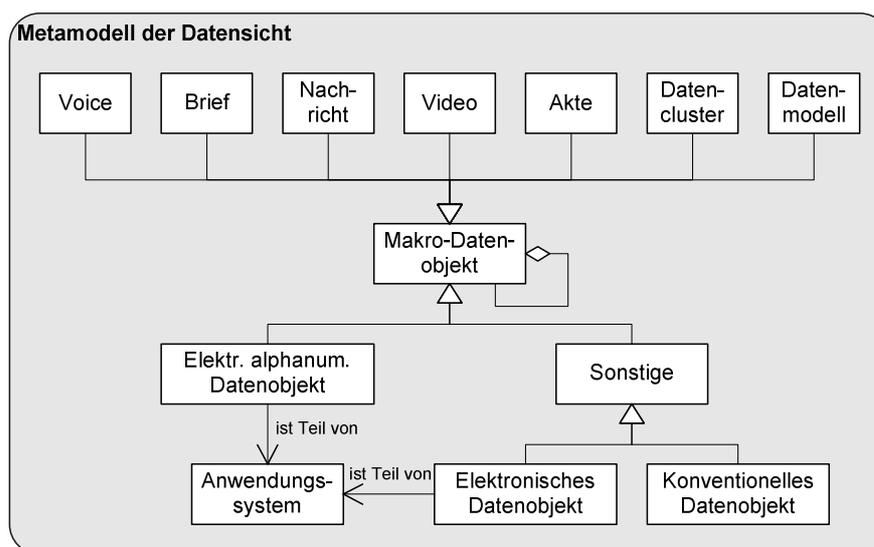


Abbildung 19: Metamodell der Datensicht (Makro)²²⁸

Die Mikrosicht zerlegt die Makro-Datenobjekte in kleinere Einheiten. In der Regel werden für die Modellierung der detaillierten Datenstruktur eines fachlichen Anwendungsbereichs objektorientierte Klassendiagramme, z.B. UML-Klassendiagramme²²⁹, oder ER-Diagramme²³⁰

²²⁵ Vgl. Scheer (2001), S. 67.

²²⁶ Vgl. Scheer (2001), S. 70.

²²⁷ Vgl. Scheer (2001), S. 70.

²²⁸ Vgl. Scheer (2001), S. 69.

²²⁹ Für eine Darstellung des Metamodells von UML-Klassendiagrammen vgl. z.B. OMG (2003b).

²³⁰ Für eine Darstellung des Metamodells von ER-Diagrammen vgl. z.B. Chen (1976) und Scheer (2001), S. 72.

verwendet. Zu den Objekten des einfachen ER-Diagramms gehören der Entitytyp, das Attribut sowie der Beziehungstyp. Das erweiterte ER-Diagramm ermöglicht zudem die Modellierung von Kardinalitäten sowie Spezialisierungs- und Generalisierungsbeziehungen.

Geschäftsprozesse erzeugen und verwenden Leistungen. Durch diese enge Beziehung hat die Granularität der Leistungsdefinition Einfluss auf die Granularität der Prozessbeschreibung. Die Modellierung der **Leistungssicht** nimmt daher innerhalb der Prozessbeschreibung eine Kernfunktion ein. Der Begriff Leistung wird mit dem Begriff Produkt gleichgesetzt.²³¹ Das Metamodell der Leistungssicht teilt den generellen Leistungs- bzw. Produktbegriff in Sach- und Dienstleistungen auf (vgl. Abbildung 20). Dienstleistungen werden weiter differenziert nach Informations- und sonstige Dienstleistungen. Für die Modellierung der Leistungssicht werden Produktbäume mit Kompositionsbeziehung „besteht aus“ verwendet. Die Leistungen werden mit den zu ihrer Erstellung benötigten Kosten bewertet.

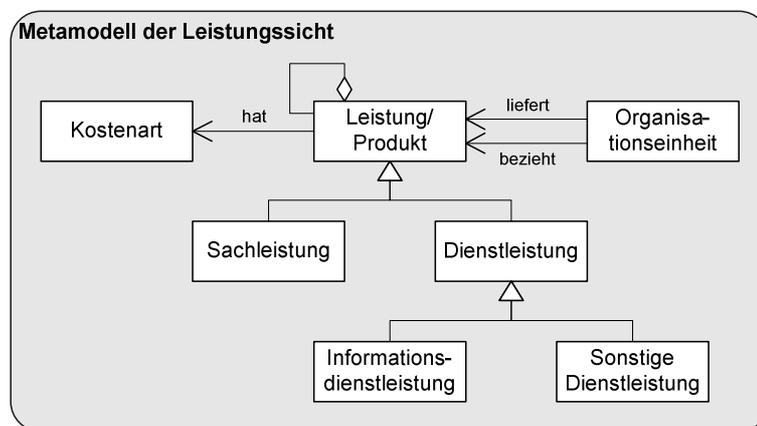


Abbildung 20: Metamodell der Leistungssicht²³²

Die **Steuerungssicht** bildet das Bindeglied zwischen den zunächst getrennt voneinander modellierten Sichten (Funktion, Organisation, Daten und Leistung).²³³ Die Steuerungssicht besteht somit aus sechs weiteren Teilsichten, die die bilateralen Beziehungen zwischen den Sichten darstellen. Für jede dieser Teilsichten existieren Metamodelle, die unterschiedliche Beziehungen zwischen den Objekttypen der Sichten definieren. Da nach SCHEER insbesondere das Prozessmodell für die Integration der unterschiedlichen Teilsichten von Bedeutung ist, wird in dieser Arbeit das Metamodell zur Beschreibung der Modellierungssprache der eEPK vorgestellt.²³⁴ Das von SCHEER entwickelte Metamodell der EPK weist laut LEIST allerdings einige Mängel auf.²³⁵ Deshalb wird in dieser Arbeit, wie bei LEIST, das von ROSEMANN entwickelte Metamodell zur Beschreibung der EPK verwendet, das die Beziehungen zwischen

²³¹ Vgl. Scheer (2001), S. 93.

²³² Vgl. Scheer (2001), S. 97.

²³³ Vgl. Scheer (2001), S. 102.

²³⁴ Vgl. Scheer (2001), S. 102ff. und Scheer/Thomas (2005).

²³⁵ Vgl. Leist (2004), S. 172.

Funktion, Ereignis und Operator der EPK anders wiedergibt (vgl. Abbildung 21).²³⁶ Diese drei Objekttypen sind Teile eines Geschäftsprozesses. Um Vorgänger- und Nachfolgerbeziehungen darstellen zu können, ist jedes dieser Elemente zweifach mit den anderen Elementen verbunden. Der Objekttyp Operator wird differenziert nach „UND“, „ODER“ und „XOR“.

Zur Darstellung der Gesamtsicht werden Elemente der Daten- (z.B. Umfelddatum, Nachricht), Organisations- (z.B. Organisationseinheit und menschliche Arbeitsleistung), Funktions- (z.B. Funktion) und Leistungssicht (z.B. Leistung) in das Metamodell integriert.

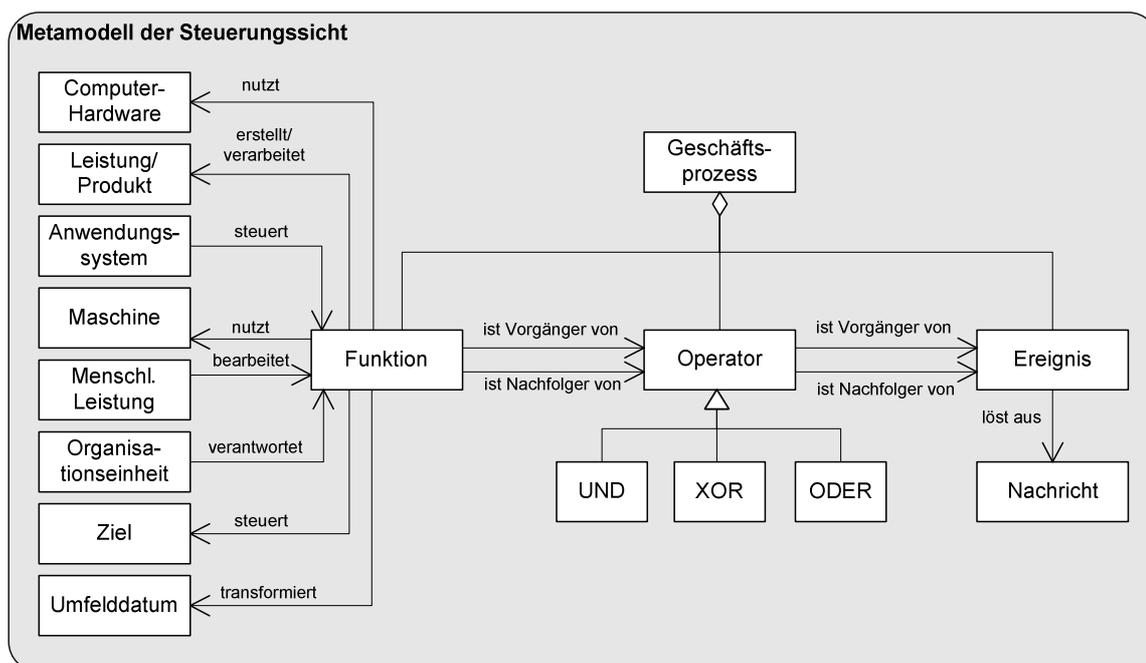


Abbildung 21: Metamodell der Steuerungssicht²³⁷

3.2.1.3 Weitere strukturelle Bestandteile

ARIS beinhaltet kein striktes Vorgehensmodell, das festlegt, in welcher Reihenfolge die einzelnen Sichten zu modellieren sind. Es bietet aber ein Phasenmodell, das für jede Sicht durchlaufen wird und festlegt, wie Geschäftsprozesse ausgehend vom Fachkonzept bis zu ihrer informationstechnischen Implementierung modelliert werden können. Das Phasenmodell besteht aus den Phasen Fachkonzept, DV-Konzept und Implementierung. Zuerst wird das Fachkonzept entwickelt, anschliessend folgt die Gestaltung des DV-Konzepts und zuletzt wird das DV-Konzept implementiert. Je Ebene des Phasenmodells können dabei die einzelnen Sichten in beliebiger Reihenfolge oder auch parallel modelliert werden.

In ARIS kommen für die Modellierung der einzelnen Sichten verschiedene Techniken zum Einsatz, wie beispielsweise ER-Diagramme, UML-Klassendiagramme, USE-CASE-Spezifikationen und EPKs (Ereignisgesteuerte Prozesskette). Zudem werden oftmals einfache

²³⁶ Vgl. dazu Rosemann (1996).

²³⁷ Vgl. Rosemann (1996), S. 122-123.

Beschreibungstechniken, wie Tabellen, Matrizen und Netz- bzw. Baumstrukturen, eingesetzt, um die Instanzen von Objekttypen (z.B. Funktionen und Ziele) einer Sicht auf unterschiedliche Weise einander gegenüberzustellen.

Für jede Sicht werden in ARIS Ergebnisdokumente vorgeschlagen, denen ein gemeinsames Metamodell zugrunde liegt, welches die Konsistenz zwischen den Dokumenten gewährleistet. Die Ergebnisdokumente sind unabhängig voneinander, d.h. keines der Dokumente wird als Input für die Erstellung eines anderen Dokumentes benötigt. Das wichtigste Dokument stellt das EPK-Diagramm dar, da es Objekte aus anderen Sichten beinhaltet und die verschiedenen Sichten dadurch miteinander verbindet.

3.2.1.4 Inhaltliche Charakterisierung

Der Fokus von ARIS liegt stark auf der Modellierung der Geschäftsprozesse (fünf Sichten) und deren informationstechnischer Implementierung. Die Abbildung der strategischen Ausrichtung eines Unternehmens wird dagegen nicht unterstützt. Das Kriterium der Ganzheitlichkeit wird von ARIS somit nicht erfüllt.

Jede Sicht auf den Geschäftsprozess wird durch ein zugrunde liegendes Metamodell beschrieben. Sechs weitere Teilsichten definieren die bilateralen Beziehungen zwischen ausgewählten Sichten bei der Modellierung der Steuerungssicht. Die Strukturierung und Komplexitätsreduktion sowie die Konsistenz zwischen den unterschiedlichen Sichten ist folglich gegeben.

ARIS verwendet für die Darstellung von Geschäftsprozessen und Organisationsstrukturen vertraute Sprachkonzepte, wodurch die Anschaulichkeit und Verständlichkeit der Modelle gesteigert wird. Dies ist auch darauf zurückzuführen, dass ARIS in der Praxis weit verbreitet ist und dessen Konzepte oftmals bereits bekannt und etabliert sind.

Bezogen auf den Modellierungszweck der Geschäftsprozessmodellierung stellt ARIS die erforderlichen Aspekte des abzubildenden Sachverhaltes mit einem hohen Detaillierungs- und Formalisierungsgrad dar. Allerdings ist gerade bei der Modellierung strategischer Aspekte die Detaillierung sehr gering. So werden beispielsweise wichtige Produkt- und Leistungsgruppen des Unternehmens lediglich von dem Metaentitätstyp „Leistungsfeld“ in sehr aggregierter Form abgebildet. Hier wäre eine detailliertere Abbildung, auch im Hinblick auf die anschließende Prozessmodellierung, angebracht.

Die Anpassbarkeit und Erweiterbarkeit der von ARIS verwendeten Modellierungssprachen ist eher gering, da diese sehr anwendungsspezifisch definiert sind.

Die von ARIS definierten Sichten auf den Geschäftsprozess werden alle anhand eines zugrunde liegenden Metamodells (semi-)formal spezifiziert. Für die Beschreibung der einzelnen Metamodelle werden Klassendiagramme nach der Unified Modeling Language (UML)-Notation verwendet. Das Kriterium der Formalisierung wird somit von ARIS erfüllt.

3.2.2 Multi-Perspective Enterprise Modeling (MEMO)

3.2.2.1 Übersicht und Architekturebenen

MEMO ist eine Methode zur multiperspektivischen Unternehmensmodellierung.²³⁸ Die Methode bietet einen Bezugsrahmen, der die Strukturierung eines Unternehmens bzw. einer Organisation auf einem hohen Abstraktionsniveau ermöglicht. Sie ist nicht allein auf die Entwicklung des Informationssystems gerichtet, sondern bezieht auch strategische und organisatorische Aspekte in die Betrachtung ein. Dadurch soll eine wechselseitige Abstimmung des Informationssystems mit der Gestaltung von Geschäftsprozessen und der Unternehmensstrategie gefördert werden.²³⁹

Ziel von MEMO ist es, methodische Unterstützung bei der Modellierung verschiedener Sichten auf ein Unternehmen zu leisten, indem es den Entwurf unterschiedlicher Unternehmensmodelle anleitet. Bei der Wahl der Sichten wurden sowohl betriebswirtschaftliche als auch softwaretechnische Sichtweisen berücksichtigt. Der Bezugsrahmen der Methode unterscheidet drei Gestaltungsebenen: Die Ebene der **Strategie** repräsentiert die Perspektive der Unternehmensführung. Die Perspektive der **Organisation** wird auf der zweiten Ebene durch Modelle der Aufbau- und Ablauforganisation dargestellt. Die dritte Ebene wird aus der Perspektive der **Informationssysteme** konzeptualisiert.

Diese drei Ebenen werden jeweils durch die Sichten „Ressource“, „Struktur“, „Prozess“ und „Ziel“ differenziert. Dadurch entsteht eine Matrix mit zwölf Zellen, die unterschiedliche Unternehmensmodelle und Modellierungssprachen beinhalten. Geschäftsprozesse können beispielsweise innerhalb der Gestaltungsebene „Organisation“ aus der Sicht „Prozess“ mit der in MEMO enthaltenen Modellierungssprache „OrgML“ modelliert werden. Die unterschiedlichen Modellierungssprachen sind Bestandteile einer Spracharchitektur. Die Metamodelle der Modellierungssprachen für die unterschiedlichen Ebenen und Sichten (z.B. MEMO-OML für Objektmodelle²⁴⁰) werden durch eine einheitliche Beschreibungsform, basierend auf einem gemeinsamen Meta-Metamodell sowie gemeinsamen Konzepten, dargestellt.²⁴¹ Dadurch ist es möglich, Beziehungen zwischen Elementen unterschiedlicher Ebenen abzubilden. Zusätzlich zu den Unternehmensmodellen und Modellierungssprachen bietet MEMO ein Vorgehensmodell, das aus einem zyklischen Phasenmodell besteht, dessen Phasen strukturiert beschrieben sind.

²³⁸ Vgl. im Folgenden *Frank* (1994), *Frank* (1995), *Frank* (1998b), *Frank* (1999a), *Frank* (1999b) und *Frank* (2002).

²³⁹ Vgl. *Frank* (1995), S. 46.

²⁴⁰ Vgl. *Frank* (1998b).

²⁴¹ Vgl. *Frank* (1998a).

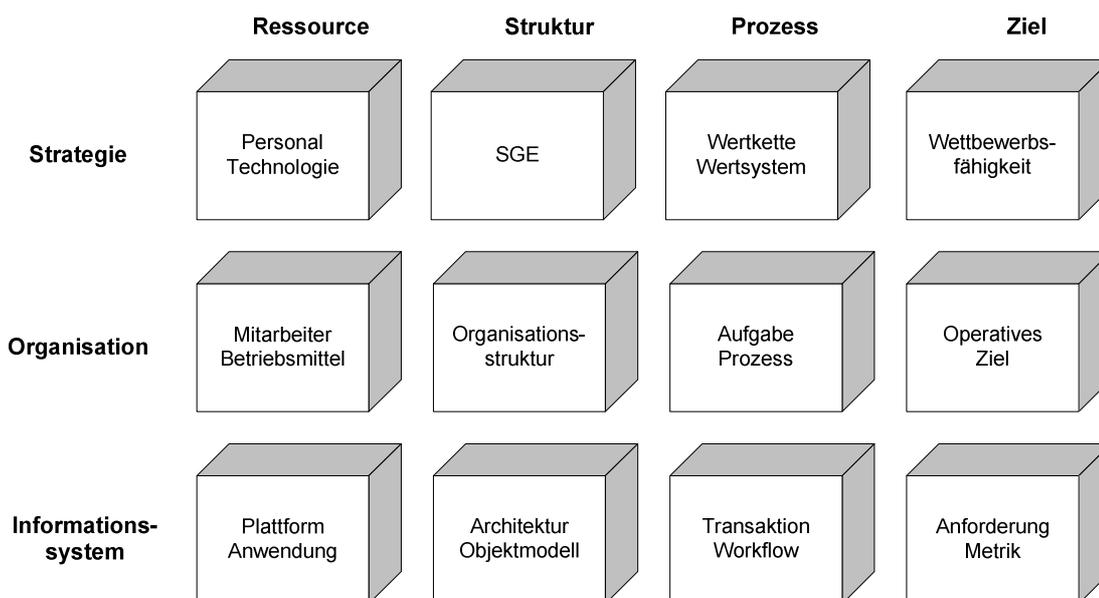


Abbildung 22: Multi-Perspective Enterprise Modeling (MEMO)²⁴²

3.2.2.2 Metamodelle von MEMO

MEMO-SML (Strategy Modeling Language) dient der Erstellung von Modellen, die die Ziele und die Geschäftsstrategie eines Unternehmens beschreiben. Die Sprache beinhaltet vier wesentliche Konzepte.²⁴³ Rollen beschreiben wichtige Akteure im Geschäftsnetzwerk, wie z.B. Kunden, Zulieferer und Wettbewerber. Diese Klassen repräsentieren keine einzelnen Instanzen, sondern eine Menge von Instanzen. Das gleiche gilt für Ressourcen, wie finanzielle Ressourcen, Mitarbeiter, Technologie etc. Märkte dienen der strategischen Planung. Dafür bietet MEMO-SML beispielsweise die Klassen Markt, Teilmarkt, Segment, etc. Um den Modellierer bei der Strukturierung und Analyse der Unternehmensstrategie zu unterstützen, bietet die Sprache Konzepte wie z.B. Aktivität, Aktivitätsgruppe, Wertschöpfungskette, Allgemeine Strategie etc. Einige Klassen bzw. Konzepte stehen in Verbindung mit Klassen aus anderen Modellierungssprachen. So ist beispielsweise die Klasse Geschäftseinheit mit der Klasse Organisationseinheit der MEMO-OrgML assoziiert. Da die Spezifikation von MEMO-SML und deren Notation strikt voneinander getrennt sind, können beliebige graphische Repräsentationen verwendet werden. Abbildung 23 zeigt die wesentlichen Klassen (Konzepte bzw. Metaentitätstypen) der MEMO-SML sowie deren Beziehungen.

²⁴² Vgl. Frank (2002), S. 3.

²⁴³ Vgl. im Folgenden Frank (2002), S. 3025-3027.

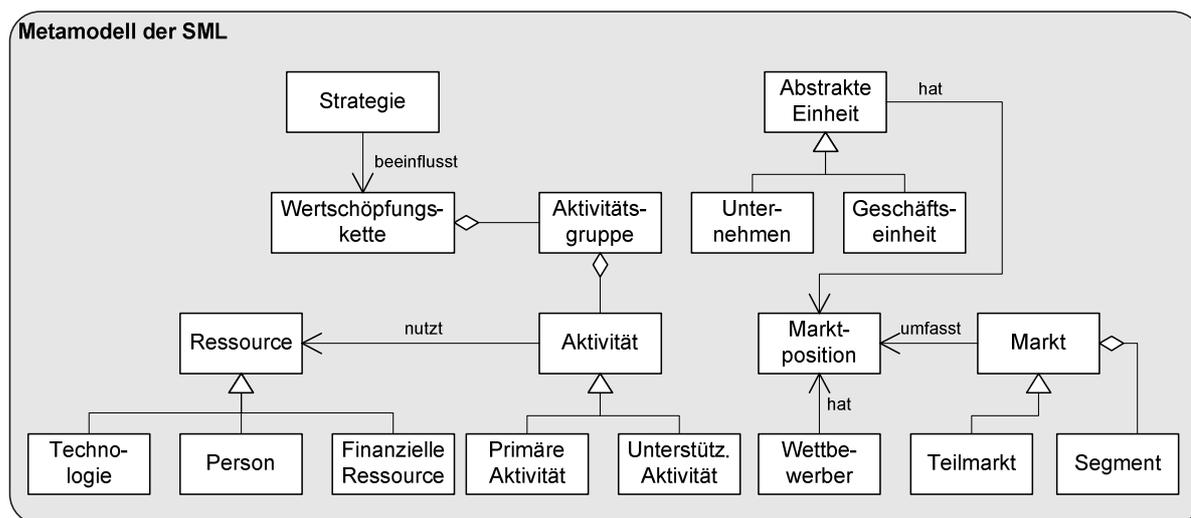


Abbildung 23: Metamodell der MEMO-SML (Strategy Modeling Language)²⁴⁴

Auf Grundlage der Unternehmensstrategie werden die wesentlichen Geschäftsprozesse sowie die darauf abgestimmte Organisationsstruktur des Unternehmens entworfen. **MEMO-OrgML** dient der Modellierung der organisatorischen Perspektive des Unternehmens.²⁴⁵ Die Sprache bietet Konzepte bzw. Metaentitätstypen für die Abbildung von Geschäftsprozessen (z.B. Prozesstyp, Komplexer Prozesstyp, Einfacher Prozesstyp), Organisationsstrukturen (z.B. Organisationseinheit, Stelle, Aggregierte Einheit) und operativen Ressourcen (z.B. Faxgerät, Telefon, Formular). Die wesentlichen Metaentitätstypen für die Prozessmodellierung in MEMO-OrgML sind „Prozesstyp“, „Prozesseinsatz“, „Kontext Prozesseinsatz“, „Input-Spezifikation“, „Output-Spezifikation“ und „Ereignistyp“ (vgl. Abbildung 24). Der abstrakte Metaentitätstyp „Prozesstyp“ wird durch die beiden Metaentitätstypen „Komplexer Prozesstyp“ und „Einfacher Prozesstyp“ spezialisiert. Eine Instanz von „Komplexer Prozesstyp“ kann aus mehreren Instanzen von „Prozesstyp“ zusammengesetzt sein. Für die Unterscheidung zwischen verschiedenen Erscheinungsformen einer Instanz von „Prozesstyp“ wurde der Metaentitätstyp „Prozesseinsatz“ definiert.

Ein „Prozesstyp“ kann eine „Input-Spezifikation“ verwenden und eine oder mehrere „Output-Spezifikationen“ erzeugen. Bei diesen Spezifikationen handelt es sich um Behälter (Container) für Informationen, wie beispielsweise Formulare, Akten oder Objekte, die in einem assoziierten Objektmodell abgebildet sind. Jeder „Output-Spezifikation“ kann für Simulationen zwecke eine Wahrscheinlichkeit zugewiesen werden. Ereignisse werden für die Modellierung des Kontrollflusses innerhalb eines Prozesses verwendet. Sie werden durch bestimmte Zustände eines Prozesses ausgelöst. Die drei Basiskonstrukte für die Modellierung des Kontrollflusses sind sequentielle, alternative und parallele Teilprozesse sowie ein abgeleitetes Konstrukt für die Modellierung von Schleifen. Ausserdem können jedem Prozess Organisations-

²⁴⁴ Vgl. Frank (2002), S. 3028.

²⁴⁵ Vgl. im Folgenden Frank (1999b), S. 3028f.

einheiten und zusätzliche Ressourcen, wie beispielsweise Rollen, Applikationen und Kommunikationsgeräte, zugeordnet werden.

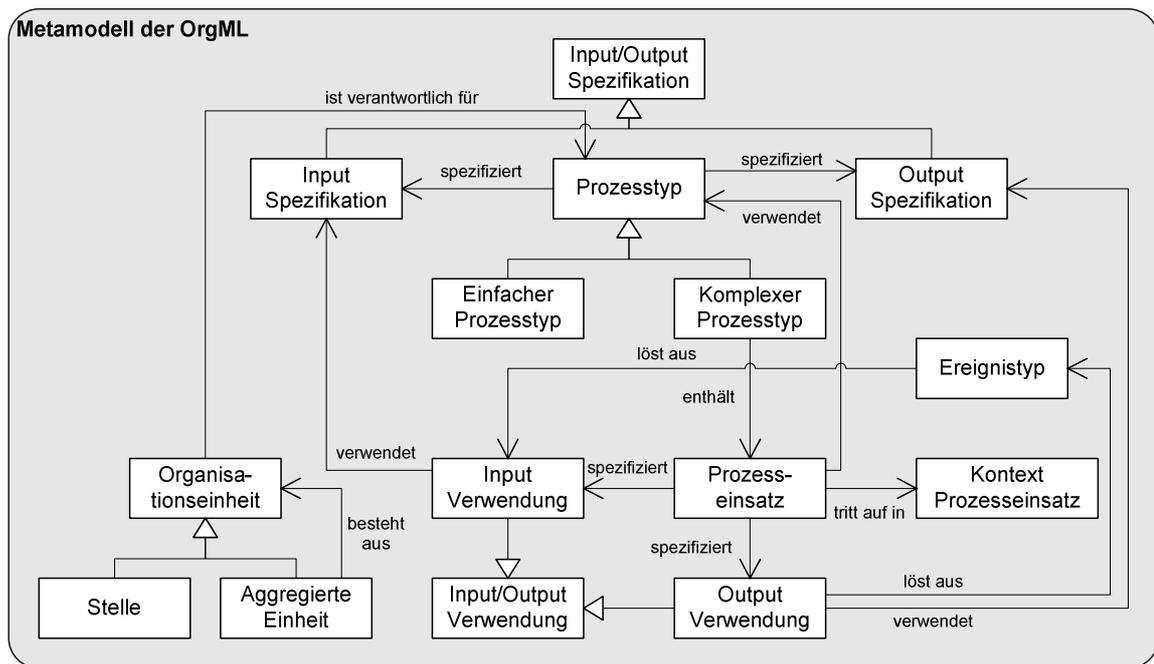


Abbildung 24: Metamodell der MEMO-OrgML²⁴⁶

Auf der Ebene der Informationssysteme dient **MEMO-OML** der Modellierung von statischen Objektmodellen. Dabei handelt es sich um eine objektorientierte Modellierungssprache, deren Konzepte durch Smalltalk inspiriert sind und die in diesem Bereich ähnlich zur UML ist.²⁴⁷ Im Unterschied zur UML ist in MEMO-OML die Semantik von Vererbung genau spezifiziert. Zudem besitzt MEMO-OML das Konzept der Delegation. Dieses soll die Vorteile der Vererbung mit denen von Assoziationen kombinieren.²⁴⁸ Der Metaentitätstyp Klasse der MEMO-OML wird ausserdem durch die Metaentitätstypen Attribut, Service, Guard und Trigger genauer definiert (vgl. Abbildung 25).

²⁴⁶ Vgl. Frank (2002), S. 3028.

²⁴⁷ Vgl. Frank (1999b), S. 162.

²⁴⁸ Vgl. Frank (2002), S. 3027.

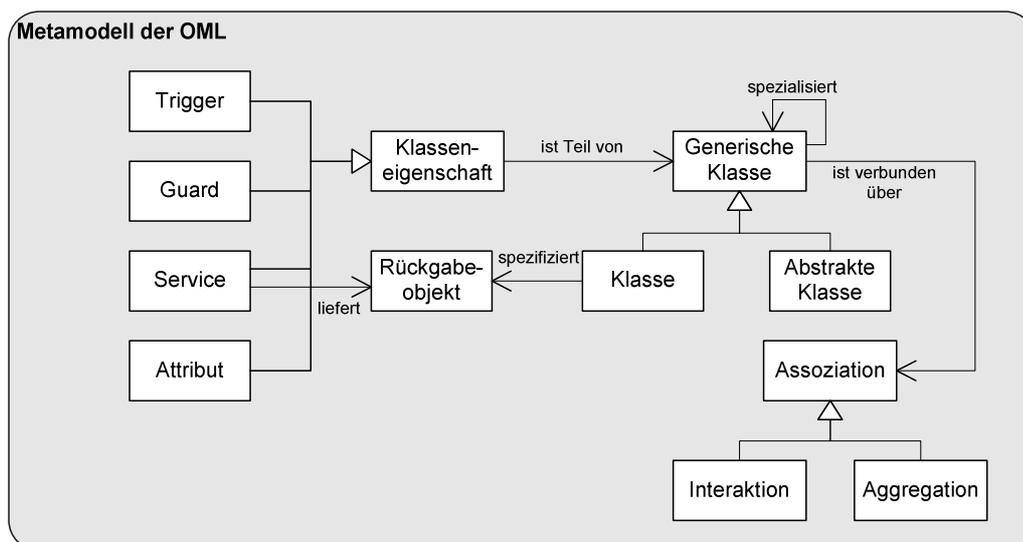


Abbildung 25: Metamodell der MEMO-OML²⁴⁹

3.2.2.3 Weitere strukturelle Bestandteile

MEMO umfasst ein zyklisches Phasenmodell, dessen Phasen strukturiert beschrieben sind und durch Heuristiken unterstützt werden.²⁵⁰ Es wurde allerdings keine veröffentlichte Quelle gefunden, in der dieses Phasenmodell detailliert beschrieben ist. In den vorhandenen Quellen wird lediglich darauf hingewiesen, dass bei der Anwendung der Methode in der Regel ein Top-Down-Ansatz verfolgt wird.

MEMO bietet für die Modellierung der unterschiedlichen Perspektiven und Sichten die Modellierungssprachen MEMO-SML, MEMO-OrgML und MEMO-OML an. Eine detaillierte Handlungsanleitung für die Anwendung dieser Modellierungssprachen liegt allerdings nur für MEMO-OML vor.²⁵¹ Somit sind insbesondere für die Organisations- und Strategieebene keine Techniken spezifiziert.

Die mit den Modellierungssprachen der MEMO-Methode erstellten Diagramme können als Ergebnis der Modellierung betrachtet werden. Es werden somit auf allen Ebenen strukturierte Ergebnisdokumente erzeugt, deren Zusammenhänge ansatzweise im Metamodell der Modellierungssprachen festgelegt sind.

3.2.2.4 Inhaltliche Charakterisierung

MEMO unterteilt das Unternehmen in drei Architekturebenen und definiert unterschiedliche Sichten. Diese decken allerdings nicht alle wesentlichen Elemente der Unternehmensarchitek-

²⁴⁹ Vgl. Frank (2002), S. 3028.

²⁵⁰ Vgl. hierzu die Beschreibung von MEMO auf der Internetseite des Instituts für Informatik und Wirtschaftsinformatik der Universität Duisburg-Essen (<http://www.wi-inf.uni-duisburg-essen.de>).

²⁵¹ Vgl. Frank (1999a).

tur ab.²⁵² Zudem werden nicht für alle Sichten entsprechende Modellierungssprachen und Metamodelle definiert. Auf Organisationsebene dient beispielsweise die Sprache MEMO-OrgML für die Abbildung der Aufbau- und Ablauforganisation. Metaentitätstypen für die Abbildung von operativen Zielen und Erfolgsfaktoren sind in dieser Sprache allerdings nicht enthalten. Ausserdem erlaubt die Matrixstruktur des verwendeten Bezugsrahmens der MEMO-Methode keine flexible Definition von unterschiedlichen Sichten auf den einzelnen Ebenen. Für jede Ebene sind die gleichen Sichten festgelegt. Das Kriterium der Ganzheitlichkeit wird daher von der MEMO-Methode nicht vollständig erfüllt.

Den in MEMO verwendeten Modellierungssprachen liegen Metamodelle zugrunde, die miteinander über gemeinsame Metaentitätstypen verknüpft sind. Zudem existieren für die unterschiedlichen Sprachen gemeinsame Konzepte in Form einer Spracharchitektur und eines Meta-Metamodells. Dadurch kann die Konsistenz zwischen den erstellten Modellen gewährleistet werden.

In Bezug auf die Erfüllung des Kriteriums der Angemessenheit bietet MEMO beispielsweise unterschiedliche Diagrammartentypen zur Darstellung von Prozessen. Ein Dekompositionsdiagramm dient der Darstellung der Kompositionshierarchie von Prozessen. Ein Generalisierungsdiagramm zeigt die Generalisierungsbeziehungen zwischen Prozessstypen. Das Prozessdiagramm erlaubt die detaillierte Beschreibung eines Prozesses. Dadurch ist eine Modellierung der Prozesse mit unterschiedlichen Detaillierungsgraden möglich. Zudem bietet MEMO eine vereinfachte Version der graphischen Notation für Prozesse an. Die anderen Modellierungssprachen werden im Hinblick auf unterschiedliche Ansprüche an die Detaillierung nicht weiter spezifiziert.

Die einzelnen Modellierungssprachen der MEMO-Methode werden formal durch ein zugrundeliegendes Metamodell definiert. Das Metamodell der Sprache MEMO-OML ist zudem in eine Modellierungstechnik eingebunden, d.h. die Ergebnisdokumente werden durch eine vorgegebene Sprache und eine Handlungsanleitung erstellt. Darüber hinaus liegen den verwendeten Modellierungssprachen gemeinsame Konzepte sowie ein einheitliches Meta-Metamodell zugrunde. Dadurch wird die Integration der mit unterschiedlichen Perspektiven korrespondierenden Modelle gefördert und die Erweiterung um weitere Modellierungssprachen im Zeitablauf ermöglicht.

Die Anforderung der Verständlichkeit der verwendeten Notationen ist bei MEMO weitgehend erfüllt, da diese die bekannten Symbole der jeweiligen Domäne umfassen. So wird die Organisationsstruktur beispielsweise mit Hilfe von Organigrammen dargestellt, und gewisse Aspekte der Unternehmensstrategie werden als Wertschöpfungsketten visualisiert. Zudem kön-

²⁵² Vgl. hierzu z.B. die in *Winter/Fischer* (2006) als wesentlich identifizierten Elemente einer Unternehmensarchitektur.

nen in dem unterstützenden Software-Werkzeug (MEMO-Center) die graphischen Symbole modifiziert werden, um eine flexible Anpassung an individuelle Präferenzen zu ermöglichen.

Die in MEMO enthaltene Architektur für die Spezifikation von Sprachen zur Unternehmensmodellierung sieht eine gegebenenfalls im Zeitverlauf wachsende Zahl von Modellierungssprachen vor, deren Integration durch die Verwendung eines einheitlichen Meta-Metamodells sowie gemeinsamer Konzepte erfolgt. Ausserdem ist, wie bereits zuvor genannt, eine Modifikation der graphischen Symbole möglich. Die Flexibilität von MEMO ist allerdings durch die Matrixstruktur des Bezugsrahmens und damit durch die für alle Ebenen (Perspektiven) fest vorgegebenen Sichten eingeschränkt. Das Kriterium der Flexibilität ist deshalb nur teilweise erfüllt.

3.2.3 Semantisches Objektmodell (SOM)

3.2.3.1 Übersicht und Architekturebenen

SOM wurde mit dem Ziel der geschäftsorientierten Gestaltung von Informationssystemen entwickelt.²⁵³ Dies soll durch ein dreistufiges Top-Down-Vorgehen erreicht werden. Die Ergebnisse der Modellierung einer Ebene gelten als Anforderungen für die jeweils nachfolgende Ebene. Die Modellierungsebenen beschreiben nicht nur eine Vorgehensweise, die schrittweise durchzuführen ist, sondern repräsentieren auch die unterschiedlichen Abstraktionsschichten und die betriebswirtschaftlichen Zusammenhänge in einem Unternehmen. SOM unterscheidet drei aufeinander aufbauende Modellierungsebenen (vgl. Abbildung 26):

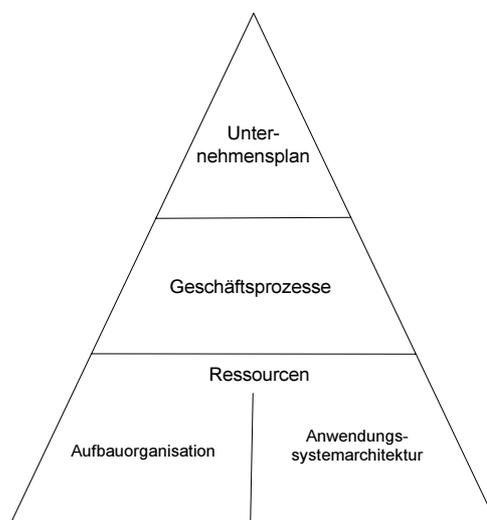


Abbildung 26: SOM-Ebenen²⁵⁴

Der **Unternehmensplan** ist das Ergebnis einer Analyse exogener (Chancen und Risiken) und endogener Erfolgsfaktoren (Stärken und Schwächen). Sie orientiert sich am Erfolg der aktuel-

²⁵³ Vgl. im Folgenden *Ferstl/Sinz* (1998), S. 1041, *Ferstl/Sinz* (1995) und *Ferstl/Sinz* (2001).

²⁵⁴ Vgl. *Ferstl/Sinz* (1995), S. 12.

len und potentiellen Geschäfte des Unternehmens. Dabei werden Wertschöpfungsketten festgelegt sowie Ziele konkretisiert. Die Beschreibung der Analyse und deren Ergebnisse erfolgt auf dieser ersten Ebene in der Regel in textueller Form.

Der Unternehmensplan wird auf der zweiten Ebene durch Geschäftsprozessmodelle umgesetzt. Diese sind durch Leistungsbeziehungen miteinander verbunden. Die **Geschäftsprozesse** werden hierarchisch in Form von semi-formalen Diagrammen modelliert.

Um die Geschäftsprozesse umsetzen bzw. unterstützen zu können, werden entsprechende **Ressourcen**, wie z.B. Anwendungssysteme, Personal und Anlagen, benötigt. Die jeweiligen Zusammenhänge werden durch eine Aufbauorganisation, eine Anwendungssystem-Architektur und eine Anlagen-Architektur dargestellt.

Auf allen Gestaltungsebenen werden verschiedene Sichten unterschieden. Der Unternehmensplan wird in die Sichten Objektsystem und Zielsystem unterteilt. Das Objektsystem beschreibt das Unternehmen sowie dessen Beziehungen zur Umwelt. Das Zielsystem legt die Unternehmensziele sowie Restriktionen und Rahmenbedingungen fest.²⁵⁵ Die Geschäftsprozessebene unterscheidet die Sichten Interaktionsschema (IAS) und Vorgangs-Ereignis-Schema (VES). Das IAS beschreibt das Unternehmen anhand von Objekten (z.B. Kunden, Lieferanten, Produkte) sowie deren Beziehungen untereinander. Das VES besteht aus einer Folge von Aufgabenzerlegungen, die als Petri-Netze dargestellt werden.

Auf Ebene der Ressourcen wird für den Ressourcentyp Anwendungssystem zwischen einem konzeptuellen Objektschema (KOS) und einem Vorgangsobjektschema (VOS) differenziert.²⁵⁶ Das KOS stellt eine objektorientierte Erweiterung eines konzeptuellen Datenmodells dar. Es wird auf der Grundlage der in den IAS und VES definierten Objekttypen und deren Beziehungen modelliert. Das VOS beschreibt darauf aufbauend das Zusammenwirken konzeptueller Objekttypen bei der Durchführung betrieblicher Aufgaben.²⁵⁷

3.2.3.2 Metamodelle der SOM-Methode

Für den Unternehmensplan definiert die SOM-Methode kein Metamodell. Das Objekt- und Zielsystem des Unternehmensplans werden lediglich in natürlichsprachlichem Fliesstext beschrieben.²⁵⁸

Auf der Geschäftsprozessebene werden die Lösungsverfahren für die Realisierung des Unternehmensplans realisiert.²⁵⁹ Die Geschäftsprozesse werden in Abgrenzung zur dritten Ebene

²⁵⁵ Vgl. *Ferstl/Sinz* (2001), S. 183.

²⁵⁶ Vgl. *Ferstl/Sinz* (1998), S. 179ff.

²⁵⁷ Vgl. *Ferstl/Sinz* (2001), S. 184.

²⁵⁸ Vgl. *Ferstl/Sinz* (2001), S. 185.

²⁵⁹ Vgl. *Ferstl/Sinz* (1998), S. 178.

ohne explizite Betrachtung von Aufgabenträgern gestaltet, um mögliche Freiheitsgrade bei der Zuordnung zwischen Aufgaben und Aufgabenträgern ermitteln und nutzen zu können.²⁶⁰

Die Geschäftsprozessebene wird in zwei unterschiedliche Sichten eingeteilt. Betriebliche Objekte und deren Transaktionen werden in einem IAS abgebildet.²⁶¹ Dieses stellt die betrieblichen Objekte (Umwelt- und Diskursweltobjekte) sowie deren Transaktionen (Anbahnungs-, Vereinbarungs-, Durchführungs-, Steuer- und Regelungstransaktionen) dar. Zusätzlich wird der durch Ereignisse gesteuerte Ablauf des Geschäftsprozesses in Form eines VES modelliert. Das VES bildet Aufgaben und Ereignisse sowie deren Beziehungen zu Transaktionen und Objekten ab. Dabei wird festgelegt, dass zwei Aufgaben, die den jeweiligen Objekten zugeordnet sind, eine Transaktion durchführen.²⁶²

Sowohl für das IAS als auch für das VES definiert die SOM-Methode ein Metamodell. Abbildung 27 zeigt das von LEIST ergänzte Metamodell der Geschäftsprozessebene.

Nach Festlegung des Unternehmensplans werden aus Umwelt und Diskurswelt auf der Geschäftsprozessebene betriebliche Objekte (beispielsweise Organisationseinheiten oder Aufgabenträger) abgeleitet. Ein betriebliches Objekt umfasst eine Menge von Aufgaben, die zusammengehörende Sach- und Formalziele verfolgen und die auf einem gemeinsamen Aufgabenobjekt (z.B. Information, materielle Güter, Energie, Zahlungen) durchgeführt werden.²⁶³ Objekte sind hierarchisch zerlegbar und Komponenten eines Lenkungs- oder Leistungssystems.²⁶⁴

Der Transfer von Leistungspaketen und der Austausch von entsprechenden Nachrichten zwischen zwei Objekten werden durch die Klasse Transaktion modelliert.²⁶⁵ Leistungspakete sind beispielsweise Güter, Dienstleistungen oder Zahlungen, die als einzelne Pakete abgrenzbar sind oder durch den Bezug auf eine Zeitperiode abgegrenzt werden können.²⁶⁶ Der Austausch von Nachrichten koordiniert den Transfer der Leistungspakete.²⁶⁷ Leistungspakete sind Bestandteile der Leistung.²⁶⁸ Mit Hilfe von Transaktionen, Nachrichten und Leistungspaketen können die Leistungs- und Lenkungsicht der betrieblichen Aufgaben dargestellt werden.

Aufgaben werden in einer Innensicht und einer Aussensicht beschrieben. Die Aussensicht definiert das Aufgabenobjekt, d.h. den Gegenstand der Aufgabe, die Ziele der Aufgabe (Sach- und Formalziele), die Vorereignisse, die eine Aufgabendurchführung auslösen, und die Nach-

²⁶⁰ Vgl. Schlitt (2003), S. 184.

²⁶¹ Vgl. Ferstl/Sinz (2001), S. 186.

²⁶² Vgl. Ferstl/Sinz (2001), S. 199.

²⁶³ Vgl. Ferstl/Sinz (2001), S. 188.

²⁶⁴ Vgl. Ferstl/Sinz (2001), S. 39.

²⁶⁵ Vgl. Ferstl/Sinz (2001), S. 61.

²⁶⁶ Vgl. Leist (2004), S. 227.

²⁶⁷ Vgl. Ferstl/Sinz (2001), S. 60-61.

²⁶⁸ Vgl. Ferstl/Sinz (2001), S. 188-189.

ereignisse, die aus einer Aufgabendurchführung resultieren.²⁶⁹ Ereignisse können auch danach differenziert werden, ob sie Beziehungen zwischen Aufgaben eines betrieblichen Objekts herstellen (objektinterne Ereignisse, repräsentiert durch den Objekttyp „O-Ereignis“) oder objektextern entstehen (Umweltereignisse, repräsentiert durch den Objekttyp „U-Ereignis“), wie beispielsweise das Eintreten eines bestimmten Zeitpunkts (z.B. der erste Tag eines Monats).²⁷⁰

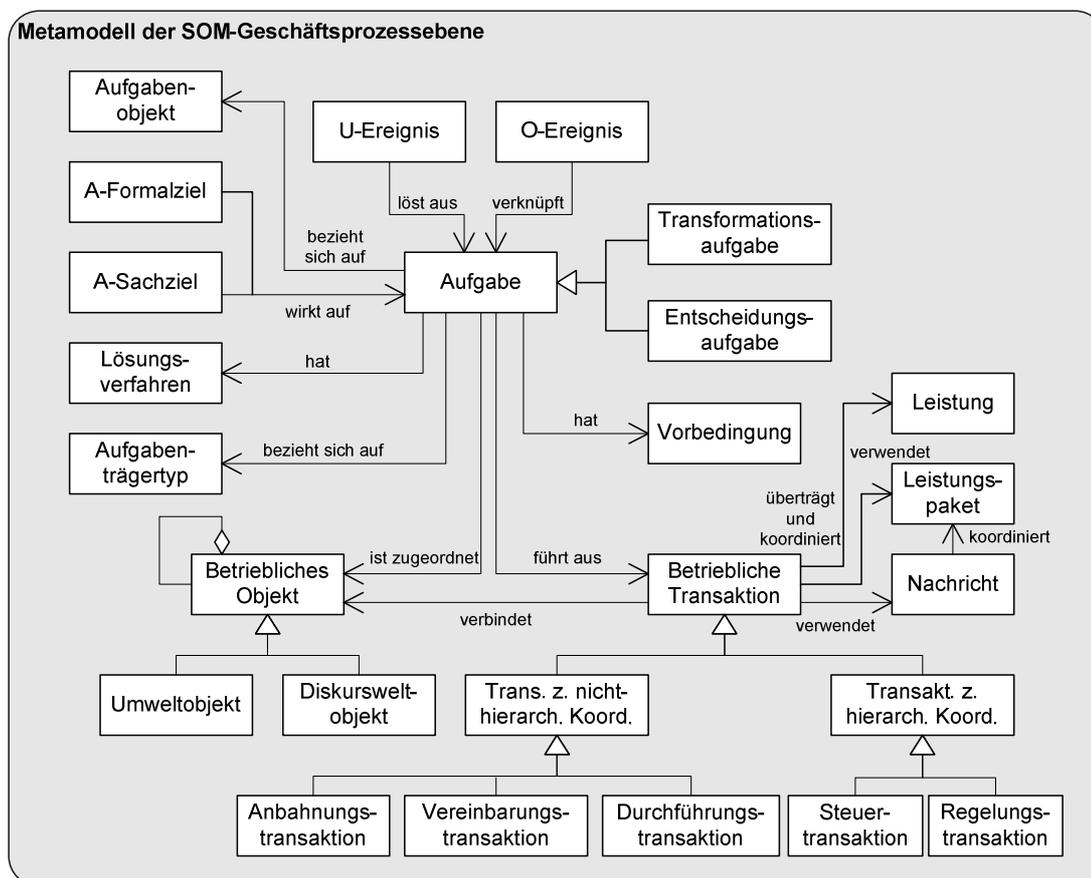


Abbildung 27: Metamodell der Geschäftsprozessebene²⁷¹

Die Innensicht der Aufgabe legt dagegen das Lösungsverfahren (den Verrichtungsvorgang) und den Aufgabenträgertyp (z.B. Mitarbeiter oder Rechner) der Aufgabe fest.²⁷² Aufgaben werden weiter differenziert nach Transformations- und Entscheidungsaufgaben. Geben die Zielsetzungen einer Aufgabe eindeutig an, welcher Nachzustand mit der Aufgabendurchführung erreicht wird, dann liegt eine Transformationsaufgabe vor. Andernfalls wird von einer Entscheidungsaufgabe gesprochen.²⁷³

²⁶⁹ Vgl. Ferstl/Sinz (2001), S. 90.

²⁷⁰ Vgl. Ferstl/Sinz (2001), S. 199.

²⁷¹ Vgl. Leist (2004), S. 232.

²⁷² Vgl. Ferstl/Sinz (2001), S. 90.

²⁷³ Vgl. Ferstl/Sinz (2001), S. 188.

Um die Ergebnisse, die auf der Geschäftsprozessebene entwickelt wurden, auch umsetzen zu können, werden auf der dritten Ebene entsprechende Ressourcen benötigt. In Abhängigkeit von den Festlegungen der ersten beiden Ebenen sind hier die Aufgabenträger für die Unternehmensaufgabe zu spezifizieren.²⁷⁴ Aufgabenträger sind Ressourcen, zu denen Personal, Anwendungssysteme sowie Maschinen und Anlagen zählen.²⁷⁵ Die SOM-Methode beschränkt sich auf die Gestaltung der Anwendungssysteme, da diese eine wesentliche Ressource zur Durchführung der Geschäftsprozesse darstellen. Ein gut detailliertes Geschäftsprozessmodell bildet die Grundlage für die fachliche Spezifikation einzelner Anwendungssysteme und ihrer Integrationsbeziehungen innerhalb der Gesamtarchitektur aller Anwendungssysteme.²⁷⁶ Für die fachliche Spezifikation von Anwendungssystemen werden die beiden Sichten konzeptuelles Objektschema und Vorgangsobjektschema unterschieden (vgl. Abbildung 28). Diese dienen als Grundlage für den systemtechnischen Entwurf und die Implementierung der Anwendungssysteme.²⁷⁷

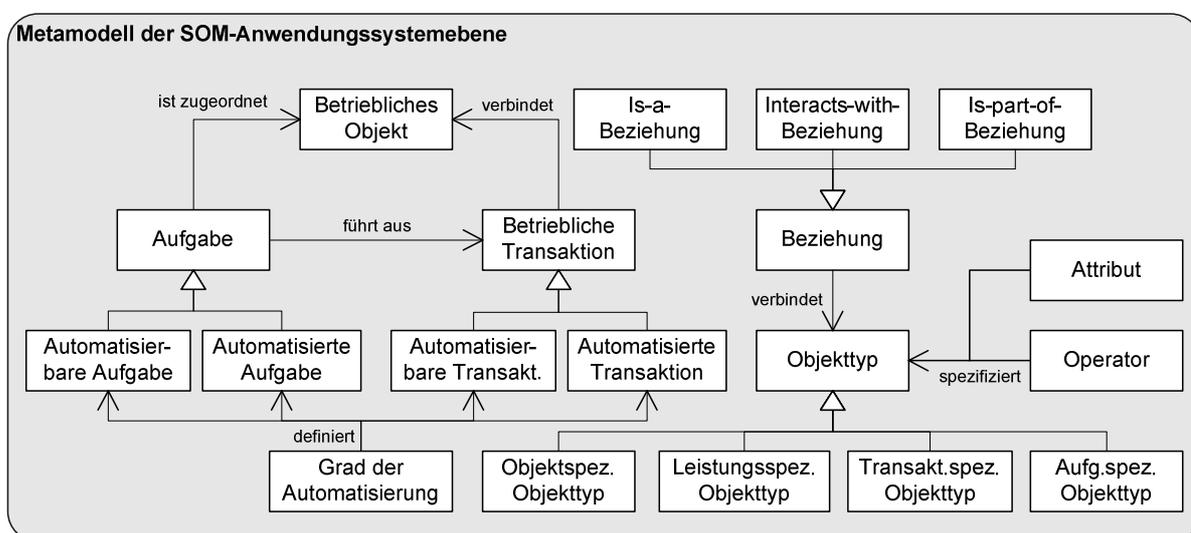


Abbildung 28: Metamodell der Anwendungssystemebene²⁷⁸

Ein konzeptueller Objekttyp wird durch seinen Namen, eine Menge von Attributen, eine Menge von Nachrichtendefinitionen sowie eine Menge von Operatoren (Methoden) spezifiziert. Mit Hilfe der Nachrichtendefinitionen wird festgelegt, welche Nachrichten die Instanzen des Objekttyps verstehen.²⁷⁹ Vorgangsobjekttypen beschreiben das Zusammenwirken von konzeptuellen Objekttypen bei der Durchführung einer betrieblichen Aufgabe. Sie werden

²⁷⁴ Vgl. Schlitt (2003), S. 184.

²⁷⁵ Vgl. Ferstl/Sinz (1998), S. 178.

²⁷⁶ Vgl. Ferstl/Sinz (1995), S. 212.

²⁷⁷ Vgl. Leist (2004), S. 232.

²⁷⁸ Vgl. Leist (2004), S. 237.

²⁷⁹ Vgl. Ferstl/Sinz (2001), S. 203.

ebenfalls durch einen Namen, Attribute, Nachrichtendefinitionen und Operatoren charakterisiert.²⁸⁰

Auf der Anwendungssystemebene werden lediglich automatisierbare Aufgaben und Transaktionen betrachtet. Folglich dient die Automatisierbarkeit als ein wichtiges Kriterium zur weiteren Verfeinerung der auf Geschäftsprozessebene erstellten Modelle. Eine Aufgabe wird als automatisierbar betrachtet, wenn ein Lösungsverfahren angegeben werden kann, für das ein Anwendungssystem zur Verfügung steht oder eines konstruierbar ist.²⁸¹ Eine Transaktion ist automatisierbar, wenn ein Kommunikationssystem die Nachrichtenübertragung und die Ausführung des Kommunikationsprotokolls durchführt.²⁸²

Für die Spezifikation neuer Anwendungssysteme wird zunächst ein KOS und darauf aufbauend ein VOS erstellt. Neben dem Kriterium der Automatisierbarkeit wird darüber hinaus der betrachtete Ausschnitt aus IAS und VES anhand eines oder mehrerer betrieblicher Objekte abgegrenzt, für die ein Anwendungssystem spezifiziert werden soll.²⁸³

Ein KOS umfasst die konzeptuellen Objekttypen, die weiter nach objektspezifischen, leistungsspezifischen und transaktionsspezifischen Objekttypen differenziert werden können. Die Objekttypen werden entsprechend aus Diskurs-/Umweltobjekten, Leistungsspezifikationen und den Transaktionen des Geschäftsprozessmodells abgeleitet.

3.2.3.3 Weitere strukturelle Bestandteile

Neben den Metamodellen für die zuvor genannten Modellierungssprachen umfasst die SOM-Methode auch ein Vorgehensmodell. Dieses beschreibt ein reines Top-Down-Vorgehen. Zunächst wird die Unternehmensaufgabe in Form des Unternehmensplans spezifiziert. Auf der Grundlage dieses Unternehmensplans werden relevante Objekte der Diskurswelt und Umwelt sowie deren Leistungsbeziehungen identifiziert und in einem IAS abgebildet. Leistungsbeziehungen werden dabei durch Transaktionen repräsentiert. Anschliessend wird korrespondierend zum IAS das VES erstellt, welches die Ablaufplanung eines Geschäftsprozesses darstellt. Die beiden Modelle werden in Folge so weit verfeinert, bis der von der Modellierungszielsetzung erreichte Detaillierungsgrad erreicht wurde. Vor der Spezifikation der Anwendungssystemebene wird der vorhandene und umzusetzende Automatisierungsgrad von Aufgaben und Transaktionen festgestellt und ebenfalls in das IAS eingetragen. Aus dem IAS und VES werden auf der Anwendungssystemebene konzeptuelle Objekttypen abgeleitet und in einem KOS abgebildet. Dieses wird daraufhin sukzessive erweitert und verfeinert. Analog zum KOS wird das VOS aus dem IAS und VES abgeleitet und daraufhin in mehreren Schritten verfeinert.

²⁸⁰ Vgl. *Ferstl/Sinz* (2001), S. 210.

²⁸¹ Vgl. *Ferstl/Sinz* (2001), S. 200.

²⁸² Vgl. *Ferstl/Sinz* (2001), S. 200.

²⁸³ Vgl. *Ferstl/Sinz* (2001), S. 206.

Das hier nur grob vorgestellte Vorgehen stellt einen idealtypischen Verlauf eines Projektes dar. Es wird unterstellt, dass die Entwicklung des Unternehmensmodells unabhängig von der aktuellen Situation ist und von Unternehmensplanebene über die Geschäftsprozessebene bis zur Ressourcenebene völlig neu gestaltet werden kann („Grüne-Wiese-Ansatz“). Abhängig von der Zielsetzung und den Rahmenbedingungen ist nach FERSTL/SINZ aber auch eine andere Vorgehensweise möglich. Die einzelnen Modelle sollten aber immer von oben nach unten aufeinander abgestimmt werden.²⁸⁴

Für die Durchführung der Aktivitäten des Vorgehensmodells können unterschiedliche Techniken eingesetzt werden, wie beispielsweise die Modellierungssprachen IAS, VES, KOS und VOS, sowie tabellarische Darstellungen, natürliche Sprache, formale Zerlegungsregeln und Ableitungsregeln.²⁸⁵ Jedes Metamodell ist in eine Modellierungstechnik eingebunden, d.h. die Ergebnisdokumente werden durch eine vorgegebene Sprache und eine Handlungsanleitung erstellt.

Die Dokumente der SOM-Methode können als Ergebnis der Modellierung betrachtet werden. Bis auf die Unternehmensplanebene werden auf allen Ebenen strukturierte Ergebnisdokumente erzeugt.

3.2.3.4 Inhaltliche Charakterisierung

SOM unterteilt das Unternehmen zwar in drei Architekturebenen und definiert für jede Ebene unterschiedliche Sichten. Allerdings ist die Anzahl der Elemente und Sichten je Ebene für einen umfassenden Ansatz, der alle Unternehmensbereiche abdecken soll, zu gering.²⁸⁶ Das Kriterium der Ganzheitlichkeit wird von der SOM-Methode daher nur ansatzweise erreicht.

Die Konsistenz der Ergebnisse der Methode wird durch die Abstimmung der erstellten Modelle in einem Top-Down-Ansatz gewährleistet. Die Ergebnisse einer Ebene sind im Kern als Anforderungen zu verstehen, die in der jeweils folgenden Ebene zu konkretisieren sind.

Bezogen auf die Angemessenheit macht SOM keine Angaben zur Zerlegung und Verfeinerung der erstellten Modelle. Es wird lediglich darauf hingewiesen, dass der Detaillierungsgrad auf die Modellierungszielsetzung auszurichten ist. Soll das Geschäftsprozessmodell beispielsweise als Grundlage für die Entwicklung des Anwendungssystems verwendet werden, so bedingt dies nach FERSTL/SINZ einen wesentlich höheren Detaillierungsgrad, als wenn nur die Leistungsbeziehungen im Unternehmen erfasst werden sollen.²⁸⁷

²⁸⁴ Vgl. *Ferstl/Sinz* (2001), S. 184.

²⁸⁵ Vgl. *Leist* (2004), S. 248.

²⁸⁶ Vgl. hierzu die in *Winter/Fischer* (2006) als wesentlich identifizierten Elemente einer Unternehmensarchitektur.

²⁸⁷ Vgl. *Ferstl/Sinz* (2001), S. 195.

Bis auf den Unternehmensplan werden die Modellierungssprachen der Methode formal durch zugrunde liegende Metamodelle definiert. Für die Erstellung der Ergebnisdokumente sind zudem Modellierungstechniken spezifiziert.

Die Modellierungssprachen der Methode besitzen eine klar definierte, aber anspruchsvolle Syntax, die von dem Anwender bzw. Modellierer sicher eine längere Einarbeitungszeit erfordert. Zudem entsprechen die verwendeten grafischen Symbole nicht direkt den Anwendungsbereichen. Eine deutliche Abbildung von Sprachkonzepten auf grafische Symbole ist demnach nicht gegeben.

Das Kriterium der Flexibilität ist nur teilweise gegeben, da die Erweiterung der Modellierungssprachen und der zugrunde liegenden Metamodelle um weitere Konzepte zwar möglich ist, es allerdings einige Arbeit erfordern dürfte, die neuen Konzepte mit den eher proprietären, der SOM-Methode sehr eigenen Sprachkonzepten in Beziehung zu setzen, um so die Konsistenz zwischen den Modellen gewährleisten zu können.

3.3 Diskussion ausgewählter Bezugsrahmen aus der Praxis

Nach der Diskussion von Methoden zur Unternehmensmodellierung, die primär aus dem universitären Umfeld stammen, werden in diesem Abschnitt ausgewählte Bezugsrahmen aus der Praxis vorgestellt. Im Unterschied zu den oben beschriebenen Methoden bieten diese Ansätze lediglich einen Bezugsrahmen (Framework) für die Kategorisierung von unternehmensrelevanten Informationen anhand von unterschiedlichen Architekturebenen und Sichten. Eine Ausnahme stellt TOGAF dar, welches auch ein detailliertes Vorgehensmodell zur Entwicklung der Architektur vorschreibt.

Zu den am häufigsten von den aktuell am Markt verfügbaren Werkzeugen unterstützten und von Organisationen in der Praxis eingesetzten Bezugsrahmen gehören das Zachman Framework, das Federal Enterprise Architecture Framework (FEAF) und The Open Group Architecture Framework (TOGAF).²⁸⁸ Die ausgewählten Bezugsrahmen werden hinsichtlich der Einteilung des Unternehmens in Gestaltungsebenen und Sichten beschrieben. Zudem erfolgt eine inhaltliche Charakterisierung entsprechend der in Abschnitt 2.6.1 definierten Kriterien.

3.3.1 Das Zachman Framework

3.3.1.1 Überblick und Architekturebenen

Einer der ersten Ansätze zur Abbildung der Informationssystemarchitektur eines Unternehmens wurde 1987 von John Zachman entwickelt und ist auch heute noch weit verbreitet.²⁸⁹ Viele der heute in Software-Werkzeugen standardmässig zur Verfügung gestellten Bezugs-

²⁸⁸ Vgl. z.B. *James* (2004b), *James* (2004a), *IFEAD* (2005) und *Schekkerman* (2004).

²⁸⁹ Vgl. im Folgenden *Zachman* (1987) und *Zachman/Sowa* (1992).

rahmen basieren auf dem Zachman Framework. Es stellt eine Klassifizierung von Konzepten zur Verfügung, um Objekte der realen Welt mit den entsprechenden Repräsentationen in den Informationssystemen zu verbinden.²⁹⁰ Es ermöglicht den Blick auf ein Informationssystem aus mehreren Perspektiven und zeigt, wie diese miteinander in Beziehung stehen. Die erste Version des Bezugsrahmens entwickelte Zachman in Analogie zur Architektur eines Hauses. Anhand verschiedener Rollen („planner“, „owner“, „designer“, „builder“ und „subcontractor“) wird gezeigt, dass jeweils unterschiedliche Konzeptualisierungen des Hauses und seiner Teile im Vordergrund stehen. Die mit den Rollen verbundenen Sichten differenziert er weiter in drei Aspekte, die mit „Was“, „Wie“ und „Wo“ bezeichnet werden. Die Übertragung dieser Überlegungen auf betriebliche Informationssysteme führt zu einem Bezugsrahmen mit fünf Sichten und drei Betrachtungsaspekten (vgl. Abbildung 29). Dabei schränken die Ergebnisse einer Sicht jeweils die Modellierung der darunter liegenden Sichten ein. Jede der auf diese Weise entstehenden fünfzehn Teilsichten wird mit einer bestimmten Repräsentationsform assoziiert.

Das Zachman Framework besteht aus den folgenden fünf Sichten:

- Anwendungsbereich: Diese Sicht ist für den „Bauherrn“ oder „Investor“ gedacht. Sie gibt einen Überblick über den Anwendungsbereich, die Kosten und den Funktionsumfang des Systems.

	Entität (Was)	Funktion (Wie)	Standort (Wo)
Anwendungsbereich (Planner)			
Geschäftsmodell (Owner)			
Systemmodell (Designer)			
Technologiemodell (Builder)			
Komponenten (Subcontractor)			

Abbildung 29: Das Zachman Framework²⁹¹

²⁹⁰ Vgl. Zachman/Sowa (1992), S. 590.

²⁹¹ Vgl. Zachman/Sowa (1992), S. 593.

- Geschäftsmodell (fachkonzeptionell): Dieses Modell beschreibt, beschränkt auf die einzelnen Teilsichten, die Geschäftsobjekte und -prozesse sowie deren Interaktion.
- Systemmodell (logisch): Das Systemmodell bildet die physische Verteilung eines Informationssystems ab und definiert die Datenobjekte und Funktionen, welche die Geschäftsobjekte und -prozesse repräsentieren.
- Technologiemoell (physisch): Für die Umsetzung des Informationssystems muss festgelegt werden, welche Programmiersprache, Hardware oder welche sonstigen Technologien verwendet werden sollen.
- Komponenten: Detaillierte Spezifikationen werden an die Programmierer übergeben. Diese kodieren dann anhand der Spezifikationen individuelle Module, ohne den Gesamtzusammenhang oder die Gesamtstruktur des Systems zu kennen.

Die ersten drei Spalten, welche bereits in der ersten Version des Zachman Framework²⁹² beschrieben wurden, repräsentieren die Entitäten, Funktionen und Standorte eines Informationssystems. In einer späteren Arbeit²⁹³ wurde das Framework um die Spalten Personen, Zeit und Motivation bzw. Strategien erweitert. Innerhalb der 30 Zellen der resultierenden Matrix können unterschiedliche Modellierungssprachen für die Beschreibung der entsprechenden Sicht auf das Informationssystem verwendet werden.

3.3.1.2 Methodenbestandteile des Zachman Framework

Für jede Zelle der Matrix des Zachman Framework werden Ergebnisdokumente vorgeschlagen, wie beispielsweise Entity-Relationship-Diagramme für die Datenmodellierung oder Flussdiagramme für die Prozessmodellierung der Geschäftsmodellsicht. Dadurch können die unterschiedlichen Diagramme und Dokumente, die in einem Unternehmen verwendet werden und die verschiedene Aspekte bzw. Sichten auf dieses Unternehmen repräsentieren, in das Zachman Framework eingeordnet werden. Das Zachman Framework beinhaltet allerdings kein Metamodell, welches die Konsistenz zwischen den einzelnen Ergebnisdokumenten garantiert.

Das Zachman Framework schreibt kein Vorgehensmodell und keine Techniken vor, um die Ergebnisdokumente der einzelnen Zellen zu erzeugen. Es werden lediglich einige generelle Regeln und Prinzipien, die bei der Anwendung des Zachman Framework beachtet werden sollten, vorgeschlagen. Die Reihenfolge, in der die einzelnen Sichten bzw. Zellen entwickelt werden sollen, ist nicht festgelegt. Es kann aber davon ausgegangen werden, dass im Sinne eines Top-Down-Vorgehens die oberen Zeilen der Matrix zu Beginn der Entwicklung einer

²⁹² Vgl. Zachman (1987).

²⁹³ Vgl. Zachman/Sowa (1992).

Unternehmensarchitektur verwendet werden, während die unteren Zellen dagegen eher bei den späteren Entwicklungsphasen von Bedeutung sind.

3.3.1.3 Inhaltliche Charakterisierung

Das Zachman Framework bildet wesentliche Aspekte der Unternehmensarchitektur ab. Der Bezugsrahmen berücksichtigt allerdings nicht die strategische Ausrichtung eines Unternehmens. Der Anwendungsbereich beschreibt lediglich den Kontext, in dem das Informationssystem eingesetzt werden soll, sowie die daraus abgeleiteten Anforderungen. Das Geschäftsmodell bildet Informationsobjekte, Geschäftsprozesse sowie Organisationseinheiten ab und korrespondiert daher mit der in Abschnitt 2.3.1 identifizierten Prozess- bzw. Organisationsarchitektur.

Das Zachman Framework erfüllt das Kriterium der Strukturierung, indem es eine Matrix definiert, in der Informationen über das Unternehmen nach unterschiedlichen Sichten klassifiziert werden können.

Das Zachman Framework stellt lediglich einen Bezugsrahmen für die Klassifizierung von Modellen zur Verfügung. Den für die Modellierung der unterschiedlichen Zellen der Matrix verwendeten Modellierungssprachen liegt aber kein gemeinsames Metamodell zugrunde. Dadurch ist die Konsistenz zwischen den erstellten Modellen nicht gewährleistet.

Da das Zachman Framework keine Modellierungssprachen vorschreibt, kann deren Verständlichkeit nicht bewertet werden. Der Bezugsrahmen sollte aber auch für Mitarbeiter aus den Fachabteilungen aufgrund der Analogie zur Architektur eines Hauses weitgehend verständlich sein.

Der Grad der Detaillierung nimmt nicht von den oberen zu den unteren Zeilen zu, sondern innerhalb einer Zelle durch die Dekomposition der jeweiligen Modelle. Hinweise, wie diese Dekomposition zu erfolgen hat und bis zu welchem Grad, werden nicht gegeben. Dies erschwert auch eine mögliche Integration der Modelle in unterschiedlichen Zellen, da diese eventuell nicht den gleichen Detaillierungsgrad aufweisen.

Für die Modellierung der Inhalte der einzelnen Zellen werden keine Modellierungssprachen fest vorgegeben. Diese sind somit flexibel anpassbar. Allerdings definiert die Matrix eine sehr starre Struktur. Es können zwar zusätzliche Spalten definiert werden. Diese erstrecken sich dann aber über alle Zeilen bzw. Sichten. Dadurch können Zellen entstehen, zu denen keine Informationen vorliegen und die folglich nicht modelliert werden können.

Das Kriterium der Formalisierung erfüllt das Zachman Framework nicht. Zum einen wird nicht formal spezifiziert, wie die Inhalte der einzelnen Komponenten dargestellt bzw. dokumentiert werden sollen. Darüber hinaus existiert kein zugrunde liegendes Metamodell, das die abzubildenden Informationen und deren Zusammenhänge definiert. Dadurch ist die Gefahr von Inkonsistenzen zwischen den einzelnen Zellen bzw. Modellen sehr hoch. Diese wird zu-

dem dadurch gesteigert, dass keine systematische Vorgehensweise für die Modellierung der Zellen vorgegeben wird.

3.3.2 Federal Enterprise Architecture Framework (FEAF)

3.3.2.1 Überblick und Architekturebenen

Aufgrund des Clinger-Cohen Act von 1996²⁹⁴, welcher von staatlichen Organisationen der USA die Entwicklung und die Pflege einer Unternehmensarchitektur fordert, wurde das Federal Enterprise Architecture Framework (FEAF)²⁹⁵ aufgestellt. Das Ziel dieses Frameworks ist es, die gemeinsame Entwicklung von Prozessen und Informationsstrukturen zwischen staatlichen Organisationen zu vereinfachen. Wie die anderen Frameworks ist auch FEAF ein Bezugsrahmen, der dazu dient, die Informationen der Unternehmensarchitektur strukturiert abzubilden.

Das Framework besteht aus den Komponenten Architekturtreiber, strategische Ausrichtung, Ist-Architektur, Soll-Architektur, Transformationsprozess, Architektursegmente, Architekturmodelle und Standards.²⁹⁶

- Architekturtreiber sind externe Einflussfaktoren (z.B. Regulationen, Budgetrestriktionen, Marktänderungen sowie neue Technologien), die eine Änderung oder Anpassung der Unternehmensarchitektur erfordern.
- Die strategische Ausrichtung beschreibt, welchen Anforderungen die Zielarchitektur genügen muss. Sie besteht aus einer Vision, Prinzipien und Zielvorgaben.
- Die Ist-Architektur beschreibt die bestehende Unternehmensarchitektur.
- Die Soll-Architektur legt die zu entwickelnde Unternehmensarchitektur fest.
- Der Transformationsprozess definiert die Migration von der bestehenden Architektur zur Soll-Architektur in Übereinstimmung mit bestimmten Architekturstandards.
- Die Architektursegmente stellen Teilbereiche des gesamten Unternehmens dar.
- Die Komponente Standards umfasst alle zur Verfügung stehenden Standards, Richtlinien und Best Practices.

Die zuvor genannten Komponenten werden anhand von vier unterschiedlichen Detaillierungsgraden („Level“) beschrieben. Level 1 beschreibt die acht Komponenten lediglich überblicksartig. Auf Level 2 und 3 wird im Anschluss entschieden, welche Informationen von Level 1 detaillierter abgebildet werden sollen. Level 4 definiert letztendlich die Modelle, wel-

²⁹⁴ Vgl. *CIO-Council* (1996).

²⁹⁵ Vgl. im Folgenden *FEA* (1999) und *FEA* (2001).

²⁹⁶ Vgl. im Folgenden *FEA* (1999), S. 6-7.

che die Geschäftsarchitektur, die Applikations- und Datenarchitektur sowie die Technologiearchitektur abbilden (vgl. Abbildung 30).

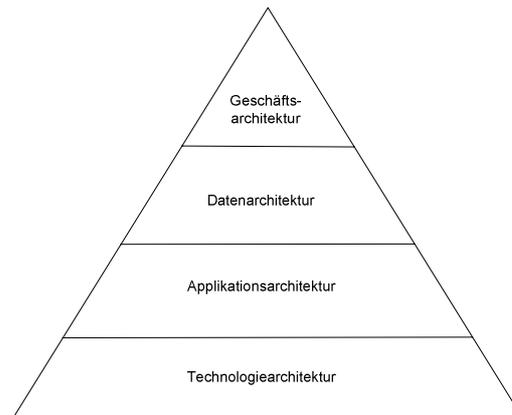


Abbildung 30: *Federal Enterprise Architecture Framework*

FEAF unterteilt somit die Unternehmensarchitektur in diese vier Architekturebenen, wobei jedes Level einen Bezugsrahmen für das nachfolgende Level darstellt. Zusätzlich bietet es auf Level 4 eine logische Struktur zur Klassifizierung der Komponenten, die auf den ersten drei Spalten des Zachman Framework basiert.

3.3.2.2 Methodenbestandteile von FEAF

FEAF schlägt für jede Zelle der Matrix bestimmte Modellierungssprachen und Ergebnisdokumente vor. So werden zum Beispiel für die Beschreibung der Applikationsarchitektur aus Sicht des „Owner“ ein Geschäftsprozessmodell in Form eines UML-Aktivitätsdiagramms oder für die Datenarchitektur aus Sicht des „Designer“ ein logisches Datenmodell in Form eines UML-Klassendiagramms vorgeschlagen. Ein Metamodell, welches die Ergebnisdokumente und vor allem deren Zusammenhänge festlegt, ist allerdings nicht vorhanden. Zudem stellt FEAF keine detaillierte Beschreibung zur Verfügung, wie die einzelnen Ergebnisdokumente erstellt werden sollen. Um die Zielarchitektur zu entwickeln, gibt FEAF lediglich ein sehr unspezifisches Vorgehensmodell an, welches aus vier wesentlichen Phasen besteht: Datenerfassung, vorläufige Produkterstellung, Bewertungs- und Anpassungsphase, Publikations- und Umsetzungsphase. Die einzelnen Aktivitäten dieses generellen Vorgehens werden allerdings nicht weiter detailliert. Für die Modellierung der ersten beiden Ebenen des Zachman Framework („Planner“ und „Owner“) auf Level 4 wird zudem die Vorgehensweise „Enterprise Architecture Planning“ (EAP) nach SPEWAK²⁹⁷ vorgeschlagen. Diese besteht aus sieben Schritten, in denen ein Plan für den Entwurf und die Implementierung aufgestellt wird. EAP ist allerdings lediglich auf die Analyse von Daten, Applikationen und den zugrunde liegenden Technologien in Form von Matrizen fokussiert.

²⁹⁷ Vgl. *Spewak/Hill* (1993).

3.3.2.3 Inhaltliche Charakterisierung

FEAF stellt einen weitgehend ganzheitlichen Ansatz zur Abbildung der Unternehmensarchitektur dar. Allerdings wird auch von FEAF die strategische Ausrichtung des Unternehmens vernachlässigt. Die Komponente „strategische Ausrichtung“ soll zwar gewährleisten, dass die Ziel-Architektur mit den Zielen der gesamten Organisation übereinstimmt. Welche Informationen dort abgebildet werden sollen, welche Modellierungstechniken verwendet werden sollen und welches Vorgehen dafür notwendig ist, wird allerdings nicht festgelegt. Überdies sind die Beziehungen zu den anderen Komponenten der Unternehmensarchitektur nicht spezifiziert.

FEAF liefert eine detaillierte Strukturierung der abzubildenden Informationen. Zum einen besteht der Bezugsrahmen aus unterschiedlichen Komponenten. Jede dieser Komponenten wird mit einem zunehmenden Detaillierungsgrad (Level) beschrieben. Auf Level 4 wird das Unternehmen in unterschiedliche Architekturebenen eingeteilt. Zusätzlich werden die Informationen der Daten-, Applikations- und Technologiearchitektur anhand der ersten drei Spalten des Zachman Framework klassifiziert.

Trotz der detaillierten Strukturierung besitzt FEAF den wesentlichen Mangel, dass den für die Modellierung der unterschiedlichen Zellen der Matrix verwendeten Modellierungssprachen kein gemeinsames Metamodell zugrunde liegt. Dadurch kann die Konsistenz zwischen den erstellten Modellen nicht durch FEAF gewährleistet werden. Der Anwender muss selbst die einzelnen Modelle konsistent halten. Bei einer derart grossen Zahl an Zellen und damit unterschiedlichen Modellierungssprachen ist dies eine erhebliche Herausforderung.

Das Kriterium der Verständlichkeit in Bezug auf die eingesetzten Modellierungssprachen kann nicht bewertet werden, da FEAF diese nicht explizit vorschreibt. Der Anwender ist selbst für die Wahl geeigneter Modellierungssprachen verantwortlich. Es ist allerdings fraglich, ob diese detaillierte Strukturierung nach unterschiedlichen Komponenten, Detaillierungsgraden, Architekturebenen sowie zahlreichen Perspektiven und Sichten noch zu der eigentlich intendierten Komplexitätsreduktion führt.

Die unterschiedlichen Level sollen einen zunehmenden Detaillierungsgrad bei der Abbildung der Informationen über das Unternehmen ermöglichen. Allerdings ist die Definition dieser Level sehr unspezifisch, so dass der Anwender bzw. Architekt nur wenig Anhaltspunkte für den richtigen Detaillierungsgrad und die entsprechende Darstellungsform für den jeweiligen Level erhält. Das Kriterium der Angemessenheit ist demzufolge nur unzureichend erfüllt.

In Bezug auf die verwendeten Modellierungssprachen macht FEAF keine Vorgaben und ist demzufolge flexibel anpassbar. Allerdings stellt die auf Level 4 definierte Matrix des Zachman Framework eine fest vorgegebene Struktur zur Verfügung, die die Flexibilität bei der Definition neuer Sichten einschränkt.

Das Kriterium der Formalisierung wird von FEAF nicht erfüllt. Zum einen wird nicht vorgeschrieben, in welcher Form die Inhalte der einzelnen Komponenten dargestellt bzw. doku-

mentiert werden sollen. Zum anderen existiert kein zugrunde liegendes Metamodell, das die abzubildenden Informationen und deren Zusammenhänge definiert.

3.3.3 The Open Group Architecture Framework (TOGAF)

3.3.3.1 Überblick und Architekturebenen

TOGAF²⁹⁸ ist ein Industriestandard, der Mitte der 90er Jahre von einer Gruppe, bestehend aus Vertretern führender IT-Anwender- und IT-Anbieterorganisationen, entwickelt wurde und bis heute stetig weiterentwickelt wird. Im Unterschied zum Zachman Framework, das lediglich einen Bezugsrahmen zur Klassifizierung von Informationen über das Unternehmen definiert, beinhaltet TOGAF auch ein zyklisches Vorgehensmodell.

Ursprünglich war die von TOGAF verwendete Methode auf die Entwicklung von IT-Architekturen fokussiert. Erst in der aktuellen Version 8.0 wurde diese Methode für die Entwicklung anderer Ebenen einer Unternehmensarchitektur erweitert. Diese unterstützt die Gestaltung der folgenden vier Ebenen:

- Die Geschäftsarchitektur definiert die Geschäftsstrategie, die Organisationsstruktur sowie die Kern-Geschäftsprozesse.
- Die Datenarchitektur beschreibt die Struktur der logischen und physischen Daten und Datenquellen eines Unternehmens.
- Die Applikationsarchitektur gibt einen Überblick über die einzelnen zu entwickelnden Anwendungssysteme sowie deren Beziehung zu den Kern-Geschäftsprozessen.
- Die Technologiearchitektur beschreibt die Software- und Hardware-Infrastruktur, die der Unterstützung der Kernapplikationen dient.

TOGAF schreibt keine speziellen Sichten auf die unterschiedlichen Architekturebenen vor. Stattdessen werden lediglich exemplarisch unterschiedliche Arten von Sichten beschrieben, die für die Gestaltung einer bestimmten Unternehmensarchitektur von Bedeutung sein könnten. Zudem bietet TOGAF Richtlinien für die Wahl bestimmter Sichten und deren Entwicklung.

3.3.3.2 Methodenbestandteile von TOGAF

Das Kernstück von TOGAF ist die Architecture Development Method (ADM). Diese dient der systematischen Ableitung einer unternehmensspezifischen, die Geschäftsanforderungen unterstützenden Architektur. Sie besteht aus einem systematischen, iterativen Vorgehen, unterschiedlichen Sichten, Vorschlägen für zu verwendende Modellierungstechniken, Verweisen auf praktische Fallstudien sowie Hinweisen zu Software-Werkzeugen. Ziel ist die Ausrich-

²⁹⁸ Vgl. im Folgenden *Opengroup* (2002) und *Varveris/Harrison* (2005).

tung der Geschäfts-, Informationssystem- und Technologiearchitektur an den Geschäftsanforderungen und -zielen des Unternehmens.

Der iterative Prozess der ADM besteht aus acht Phasen (vgl. Abbildung 31): Architekturvision, Geschäftsarchitektur, Informationssystemarchitektur, Technologiearchitektur, Lösungen, Migrationsplanung, Umsetzung und Change Management. Jede dieser acht Phasen enthält weitere Schritte, um eine spezifische Unternehmensarchitektur zu entwickeln.

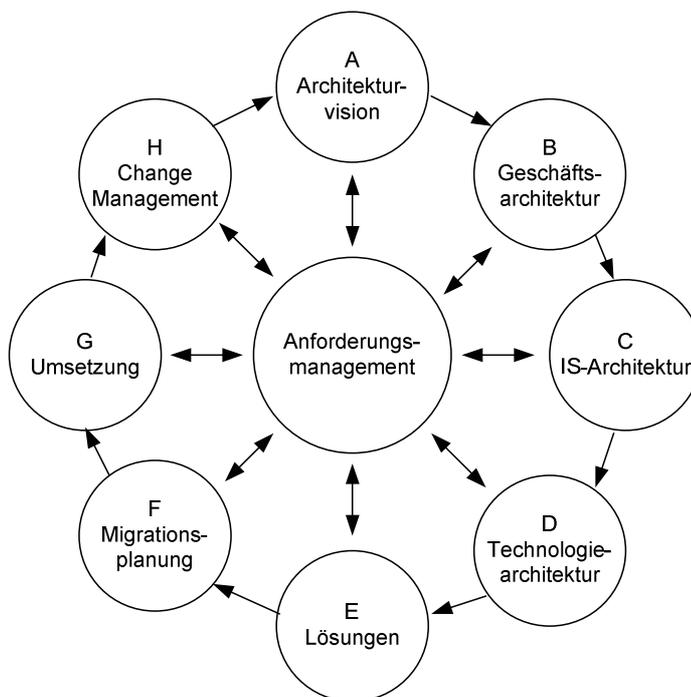


Abbildung 31: TOGAF ADM

In der Phase „Architekturvision“ werden zunächst bereits existierende Architekturen im Hinblick auf eine mögliche Wiederverwendung betrachtet. Jede Phase verwendet und erzeugt Ergebnisse. In der ersten Phase wird ein „Request for Architecture Work“ als Input verwendet. Dabei handelt es sich um ein Dokument, das Informationen über Geschäftsziele, strategische Geschäftspläne und verfügbare Ressourcen enthält.

In der zweiten Phase werden die Ist- und Soll-Geschäftsarchitektur modelliert. TOGAF schlägt dafür die Verwendung von Industriestandards vor, wie beispielsweise BPMN²⁹⁹, IDEF³⁰⁰ und UML³⁰¹. Die Geschäftsarchitektur enthält zum einen die Aufbauorganisation mit den Standorten der einzelnen Organisationseinheiten. Zum anderen werden in der Geschäftsarchitektur die Vision und Geschäftsziele jeder Organisationseinheit in textueller Form dokumentiert. Neben den Organisationseinheiten werden auch die Geschäftsfunktionen mit unterschiedlichen Detaillierungsgraden abgebildet. Dafür werden IDEF0 und UML-Use-Case-

²⁹⁹ BPMI (2002).

³⁰⁰ NIST (1993).

³⁰¹ OMG (2005).

Diagramme vorgeschlagen. Weitere Elemente der Geschäftsarchitektur sind Leistungen, die das Unternehmen sowohl externen als auch internen Kunden anbietet, sowie Ist- und Soll-Geschäftsprozesse. Für die Modellierung der Geschäftsprozesse werden beispielsweise BPMN, IDEF3 und UML-Aktivitätsdiagramme vorgeschlagen. Zuletzt werden noch Rollen sowie die Beziehung zwischen Organisationseinheiten und Geschäftsfunktionen abgebildet. Unter einer Rolle wird dabei ein Mitarbeiter des Unternehmens verstanden, der bestimmte Qualifikationen besitzt. Die Gegenüberstellung von Organisationseinheiten und Geschäftsfunktionen in Form von Matrizen soll deutlich machen, welche Organisationseinheiten für die Durchführung welcher Geschäftsfunktionen verantwortlich sind.

Im dritten Schritt findet die Abbildung der Informationssystemarchitektur statt. Diese wird weiter in eine Daten- und Applikationsarchitektur unterteilt. Die Datenarchitektur spezifiziert die wesentlichen Daten und Datenquellen auf einem hohen abstrakten Niveau. Als mögliche Modellierungstechniken werden konzeptionelle Datenmodellierung und logische Datenmodellierung genannt. Zudem werden die identifizierten Datenobjekte und Geschäftsfunktionen einander gegenübergestellt, um zu veranschaulichen, welche Datenobjekte von welchen Funktionen manipuliert werden, und um mögliche Schnittstellen zu identifizieren. Nach der Datenarchitektur wird die Applikationsarchitektur betrachtet. Dabei werden die wesentlichen Applikationen spezifiziert, die für die Datenverarbeitung und die Unterstützung der Geschäftsprozesse benötigt werden. Als Techniken werden wiederum Matrizen für die Gegenüberstellung von Applikationen und Geschäftsfunktionen vorgeschlagen.

Nach der Abbildung der Informationssystemarchitektur folgt im nächsten Schritt die Abbildung der Technologiearchitektur. Diese bildet die Grundlage für die anschließende Umsetzung der Soll-Architektur. Sie beschreibt die Netzwerk-Infrastruktur sowie Hardware- und Softwareplattformen des Unternehmens. Für jede Plattform werden Informationen, wie beispielsweise Verantwortlichkeiten, unterstützte Geschäftsfunktionen und Organisationseinheiten, unterstützte Applikationen, Standorte und Netzwerke, erhoben. Der Grad der Detaillierung für die Beschreibung der einzelnen Komponenten wird von TOGAF nicht definiert.

In den drei nachfolgenden Phasen des Vorgehensmodells wird nach möglichen Lösungen gesucht, darauf folgt eine Migrationsplanung und zuletzt die Umsetzung sowie das Change Management.

Die Verwendung von Referenzmodellen und Richtlinien kann als wichtige Aktivität im Rahmen der Entwicklung einer Unternehmensarchitektur betrachtet werden. Dafür stellt TOGAF zusätzlich zur ADM eine Sammlung von Referenzmodellen, Architekturmustern und Architekturbeschreibungen zur Verfügung. Diese können von Unternehmen für die Entwicklung ihrer eigenen Architekturen herangezogen werden. Allerdings sind sie rein auf technische Aspekte bezogen.

TOGAF besteht im Wesentlichen aus den drei Teilen ADM, Enterprise Continuum und Resource Base. Für keines dieser Teile liegt ein Metamodell vor, das eine konsistente (Wieder)Verwendung von Elementen gewährleistet.

Obwohl TOGAF die unterschiedlichen Inputs und Ergebnisse der einzelnen Phasen der ADM definiert, liegen keine Ergebnisdokumente vor, die die Ergebnisse strukturiert beschreiben. Die Ergebnisse bestehen meist aus Prinzipien und Richtlinien. Durch die Anwendung bestimmter Techniken, wie z.B. ER-Modellierung, werden zwar eindeutige Ergebnisdokumente erzeugt (ER-Diagramm), die Vorgehensweise von TOGAF definiert allerdings nicht, in welcher Phase des Entwicklungsprozesses welche Ergebnisdokumente erzeugt werden.

TOGAF definiert keine spezifischen Modellierungssprachen oder -techniken für eine Ebene, sondern beschreibt lediglich exemplarisch einige Modellierungssprachen, die für die Modellierung der Architekturebenen verwendet werden können. Für einige Entwicklungsphasen der TOGAF ADM werden Techniken vorgeschlagen. So werden beispielsweise in der Phase „Informationssystemarchitektur“ (Schritt 2 Architekturmodell) ER-Diagramme für die Modellierung von Sichten auf die Datenarchitektur und Entitäten-Geschäftsfunktionen-Matrizen für die Abbildung von Beziehungen oder Geschäftsszenarios verwendet.

Weitere Bestandteile, die durch das Framework abgebildet werden können, sind Informationsflüsse, die den Austausch von Daten zwischen Applikationen und den unterstützenden Softwarekomponenten beschreiben, sowie Geschäftsobjekte und deren Zustand, die von Geschäftsprozessen oder Applikationen verwendet werden, um „reale“ Daten, wie z.B. Kunden, Produkte und Verträge, zu beschreiben.

3.3.3.3 Inhaltliche Charakterisierung

Seit der Version 8.0 geht TOGAF in die Richtung eines ganzheitlichen Ansatzes, der nicht mehr nur die Abbildung der IT-Architektur unterstützt, sondern auch Elemente der Geschäftsarchitektur umfasst. Dennoch fehlt gerade auf der Ebene der Geschäftsarchitektur noch eine umfassende und formale Beschreibung der strategischen Ausrichtung eines Unternehmens. Der Fokus von TOGAF liegt deshalb auch in der aktuellen Version immer noch zu stark auf der Abbildung der Technologiearchitektur.

Das Kriterium der Strukturierung wird von TOGAF zumindest im Hinblick auf die Einteilung in unterschiedliche Architekturebenen erfüllt. Allerdings werden keine speziellen Sichten für die unterschiedlichen Architekturebenen definiert.

Einer der wesentlichsten Mängel von TOGAF ist, dass die Konsistenz zwischen den Architekturebenen und unterschiedlichen, vom Anwender bzw. Architekten selbst zu definierenden Sichten, von der Methode nicht gewährleistet wird. TOGAF definiert kein zugrunde liegendes Metamodell, das die wesentlichen Metaentitätstypen auf unterschiedlichen Architekturebenen zueinander in Beziehung setzt. Zwar werden verschiedene Modellierungssprachen vorge-

schlagen, wie z.B. UML und BPMN, deren Syntax durch ein Metamodell spezifiziert ist. Allerdings werden dadurch nicht die Abhängigkeiten zwischen den mit der jeweils verwendeten Modellierungssprache erstellten Modellen spezifiziert. Der Anwender ist folglich mit dem Problem konfrontiert, selbst die Abhängigkeiten zwischen den Modellen zu identifizieren und deren Konsistenz zu überprüfen.

Das Kriterium der Verständlichkeit der verwendeten Notationen kann nicht beurteilt werden, da TOGAF nicht die Verwendung bestimmter Modellierungssprachen vorschreibt. Die Erfüllung dieses Kriteriums hängt davon ab, welche Modellierungssprachen der Anwender im konkreten Fall auswählt.

TOGAF gibt keine konkreten Hinweise, mit welchem Detaillierungsgrad bestimmte Sachverhalte abgebildet werden sollten. Somit liegt es im Ermessen des Architekten, die passenden Modellierungssprachen und einen sinnvollen Detaillierungsgrad zu wählen.

Aufgrund der zuvor genannten Freiheitsgrade, die TOGAF dem Anwender zugesteht, besitzt die Methode eine hohe Flexibilität. Dies geht allerdings auf Kosten der Konsistenz der erzeugten Ergebnisse. Zudem werden viele Entscheidungen bei der konkreten Modellierung dem Anwender selbst überlassen.

Aus den vorhergehenden Punkten lässt sich schliessen, dass TOGAF einen sehr niedrigen Formalisierungsgrad aufweist. TOGAF definiert keine bestimmten Modellierungssprachen und Sichten sowie kein zugrunde liegendes Metamodell.

Insgesamt betrachtet ist die Dokumentation von TOGAF zwar sehr umfangreich. Dieser mangelt es aber sehr häufig an konkreten Hinweisen, wie die Modellierung der Architektur praktisch erfolgen soll. Sie bietet dem Anwender somit bei entscheidenden Problemen oftmals nicht ausreichend Unterstützung.

3.4 Zusammenfassung und Übersicht der Bewertung

Zusammenfassend ist festzuhalten, dass alle zuvor beschriebenen Ansätze eine Hierarchie von Modellierungsebenen zur Strukturierung der einzelnen Modelle verwenden. Des Weiteren verfolgen alle Ansätze das Prinzip, dass die Informationssysteme eines Unternehmens entsprechend den fachlichen Anforderungen gestaltet sein müssen. Bei einem Top-Down-Vorgehen schränken somit die Ergebnisse der Modellierung einer Ebene jeweils die Gestaltungsmöglichkeiten auf der darunter liegenden Ebene ein, wobei gleichzeitig die Implementierungsnähe zunimmt.

Konsolidiert man die in den verwandten Ansätzen beschriebenen Ebenen, so erhält man im Wesentlichen die vier Ebenen „Strategieebene“, „Organisationebene“, „Applikationsebene“ und „IT-Ebene“. Die vier genannten Ebenen finden sich nicht in allen Ansätzen. ARIS sowie Zachman und FEAF definieren nicht explizit eine Strategieebene. SOM und TOGAF bilden die Strategieebene lediglich in natürlichsprachlichem Fliesstext ab. Einige Ansätze wiederum

bilden die Sachverhalte einer Ebene nur sehr vereinfachend ab. ARIS und Zachman sind sehr stark von softwareentwicklungsrelevanten Aspekten dominiert. Andere, wie beispielsweise TOGAF, sind dagegen stark auf die detaillierte Dokumentation der IT-Architektur fokussiert.

ARIS integriert zwar wichtige Sichten auf den Geschäftsprozess, für die auch eine umfassende Werkzeugunterstützung verfügbar ist, doch ist ARIS nur beschränkt hierarchisierbar, da die Ebenen an den Phasen des Entwicklungsprozesses ausgerichtet sind (Fachkonzept, DV-Konzept und Implementierung) und somit zu stark auf softwareentwicklungsrelevante Aspekte fokussiert. Zudem beziehen sich die fünf Sichten lediglich auf den Geschäftsprozess. Andere Sichten auf das Unternehmen, wie z.B. Geschäftsziele und Geschäftsnetzwerk, werden vernachlässigt.

Zusätzlich zu den Ebenen werden in den meisten Ansätzen Sichten eingeführt, um die Komplexität der abzubildenden Sachverhalte auf einer Ebene zu reduzieren. Die Anzahl der Sichten pro Ebene variiert bei den einzelnen Ansätzen sehr stark. Aufgrund der Prozessorientierung bietet ARIS auf Ebene der Geschäftsprozesse sechs verschiedene Sichten. SOM definiert dagegen lediglich zwei Sichten. Der Bezugsrahmen von Zachman differenziert auf allen Ebenen nach sechs Sichten, wodurch eine Matrix zur Strukturierung der Artefakte entsteht.

Die beschriebenen Bezugsrahmen aus der Praxis unterscheiden zwischen zahlreichen Teilsichten, die allerdings eine reine Aufzählung darstellen und nicht durch ein umfassendes Metamodell spezifiziert und zueinander in Beziehung gesetzt sind, so dass die Konsistenz zwischen den Modellen der unterschiedlichen Teilsichten nicht gewährleistet werden kann. Ebenso wird nicht festgelegt, welche Modellierungssprachen oder -techniken für die einzelnen Sichten verwendet werden sollen. Es werden lediglich Beispiele angegeben. Zachman unterscheidet beispielsweise 30 Teilsichten. Aufgrund der Vielfalt möglicher Perspektiven auf die Informationssysteme eines Unternehmens ist diese Anzahl sicher gerechtfertigt. Allerdings besteht dadurch die Gefahr, der ursprünglichen Intention einer Unternehmensarchitektur, nämlich der komplexitätsreduzierenden und damit anschaulichen und verständlichen Abbildung des Unternehmens und seiner Informationssysteme, entgegenzuwirken. Unternehmensmodelle sind nicht vordergründig auf die detaillierte Beschreibung für Entwickler von Informationssystemen ausgerichtet, sondern sollen in erster Linie dazu dienen, auch Personen aus den Fachbereichen eine anschauliche Beschreibung wichtiger Sachverhalte und Zusammenhänge in einem Unternehmen zu liefern und damit den Diskurs über das Unternehmen zu fördern. Deshalb sollte eine möglichst breite Darstellung der Unternehmensstrukturen mit einem angemessenen Detaillierungsgrad angestrebt werden. Darüber hinaus definieren die genannten Bezugsrahmen zur Strukturierung der Unternehmensmodelle bis auf TOGAF keine Vorgehensweise, die die systematische Erstellung der unterschiedlichen Sichten auf das Unternehmen ermöglicht.

Mit Ausnahme von ARIS und MEMO verfügen die betrachteten Ansätze über kein oder ein nur wenig ausgearbeitetes Metamodell, in dem die Beschreibungskonzepte der verschiedenen

Sichten zueinander in Beziehung gesetzt werden. Für die Strategieebene definieren die meisten Ansätze keine Metamodelle. Das Objekt- und das Zielsystem des Unternehmensplans der SOM-Methode werden beispielsweise lediglich in natürlichsprachlichem Fliesstext dargestellt. Die Modellierungssprachen der SOM-Methode sind häufig für mehrere Modelle einer Ebene spezifiziert, so dass das Metamodell Modellierungskonstrukte enthalten kann, die in einem Modell nicht verwendet werden.

In ARIS liegen für alle Sichten auf den Geschäftsprozess Metamodelle vor, die zu einem integrierten Metamodell verknüpft werden können. ARIS schreibt aber keine bestimmten Metamodelle für die Ausgestaltung der Modellierungssprachen anderer Ebenen vor. MEMO verfügt zwar über ein Metamodell und eine Spracharchitektur für die verwendeten Modellierungssprachen. Allerdings verwendet MEMO, wie die Bezugsrahmen aus der Praxis, eine Matrix zur Festlegung der Ebenen (Perspektiven) und Sichten. Dies vereinfacht zwar die Definition der Beziehungen zwischen Elementen auf unterschiedlichen Ebenen, dadurch wird aber gleichzeitig die Flexibilität im Hinblick auf die Definition unterschiedlicher Sichten auf unterschiedlichen Ebenen eingeschränkt, da die Sichten bzw. Spalten für alle Ebenen der Matrix festgelegt sind. Ausserdem kann es vorkommen, dass bei der Definition einer neuen Sicht nicht alle Zellen „ausgefüllt“ bzw. modelliert werden können.

Tabelle 6 zeigt zusammenfassend den Erfüllungsgrad der inhaltlichen sowie strukturellen Kriterien durch die zuvor betrachteten Ansätze. Darin ist ersichtlich, dass keiner der Ansätze alle Methodenbestandteile umfasst und jeder Ansatz unterschiedliche Stärken und Schwächen in Bezug auf die inhaltlichen Kriterien aufweist. Auffallend ist, dass die in der Praxis weit verbreiteten Bezugsrahmen (Framework) die Kriterien am wenigsten erfüllen, insbesondere die Kriterien der Ganzheitlichkeit, Konsistenz und Formalisierung.

Methode/ Framework	Inhaltliche Charakteristika							Strukturelle Charakteristika				
	Ganzheitlichkeit	Strukturierung	Konsistenz	Verständlichkeit	Angemessenheit	Flexibilität	Formalisierung	Vorgehen	Rollen	Ergebnisse	Techniken	Metamodell
ARIS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MEMO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							
SOM	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							
FEAF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TOGAF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Zachman	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Legende: erfüllt/vorhanden teilweise erfüllt/vorhanden nicht erfüllt/vorhanden

Tabelle 6: Übersicht über die Bewertung existierender Ansätze

Aufgrund dieser Schwächen wird in Kapitel 4 ein eigener Ansatz vorgeschlagen, der die Stärken der bewerteten Ansätze berücksichtigt und vor allem die Kriterien der Ganzheitlichkeit, Konsistenz und Formalisierung adressiert. Der Fokus des eigenen Ansatzes liegt deshalb auf der Entwicklung eines möglichst umfassenden bzw. ganzheitlichen, (semi-)formal definierten Metamodells, das auch die ebenen- und sichtenübergreifenden Abhängigkeiten berücksichtigt.

4 Ansatz zur Modellierung der Unternehmensarchitektur

In den folgenden vier Abschnitten dieses Kapitels wird ein Ansatz zur Modellierung der Unternehmensarchitektur bestehend aus den Methodenelementen Vorgehensmodell, Metamodell und Ergebnisdokumente auf Strategie-, Organisations-, Informationssystem- und Infrastrukturebene präsentiert. Die folgende Abbildung 32 visualisiert überblicksartig die einzelnen Gestaltungsebenen und Sichten des entwickelten Ansatzes.

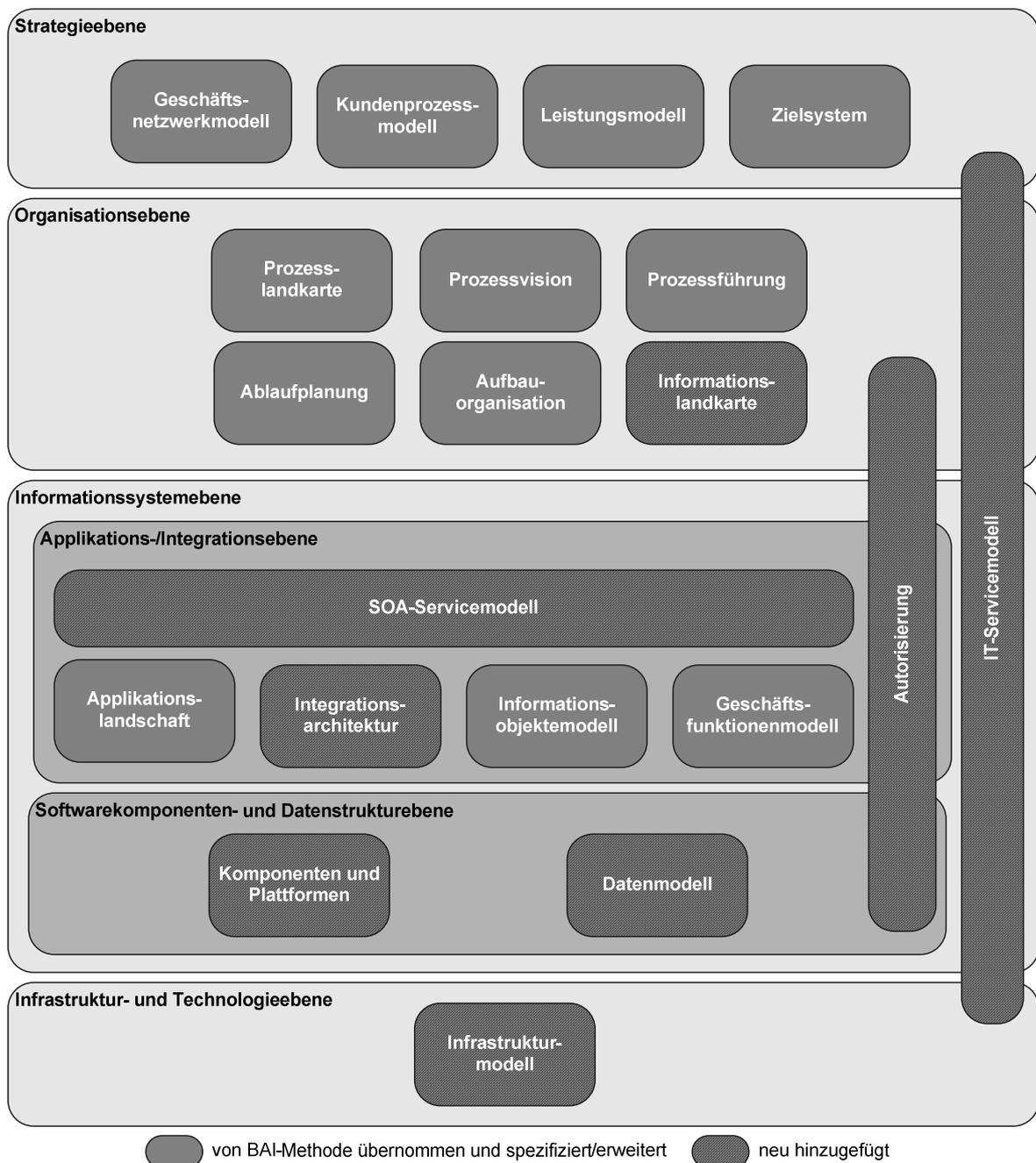


Abbildung 32: Gestaltungsebenen und Sichten des entwickelten Ansatzes

Als Ausgangsbasis für die Entwicklung des in der vorliegenden Arbeit präsentierten Ansatzes dient die bereits existierende BAI-Methode.³⁰² Diese wird einerseits genauer spezifiziert und erweitert anhand der zuvor beschriebenen verwandten Methoden und Frameworks (vgl. Kapitel 3) sowie andererseits anhand der Ergebnisse einer Literaturanalyse (vgl. Tabelle 7), die unterschiedliche Teilbereiche der Unternehmensarchitektur abdeckt (z.B. Strategiegestaltung, Prozessmanagement, Applikationsintegration, Autorisierung, Servicemanagement). In Abbildung 32 sind die im Vergleich zur BAI-Methode neu hinzugefügten Sichten schraffiert dargestellt. Nicht schraffiert dargestellte Sichten sind von der BAI-Methode übernommen und werden genauer spezifiziert und/oder erweitert.

Domäne	Primär verwendete Literaturquellen
Strategiegestaltung	<i>Böh/Meyer (2004), Buhl et al. (1999), Grant (1998), Hahn (1997), Harengel (2000), Heinrich/Winter (2004), Kaplan/Norton (1997), Kühn/Karagiannis (2005), Leidecker/Bruno (1984), Leist/Winter (2002), Mintzberg (1987), Müller-Stewens/Lechner (2003), Österle et al. (2001), Scholz (1998), Winter (2003d), Winter (2003b)</i>
Prozess- und Organisationsgestaltung	<i>Brecht (2002), Hess/Brecht (1996), IMG (1997), Jonkers et al. (2004), Junginger et al. (2000), Kühn/Karagiannis (2005), Legner (1999), Nüttgens/Rump (2002), Oestereich et al. (2004), Österle (1995), Rosemann/zur Mühlen (1997), Scheer (1998), Scheer (2001), Scheer/Thomas (2005), Strauch (2002), Vogler (2003), Winter (2003d), Winter (2004a), Zellner (2003), Wortmann (2005)</i>
Informationssystemgestaltung und Autorisierung	<i>Ferraiolo et al. (2001), Greenfield/Short (2003), IBM-Corporation (1984), IMG (1999), Jeckle et al. (2004b), Jonkers et al. (2004), Kern et al. (2002), Kühn/Karagiannis (2005), OMG (2003a), OMG (2005), Samarati/de Capitani di Vimercati (2002), Schwinn (2005), Seufert (2001), Winter (2003a), Wortmann (2005)</i>
Serviceorientiertes IT-Management/ Serviceorientierte Architektur (SOA)	<i>Arsanjani (2004), Böhmman et al. (2002), Commerce (2000), Dan et al. (2004), Hafner et al. (2006), Hegering et al. (1999), Hochstein et al. (2004), Hochstein/Hunziker (2003), Mayerl et al. (2005), Schelp et al. (2004), Schemm et al. (2006), Stojanovic (2005), Winter/Schelp (2005), W3C (2004a), Zarnekow et al. (2004)</i>

Tabelle 7: Domänenspezifische Literaturquellen

Wie eingangs ausgeführt (vgl. Abschnitt 1.2), stellt diese Arbeit lediglich einen Ansatz und keine vollständige Methode³⁰³ zur Gestaltung der Unternehmensarchitektur dar. Es wird nicht auf einzelne Techniken sowie die ausführenden Rollen eingegangen, da die Beschreibung dieser Methodenelemente zu umfangreich wäre und den Rahmen dieser Arbeit sprengen würde. Der Fokus liegt auf der Definition der Metamodelle, deren wesentlichen Metaentitätstypen und Zusammenhänge sowie auf den wesentlichen Aktivitäten des Vorgehensmodells (lediglich Level 1). Die einzelnen Vorgehensmodelle werden in Form von UML-Aktivitätsdiagramm-

³⁰² Vgl. *Choinowski et al. (2003)* und Abschnitt 2.3.4.

³⁰³ Vgl. hierzu die in Abschnitt 2.1.2 beschriebenen Methodenbestandteile.

men abgebildet (vgl. Abschnitt 2.2.4.2). Für die Darstellung der Metamodelle werden UML-Klassendiagramme³⁰⁴ verwendet (vgl. Abschnitt 2.2.4.1).

Das Metamodell ist das konzeptionelle Datenmodell der Ergebnisse einer Methode, das die Bestandteile und deren Zusammenhänge aller Entwurfsergebnisse gesamthaft darstellt.³⁰⁵ Es besteht aus den Elementen Metaentitätstyp, Attribut und Beziehungstyp.³⁰⁶ Metaentitätstypen sind Bestandteile von Entwurfsergebnissen. Beziehungstypen beschreiben die logische Verbindung zwischen Metaentitätstypen. Attribute sind Eigenschaften von Metaentitätstypen.

Durch die Darstellung einer grösseren Zahl an Metaentitätstypen sowie der Beziehungstypen wird das Metamodell rasch unübersichtlich. Aus diesem Grund wird das Metamodell in problemorientierte Sichten bzw. (Teil-)Modelle unterteilt, welche thematisch zusammengehörende Metaentitätstypen zusammenfassen.³⁰⁷ Die problemorientierten Sichten müssen nicht zwingend disjunkt sein.³⁰⁸

Das Metamodell umfasst neben der grafischen Darstellung der Metaentitätstypen und ihrer Beziehungen auch eine detaillierte verbale Beschreibung, um mögliche Fehlinterpretationen zu vermeiden. Zur Beschreibung der Metaentitätstypen wird folgende Struktur angewendet:

- Bezeichnung: Name des Metaentitätstyps,
- Definition: genaue Beschreibung des Metaentitätstyps,
- Beziehungen zu: Beschreibung der Beziehungstypen zu anderen Metaentitätstypen.

Aus Gründen der Übersichtlichkeit wird auf die Definition von Attributen für die einzelnen Metaentitätstypen verzichtet. Ausserdem ist es sinnvoll, die Attribute erst bei der Anwendung in einem unternehmensspezifischen Umfeld zu definieren.

Das Vorgehensmodell des hier vorgestellten Ansatzes verfolgt entsprechend der Vorgehensweise im Business Engineering einen Top-Down-Ansatz³⁰⁹, wengleich die Ebenenbildung keine ausschliessliche Top-Down-Vorgehensweise erfordert.³¹⁰ Demzufolge impliziert eine Veränderung innerhalb der Strategieebene die Ausrichtung der Organisationsebene, diese anschliessend die Umgestaltung der Applikationsebene und diese wiederum die Umgestaltung der Softwarekomponenten- und Datenstrukturebene.

³⁰⁴ Vgl. *OMG* (2003b).

³⁰⁵ Vgl. *Gutzwiller* (1994), S. 14 und Abschnitt 2.1.2.

³⁰⁶ Vgl. *Gutzwiller* (1994), S. 24.

³⁰⁷ Vgl. *Gutzwiller* (1994), S. 24.

³⁰⁸ Vgl. *Kaiser* (2000), S. 80.

³⁰⁹ Vgl. *Österle* (1995), S. 24.

³¹⁰ Vgl. *Österle/Blessing* (2003), S. 77.

4.1 Modellierung auf Strategieebene

Die Strategieebene stellt die oberste Gestaltungsebene bei der Entwicklung einer Unternehmensarchitektur dar (vgl. Abschnitt 2.3.1 und 2.3.2). Bei ihrer Gestaltung müssen die Potenziale der Informationstechnologie erkannt und deren Umsetzungsrestriktionen berücksichtigt werden. Wesentliche Anforderungen an einen Ansatz für die Modellierung auf Strategieebene sind:³¹¹

- Abbildung des Zusammenwirkens von Unternehmungen bzw. Geschäftseinheiten zur gemeinsamen Abdeckung des Kundenprozesses,
- Beschreibung der wesentlichen Elemente der strategischen Ausrichtung eines Unternehmens bzw. einer Geschäftseinheit,
- Operationalisierung der Strategie durch Formulierung entsprechender Ziele und Kennzahlen.

Als Bezugseinheiten für die Strategiegestaltung kommen Unternehmen, einzelne Geschäftseinheiten oder Rollen im Geschäftsnetzwerk in Betracht.

Da die im Kapitel zuvor beschriebenen Ansätze zur Gestaltung der Unternehmensarchitektur nur eine sehr geringe Menge an Metaentitätstypen und Sichten für die Abbildung der strategischen Ausrichtung eines Unternehmens bieten, wird für die Beschreibung der Modelle, Metaentitätstypen und deren Beziehungen auf Strategieebene primär auf die BAI-Methode sowie die in Tabelle 7 enthaltenen Literaturquellen zurückgegriffen.

Geschäftsstrategien werden bisher in der Praxis nur selten systematisch entwickelt und lediglich in natürlicher Sprache dokumentiert und kommuniziert.³¹² Im Hinblick auf die ständige Erweiterung und schnelle Änderung von Strategien in der heutigen Geschäftswelt birgt diese Vorgehensweise die Gefahr, dass die Geschäftsstrategie eines Unternehmens unvollständig dokumentiert ist und Unklarheiten sowie Inkonsistenzen aufweist. Dies kann durch den Einsatz von zumindest semi-formalen Modellen und damit durch eine strukturierte und standardisierte Dokumentation vermieden werden.

Im Unterschied zu ARIS und MEMO (vgl. Abschnitt 3.2.1 und 3.2.2) wird auf Strategieebene nicht auf das Konzept der Wertschöpfungskette nach PORTER³¹³ zurückgegriffen. Aufgrund der Positionierung der vorliegenden Arbeit im BE orientieren sich die Modelle zur Strategiegestaltung an der Kundenprozessorientierung und der Bildung von Geschäftsnetzwerken. Nach ÖSTERLE ist das Unternehmen des Informationszeitalters nicht mehr produkt-, sondern

³¹¹ Vgl. Winter (2004b), Kühn/Karagiannis (2005), S. 1494 sowie Abschnitt 2.3.1 und 2.3.2.

³¹² Vgl. Heinrich/Winter (2004), S. 2.

³¹³ Vgl. Porter (1999).

kundenorientiert.³¹⁴ Es muss das Kundenbedürfnis umfassend abdecken und dem Kunden so viele zusammenhängende Teilprobleme wie möglich abnehmen, also den gesamten Prozess des Kunden bei seiner Problemlösung abdecken. Dazu ist die Integration von Leistungen notwendig, die von unterschiedlichen Geschäftspartnern bezogen werden. Das Ergebnis sind die Aufspaltung der Wertschöpfungskette und die Bildung von Geschäftsnetzwerken. Die lineare vertikale Wertschöpfungskette hat sich so zu einem verzweigten Wertschöpfungsnetzwerk gewandelt, in dem unterschiedliche Unternehmen und Geschäftseinheiten Teilaufgaben in der Wertschöpfungskette übernehmen.³¹⁵

Auf der Strategieebene können folgende Ergebnisdokumententypen unterschieden werden:

- Das **Geschäftsnetzwerk** beschreibt das Zusammenwirken von Unternehmen bzw. Geschäftseinheiten in einem Wertschöpfungsnetzwerk zur gemeinsamen Abdeckung von bestimmten Kundenbedürfnissen.³¹⁶
- Das **Kundenprozessmodell** beschreibt die Schritte, die der Kunde durchläuft, um ein bestimmtes Bedürfnis zu befriedigen, sowie die zur Bedürfnisbewältigung notwendigen Serviceleistungen.³¹⁷
- Das **Leistungsmodell** legt die strategische Ausrichtung eines Unternehmens bzw. einer Geschäftseinheit in Bezug auf einen bestimmten Stichtag anhand von Ausprägungen bestimmter vorgefundener (z.B. Markt) und gestaltbarer (z.B. Produkte, Organisation) Dimensionen fest.³¹⁸
- Das **Zielsystem** systematisiert in Form einer Balanced Scorecard (BSC) die zur Erreichung der Ziele notwendigen zentralen strategischen Erfolgsfaktoren eines Unternehmens bzw. einer Geschäftseinheit und operationalisiert diese durch die Definition entsprechender Kennzahlen.³¹⁹

Im folgenden Abschnitt wird das Vorgehen zur Erstellung der zuvor genannten Ergebnisdokumente beschrieben.

³¹⁴ Zur Aufspaltung der Wertschöpfungskette und der Bildung von Wertschöpfungsnetzwerken vgl. *Österle* (2003), *Buhl et al.* (1999), *Leist/Winter* (2000) und *Scholz* (1998).

³¹⁵ Vgl. *Österle* (2003), S. 32-34.

³¹⁶ Vgl. *Winter* (2003d), S. 96 und *Österle* (2003), S. 32-34.

³¹⁷ Vgl. *Heinrich* (2002b), S. 84 und *Österle et al.* (2001), S. 45-51.

³¹⁸ Zur Definition des Leistungsmodells vgl. *Heinrich* (2000), *Heinrich* (2002a) und *Heinrich/Winter* (2004). HEINRICH verwendet allerdings im Unterschied zu dieser Arbeit den Begriff Geschäftsmodell.

³¹⁹ Vgl. *Kaplan/Norton* (1997) und *Harengel* (2000).

4.1.1 Vorgehensmodell

Das Vorgehensmodell zur Strategiegestaltung besteht auf Level 1 aus vier wesentlichen Aktivitäten (vgl. Abbildung 33):³²⁰

Zunächst wird das *Geschäftsnetzwerk festgelegt*. Dabei werden den Unternehmen bzw. Geschäftseinheiten entsprechende Rollen im Geschäftsnetzwerk zugeordnet. Zusätzlich sollten die wichtigsten Leistungsverflechtungen abgebildet werden. Ausgangspunkt für die Gestaltung eines Geschäftsnetzwerks sind bestimmte Kundenprozesse, hinter denen in den meisten Fällen auch spezifische Kundensegmente stehen. Ausgewählte Kundenprozesse werden durch Service Integrators ganzheitlich unterstützt, wobei ein Kundenprozess (bzw. Kundensegment) auch durch mehrere Service Integrators unterstützt werden kann.

Im nächsten Schritt sind die abzudeckenden *Kundenprozesse zu analysieren*. Diese bilden die Grundlage für die detaillierte Leistungsbeschreibung sowie für die auf Organisationsebene zu definierenden Geschäftsprozesse. Zunächst werden bei der Kundenprozessanalyse die verschiedenen Teilaspekte oder Teilschritte des zu unterstützenden Kundenprozesses und deren Abfolge strukturiert. Darauf aufbauend wird für jeden Teilaspekt bzw. Teilschritt geprüft, welche Teilleistungen erbracht werden müssen, um den jeweiligen Teilaspekt bzw. Teilschritt zu unterstützen. In nachfolgenden Schritten ist zu prüfen, ob die jeweiligen Teilleistungen auf Grundlage des Geschäftsmodells des jeweiligen Unternehmens bzw. der jeweiligen Geschäftseinheit selbst erbracht werden sollten oder besser fremdbezogen werden. Schliesslich sind geeignete Bündelungen und Vertriebskanäle zu identifizieren, die sich auf die eigenerstellten Teilleistungen, aber auch auf zu integrierende Fremdleistungen beziehen.

Im dritten Schritt wird das *Leistungsmodell* sowohl aus einer exogenen als auch aus einer endogenen Perspektive *entwickelt*. Bei der Spezifikation der exogenen Perspektive stehen Dimensionen zur Beschreibung der marktbezogenen strategischen Ausrichtung des Unternehmens bzw. der Geschäftseinheit im Mittelpunkt, wie beispielsweise Land/Region, Kunde, Kernprodukte und Leistungen, Marke und Vertriebsweg.³²¹ Die endogene Perspektive umfasst dagegen Dimensionen, die die interne Konstitution des Unternehmens bzw. der Geschäftseinheit beschreiben, wie beispielsweise Kompetenzfelder, Ziele, Partner und Organisationsstruktur.

Im letzten Schritt wird das *Zielsystem* des Unternehmens bzw. der Geschäftseinheit anhand einer Balanced Scorecard³²² *beschrieben*. Diese bildet die Grundlage für die Spezifikation der zur Prozessgestaltung benötigten Führungsgrössen und Erfolgsfaktoren. Die zentralen strategischen Erfolgsfaktoren eines Unternehmens werden nach den Perspektiven Finanzen, Kun-

³²⁰ Vgl. Choinowski et al. (2003), S. 16f.

³²¹ Vgl. Heinrich (2002a), S. 62.

³²² Vgl. Kaplan/Norton (1997).

den, Prozess und Potenzial systematisiert und anschliessend durch die Bildung perspektivenorientierter Kennzahlen operationalisiert.

Die zuvor genannten Aktivitäten des Vorgehensmodells auf Strategieebene sollten nacheinander durchgeführt und durch Konsistenzprüfungen ergänzt werden (vgl. hierzu die in Abschnitt 4.6.1 identifizierten Abhängigkeiten auf Strategieebene).

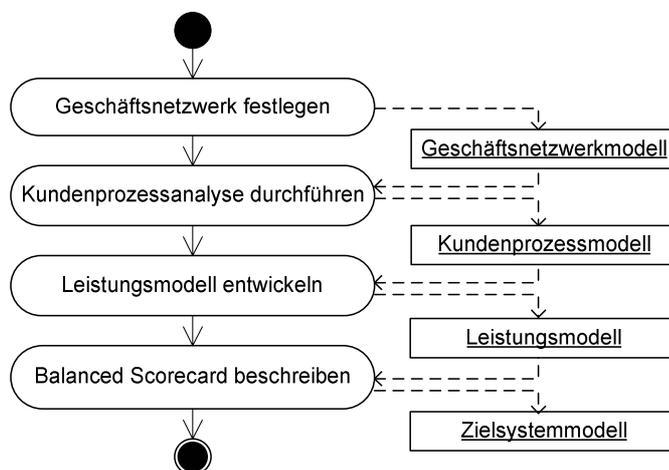


Abbildung 33: Vorgehensmodell auf Strategieebene (Level 1)

4.1.2 Metamodell

Das Metamodell der Strategieebene lässt sich entsprechend den bei der Strategiegestaltung erstellten Ergebnisdokumententypen in die Sichten Geschäftsnetzwerk, Kundenprozess, Leistungsmodell sowie Zielsystem (BSC) einteilen.

4.1.2.1 Metamodell des Geschäftsnetzwerks

In Geschäftsnetzwerken nimmt jedes Unternehmen bzw. jede Geschäftseinheit eine oder mehrere Rolle(n) wahr. Die **Rollen** orientieren sich dabei an den kompetenzbasierten Typen von Organisationseinheiten:³²³

Ein **Service Integrator** integriert (im Normalfall nicht selbst erstellte) Leistungskomponenten, um damit einen bestimmten Kundenprozess ganzheitlich zu unterstützen. Seine Kernkompetenz ist demzufolge die umfassende Kenntnis von Kundenbedürfnissen bzw. Kundenprozessen.

Ein **Shared Service Provider** produziert Leistungskomponenten für verschiedene andere Service Providers oder Service Integrators. Im Gegensatz zu Service Providern decken sie keine Bedürfnisse ganzheitlich ab, sondern spezialisieren sich auf die effiziente Herstellung spezifischer, aber standardisierter Leistungskomponenten in grossen Mengen.

³²³ Vgl. im Folgenden *Winter* (2003d), S. 53-54.

Ein **Exclusive Service Provider** erzeugt ebenfalls Leistungskomponenten für andere Service Providers oder Service Integrators. Im Gegensatz zu Shared Service Providers werden allerdings nicht grosse Mengen von Leistungskomponenten für mehrere Abnehmer produziert, sondern spezifischere Leistungskomponenten für einen oder wenige Abnehmer und damit in eher kleineren Mengen. An die Stelle von Kunden- bzw. Kundenprozessorientierung (Service Integrator) bzw. Kostenführerschaft (Shared Service Provider) treten als Haupterfolgskriterien Schnelligkeit, Flexibilität und/oder spezifische Prozesskompetenzen.

Ein **Public Service** stellt Leistungskomponenten bereit, die keinen spezifisch fachlichen Charakter haben, aber zur Abwicklung der Leistungserstellung unverzichtbar sind. Im engeren Sinne handelt es sich dabei um Dienste mit quasi-hoheitlicher Funktion wie z.B. Zertifizierung oder Authentifizierung. Im weiteren Sinne können auch andere elektronische Dienste wie z.B. Informationsdienste, Katalogdienste, Zahlungsabwicklung und Logistikabwicklung darunter subsumiert werden.

Die folgende Abbildung 34 zeigt das Metamodell des Geschäftsnetzwerkmodells. Im Geschäftsnetzwerkmodell werden die Geschäftseinheiten bzw. Unternehmen entsprechend der ihnen zugeordneten Rollen dargestellt. Zusätzlich sollten die wichtigsten Leistungsverflechtungen abgebildet werden. Jede Rolle erbringt und/oder bezieht eine oder mehrere Marktleistungen. Eine **Marktleistung** ist ein Produkt oder eine Dienstleistung, die ein Unternehmen oder eine Geschäftseinheit am Markt bzw. im Geschäftsnetzwerk anbietet.³²⁴ Sie wird durch das Zusammenwirken eines Bündels von Leistungen erbracht. Eine Marktleistung kann standardisiert oder individualisiert sein. **Standardisierte Marktleistungen** werden von Shared Service Providers in grossen Mengen für zahlreiche andere Service Provider oder Service Integrators erbracht. Merkmal von standardisierten Marktleistungen ist, dass die Leistung für einen fiktiven Durchschnittskunden (Dienstnehmer) erstellt wird. **Individualisierte Marktleistungen** werden dagegen von Exclusive Service Providern für einige wenige andere Service Provider oder Service Integrators in geringer Menge erbracht. Der Grad der Beteiligung (Integrationsgrad) des Dienstnehmers (Service Provider oder Integrator) ist bei der Erstellung individualisierter Marktleistungen höher, da sie auf die individuellen Bedürfnisse des Dienstnehmers zugeschnitten sind.

Ausgangspunkt für die Gestaltung eines Geschäftsnetzwerks sind bestimmte **Kundenprozesse**, hinter denen in den meisten Fällen auch spezifische **Kunden** und **Kundengruppen** stehen. Ausgewählte Kundenprozesse werden durch Service Integrators ganzheitlich unterstützt, wobei ein Kundenprozess auch durch verschiedene Service Integrators unterstützt werden kann. Bei Längsschnittbetrachtungen ist dies allgemein der Fall, da sich die Unterstützung oft auf Kundenprozesse bezieht, die nur selten oder gar einmalig auftreten: Das Kundensegment „wandert“ dann von einem prozessspezifischen Service Integrator zum Nächsten. Service

³²⁴ Vgl. Glossar in *IMG* (1997).

Integrators sowie Service Providers beziehen Leistungskomponenten von Shared Service Providers, Exclusive Service Providers und Public Services. Dabei werden im Normalfall mehrstufige Leistungsketten entstehen. Während Exclusive Service Provider nur einen oder wenige Service Provider bzw. Service Integrators beliefern und deshalb unter Umständen aus Effizienzgründen direkte Vernetzungen aufbauen, erfolgen alle anderen Vernetzungen im Normalfall über eine gemeinsame, offene Kollaborationsinfrastruktur, genannt **Business Collaboration Infrastructure (BCI)**.

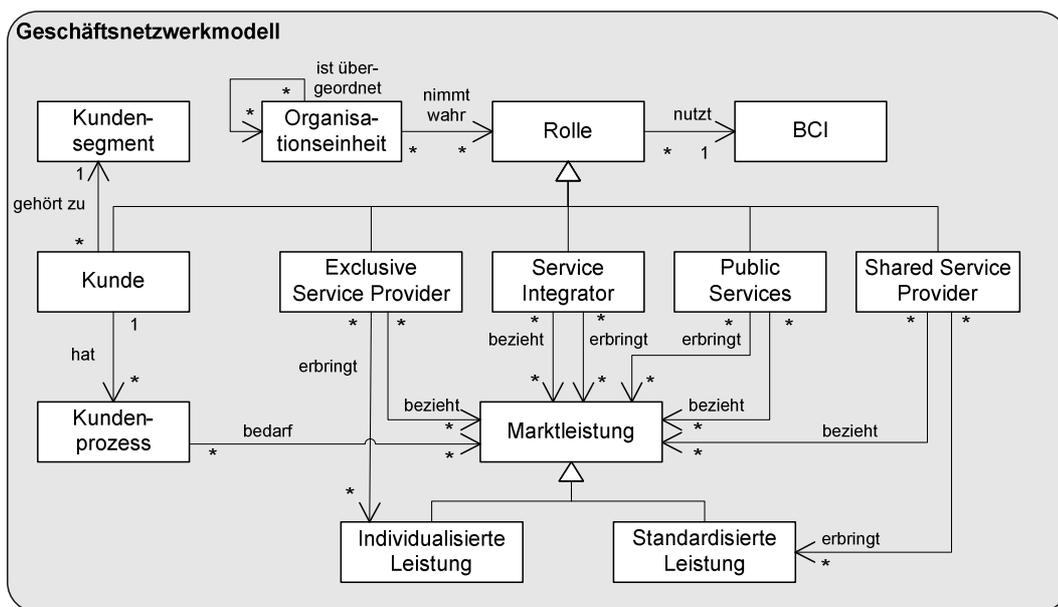


Abbildung 34: Metamodell des Geschäftsnetzwerkmodells

An die Stelle einer Vielzahl bilateraler Vernetzungen zwischen Geschäftsmodellen tritt dann jeweils nur ein einziger „Adapter“ zu dieser Kollaborationsinfrastruktur. Der direkte Zugriff von Konsumenten auf Leistungskomponenten von Service Providers ist – sinnvollerweise über die Kollaborationsinfrastruktur – möglich, wenn seitens der Konsumenten die Nutzung eines Service Integrator nicht erwünscht ist oder wenn die zu unterstützenden Prozesse so spezifisch sind, dass eine individuelle Kombination von Leistungskomponenten unausweichlich ist.

4.1.2.2 Metamodell des Kundenprozessmodells

Der **Kundenprozess** beschreibt, welche Aktivitäten der Kunde durchführt und in welcher Reihenfolge diese ablaufen. Für jede **Kundenaktivität** wird festgelegt, welche **Serviceaktivitäten** durchgeführt werden müssen, um die jeweilige Kundenaktivität zu unterstützen. Die einzelnen Serviceaktivitäten sowie deren Reihenfolge werden durch den **Serviceprozess** beschrieben. Das Kundenprozessmodell definiert zudem für jede Serviceaktivität, ob diese von dem jeweiligen Unternehmen bzw. der jeweiligen Geschäftseinheit selbst erbracht oder fremdbezogen wird. Zusätzlich werden in dem Kundenprozessmodell geeignete Bündelungen

und Vertriebskanäle identifiziert, die sich auf die eigenerstellten Teilleistungen, aber auch auf zu integrierende Fremdleistungen beziehen.

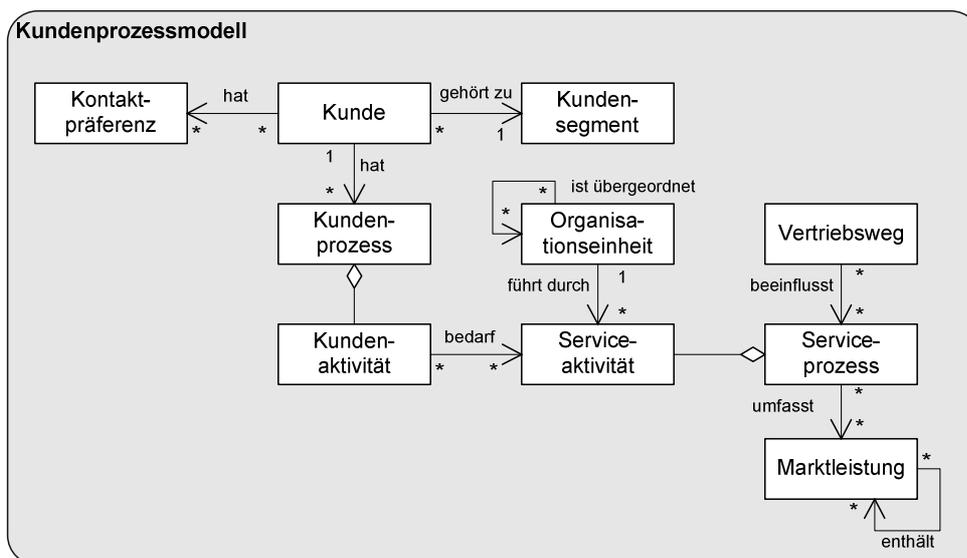


Abbildung 35: Metamodell des Kundenprozessmodells

Die Modellierung des Kundenprozesses und des korrespondierenden Serviceprozesses ermöglicht die systematische Planung der Leistungs- und Informationsflüsse zwischen Kunde, internen Geschäftseinheiten sowie einzubindenden Partnern.³²⁵ Zudem bietet die umfassende Unterstützung eines Kundenprozesses und damit eines Kundenbedürfnisses eine Chance zur Wettbewerbsdifferenzierung gegenüber Konkurrenten sowie zur Steigerung der Kundenloyalität.

4.1.2.3 Metamodell des Leistungsmodells

Das Leistungsmodell beschreibt die strategische Ausrichtung eines Unternehmens bzw. einer Geschäftseinheit in Bezug auf die Ausprägung bestimmter vorgefundener (z.B. Markt) und gestaltbarer (z.B. Produkte, Organisation) Dimensionen.³²⁶ Eine **Dimension** beschreibt einen Teilaspekt der strategischen Ausrichtung eines Unternehmens oder einer Geschäftseinheit. Jede Dimension besitzt eine Menge möglicher (meist branchenspezifischer) **Ausprägungen**. Dadurch steht eine Beschreibungsstruktur für die strategische Ausrichtung von Unternehmen oder Geschäftseinheiten zur Verfügung, die gegenüber der meist nur als Fliesstext vorliegenden konventionellen Beschreibungen den Vorteil hat, die strategische Ausrichtung systematisch kommunizieren und standardisiert dokumentieren zu können. Darüber hinaus ist es möglich, auf Grundlage einer ausreichend grossen Fallbasis Abhängigkeiten und Typisierungen zu ermitteln. Während **Abhängigkeitsregeln** helfen, bei der Strategiebildung auf mögliche Widersprüche aufmerksam zu machen, kann Typisierung dazu benutzt werden, den weiter oben

³²⁵ Vgl. Heinrich (2002b), S. 87-88.

³²⁶ Vgl. Heinrich (2000), S. 11.

beschriebenen Rollen in Geschäftsnetzwerken bestimmte Dimensionen und Ausprägungen im Sinne von Referenzmodellen zuzuordnen. Während beispielsweise die Spezifikation von Zielkundengruppen oder Vertriebskanal-Zuordnungen für Service Integrators einen hohen Stellenwert hat und in der Beschreibung von deren strategischer Ausrichtung besonders detailliert erfolgen sollte, ist aus Sicht von Exclusive Service Providers eher die Spezifikation von Produktionsverfahren oder Ressourcenmanagement von Interesse.

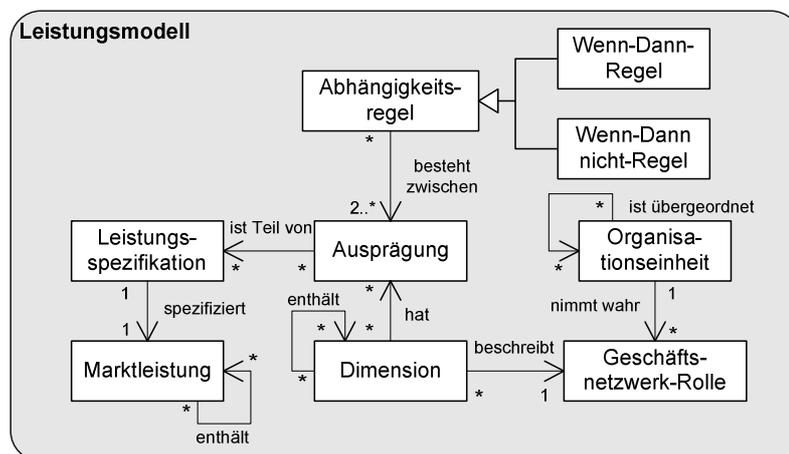


Abbildung 36: Metamodell des Leistungsmodells

Das Leistungsmodell soll nicht nur die fokussierten Märkte und das Wettbewerbsverhalten repräsentieren (Aussensicht), sondern auch die internen Abläufe, Strukturen und Potenziale der Organisation abbilden (Innensicht). Die beiden Sichten auf das Leistungsmodell können wie folgt strukturiert werden:³²⁷

- Aussensicht („Leistungsangebotsmodell“): Die Aussensicht bildet ab, auf welchem Markt die Bezugseinheit tätig sein will, d.h. mit welchen Produkttypen/Prozessklassen das Unternehmen den Markt „bedienen“ will und welche Endkunden in welchen Regionen letztendlich angesprochen werden sollen. Dabei sind die Produkttypen/Prozessklassen nicht alleine nach ihrer Art zu unterscheiden (Kredit, Anlage usw.), sondern auch dahingehend, ob sie einzeln, in einem Bündel oder ausgerichtet an den Kundenprozessen angeboten werden. Abhängig davon sind auch die Festlegung der Marke sowie die Auswahl des Vertriebswegs und des Vertriebskonzepts. In der Aussensicht wird somit im Wesentlichen der Marktauftritt der Bezugseinheit bestimmt. Typische Dimensionen für die Spezifikation der jeweiligen Merkmale sind Land/Region, Kunde, Kernprodukte und Leistungen, Marke und Vertrieb.
- Innensicht („Leistungserstellungsmodell“): Die Innensicht bildet die Kompetenzen und die Form der Faktorkombination einer Bezugseinheit ab. Orientierung für eine Spezifikation bieten dabei die Ziele und Erfolgsfaktoren. Darüber hinaus sind Kooperationen und

³²⁷ Vgl. im Folgenden *Heinrich/Winter* (2004).

die Organisationsstruktur (eher zentrale oder eher dezentrale Ausrichtung) zu berücksichtigen. Typische Dimensionen für die Spezifikation der jeweiligen Merkmale sind beispielsweise Kompetenzfelder, Ziele/ Erfolgsfaktoren, Kooperationen und Organisationsstruktur.

4.1.2.4 Metamodell des Zielsystems

Das Zielsystem eines Unternehmens bzw. einer Geschäftseinheit bildet die Grundlage für die nachfolgende Spezifikation der zur Prozessgestaltung benötigten Führungsgrößen und Erfolgsfaktoren. Es kann beispielsweise anhand der Balanced Scorecard (BSC)³²⁸ beschrieben werden. Diese hat sich in den letzten Jahren als Managementmethode für die strategische Unternehmensführung weitgehend etabliert³²⁹ und wird deshalb auch in der vorliegenden Arbeit verwendet. Die BSC ist ein Kennzahlen- und Managementsystem, bei dem eine rein finanzielle Betrachtung des Unternehmens um die Berücksichtigung immaterieller Werte erweitert wird.³³⁰ Die Strategie wird dabei durch strategische Ziele und Erfolgsfaktoren abgebildet und mittels monetärer und nicht monetärer Kennzahlen messbar gemacht. Die BSC systematisiert die zentralen strategischen Erfolgsfaktoren eines Unternehmens in die vier Perspektiven Finanzen, Kunden, Prozesse und Potenziale. Diese werden durch die Bildung perspektivenorientierter Kennzahlen operationalisiert. Abbildung 37 zeigt das Metamodell des Zielsystems (BSC).

Ein **kritischer Erfolgsfaktor** (KEF) bezeichnet eines der wenigen Merkmale, die den Erfolg eines Unternehmens massgeblich beeinflussen. Die kritischen Erfolgsfaktoren konkretisieren die **Unternehmensziele** und helfen damit der Unternehmensführung, sich auf die wesentlichen Aspekte zur Erreichung der Unternehmensziele zu konzentrieren.³³¹ Eine **Massnahme** bezeichnet eine Tätigkeit zur Realisierung eines bestimmten Zieles. Dabei kann es sich beispielsweise um ein Vorhaben, strategisches Projekt oder Programm handeln. Die Durchführung einer Massnahme liegt in der Verantwortung einer bestimmten Organisationseinheit. Jeder KEF wird durch eine oder mehrere Kennzahlen operationalisiert. Eine **Kennzahl** kann als Zahl betrachtet werden, die einen quantitativ erfassbaren Sachverhalt in konzentrierter Form als Information erfasst.³³² Das Ziel von Kennzahlen ist es, komplexe Sachverhalte in Unternehmen in relativ einfacher Form darzustellen. Dabei steht insbesondere der Aspekt der optimalen Aggregation im Vordergrund, da zum einen die Anzahl der Kennzahlen möglichst gering und sie selbst möglichst einfach gehalten werden sollten. Andererseits sollten sie aber auch den realen Sachverhalt möglichst originalgetreu und unverfälscht wiedergeben.³³³ Diese

³²⁸ Vgl. Kaplan/Norton (1997).

³²⁹ Vgl. hierzu z.B. Horváth&Partners (2005).

³³⁰ Vgl. Böh/Meyer (2004), S. 106.

³³¹ In Anlehnung an die Definition eines KEF des Prozessmanagements. Vgl. Brecht (2002), S. 159.

³³² Vgl. Reichmann (1997), S. 19.

³³³ Vgl. Harengel (2000), S. 34.

Originaltreue kann nur begrenzt erreicht werden, da die Ermittlung der Kennzahlen von der Genauigkeit der Messmethode und von stochastischen Schwankungen abhängig ist.³³⁴ Ausserdem können einzelne Kennzahlen unterschiedlich interpretiert werden, da weitere Informationen für eine genaue Beurteilung des Sachverhalts fehlen.³³⁵

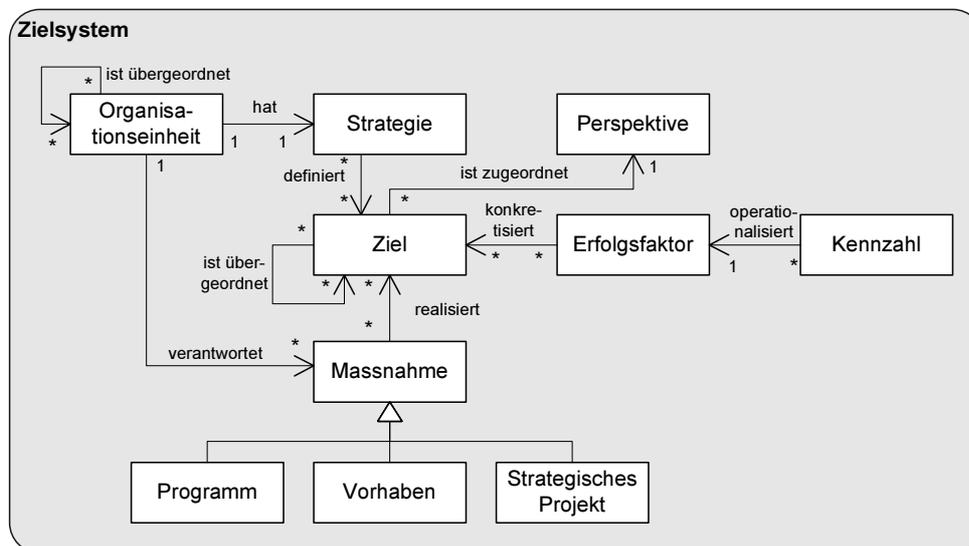


Abbildung 37: Metamodell des Zielsystems (BSC)

Diese Nachteile einzelner Kennzahlen können teilweise durch Kennzahlensysteme behoben werden. Unter einem **Kennzahlensystem** versteht man eine geordnete Gesamtheit von Kennzahlen, die in sachlich sinnvoller Beziehung zueinander stehen und somit einen bestimmten Sachverhalt vollständig beschreiben sollen.³³⁶

Nach KAPLAN/NORTON ist die Balanced Scorecard (BSC) ein Kennzahlensystem, bestehend aus finanziellen und nichtfinanziellen Grössen, das die verschiedenen Dimensionen (Kunden, Finanzen, interne Prozesse, Potenzial) eines Unternehmens gleichberechtigt nebeneinander darstellen soll.³³⁷ Die BSC stellt eine Weiterentwicklung konventioneller Kennzahlensysteme dar (z.B. des DuPont- oder ZVEI-Kennzahlensystems). Durch die BSC sollen alle Mitarbeiter und Manager motiviert und in die Lage versetzt werden, die Unternehmensstrategie erfolgreich umzusetzen.³³⁸ Die BSC wird deshalb oftmals mit einem Flugzeug-Cockpit verglichen, in dem der Pilot (Manager) einen Überblick über die relevanten Informationen zur Steuerung des Flugzeugs (Unternehmens) angezeigt bekommt.³³⁹

³³⁴ Vgl. Harengel (2000), S. 35.

³³⁵ Vgl. Weber et al. (1997), S. 442.

³³⁶ Vgl. Weber et al. (1997), S. 35f.

³³⁷ Vgl. Kaplan/Norton (1997).

³³⁸ Vgl. Kaplan/Norton (1997), S. 7f.

³³⁹ Vgl. Kaplan/Norton (1997), S. 1.

KAPLAN/NORTON schlagen vier wesentliche Dimensionen für die Systematisierung der zentralen Erfolgsfaktoren und Kennzahlen eines Unternehmens bzw. einer Geschäftseinheit vor:³⁴⁰

- Die **finanzielle Perspektive** enthält monetäre Kennzahlen, um den wirtschaftlichen Erfolg des Unternehmens bzw. der Geschäftseinheit abzubilden. Im Mittelpunkt stehen finanzwirtschaftliche Messgrößen, basierend auf vergangenheitsorientierten Daten, wie z.B. Rentabilität und Liquidität. Indirekt geben diese Daten Aufschluss darüber, ob die Unternehmensstrategie und die daraus abgeleiteten Massnahmen erfolgreich waren oder nicht.
- Die **Markt-/Kundenperspektive** identifiziert Kunden- und Marktsegmente, die für das Unternehmen bzw. die Geschäftseinheit wichtig sind. Entsprechend dieser Analyse können darauf aufbauend Kennzahlen für die Kundenzufriedenheit, die Anzahl der Kundenakquisitionen, die Kundenrentabilität sowie für Gewinn- und Marktanteile festgelegt werden. Dadurch soll sichergestellt werden, dass das Unternehmen die Möglichkeit hat, sich besser am Markt zu orientieren.
- Die **Geschäftsprozessperspektive** konzentriert sich auf Leistungsprozesse zur Erreichung der in der Finanz- und Markt-/Kundenperspektive definierten Ziele. Zielsetzung in diesem Bereich muss es sein, kritische Prozesse anhand zweier Einflussfaktoren zu identifizieren: Kundenbedürfnisse und Erwartungen der Anteilseigner. Ausgehend von dieser Analyse müssen auf diese Prozesse Verbesserungsschwerpunkte gelegt werden. Dies beinhaltet nicht nur die existierenden Prozesse, vielmehr fördert die BSC auch die Entwicklung neuer Prozesse. Die BSC erweitert ausserdem den Begriff der Wertkette in der Hinsicht, dass auch der Innovationsprozess völlig neuer Produkte und Dienstleistungen integriert wird. Somit soll sichergestellt werden, dass auch ein langfristiger Aspekt der Wertschöpfung berücksichtigt wird.
- Die **Lern- und Entwicklungsperspektive** identifiziert und schafft eine Infrastruktur im Unternehmen, die ein kontinuierliches Wachstum und Lernen ermöglicht. Diese Entwicklung betrifft drei Bereiche im Unternehmen: Menschen, Systeme und Unternehmenskultur. Es muss darauf geachtet werden, dass Wachstum und Lernen dieser drei Elemente aufeinander abgestimmt sind.

Mit der Balanced Scorecard wird versucht, jede Kennzahl auf der Basis einer Ursache-Wirkungskette mit monetären und nicht-monetären Zielgrößen zu verbinden.

4.1.2.5 Definition der Metaentitätstypen auf Strategieebene

In der folgenden Tabelle 8 sind die im vorhergehenden Abschnitt erwähnten Metaentitätstypen sowie deren Beziehungen genauer definiert.

³⁴⁰ Vgl. *Kaplan/Norton* (1997), S. 24f. Die von Kaplan und Norton vorgeschlagenen Dimensionen der BSC sind nur Anhaltspunkte. In den letzten Jahren haben sich in einigen Unternehmen weitere Dimensionen etabliert, wie beispielsweise die ökologische Perspektive.

Abhängigkeitsregel		
Definition	Eine Abhängigkeitsregel definiert eine direkte Abhängigkeit zwischen zwei oder mehreren Dimensionen des Leistungsmodells. Sie kann beispielsweise als Wenn-Dann-Regel formuliert sein.	
Beziehungen zu	Dimension	Eine Abhängigkeitsregel besteht zwischen zwei oder mehreren Dimensionen.
Business Collaboration Infrastructure (BCI)		
Definition	Die BCI ist definiert als multilaterale Kooperationsinfrastruktur, die die Herstellung einer technischen n:m-Konnektivität ermöglicht und von Unternehmen bzw. Geschäftseinheiten für den übergreifenden Informations- oder Leistungsaustausch innerhalb des Geschäftsnetzwerkes eingesetzt wird. ³⁴¹	
Beziehungen zu	Rolle	Die BCI wird von zwei oder mehreren Rollen im Geschäftsnetzwerk verwendet.
Dimension		
Definition	Eine Dimension beschreibt einen Teilaspekt der strategischen Ausrichtung eines Unternehmens oder einer Geschäftseinheit. Typische Dimensionen für die Spezifikation der strategischen Ausrichtung sind z.B. Region, Kernprodukte und -leistungen, Marke, Vertrieb, Kompetenzfelder und Erfolgsfaktoren. ³⁴²	
Beziehungen zu	Ausprägung	Eine Dimension hat eine oder mehrere Ausprägung/en.
	Rolle	Eine Dimension beschreibt eine oder mehrere Rolle/n.
	Abhängigkeitsregel	Zwischen zwei oder mehreren Dimensionen kann/können keine, eine oder mehrere Abhängigkeitsregel/n definiert sein.
Dimensionsausprägung		
Definition	Eine Dimensionsausprägung beschreibt ein konkretes, für eine Dimension festgelegtes Merkmal. ³⁴³	
Beziehungen zu	Dimension	Eine Dimensionsausprägung ist genau einer Dimension zugeordnet.
	Leistungsspezifikation	Eine Dimensionsausprägung ist Teil einer oder mehrerer Leistungsspezifikation/en.
Exclusive Service Provider (ESP)		
Definition	Ein ESP erzeugt Leistungskomponenten für andere Service Provider oder Service Integrators. Im Gegensatz zu Shared Service Providers werden aber nicht grosse Mengen von Leistungskomponenten für mehrere Abnehmer produziert, sondern spezifischere Leistungskomponenten für einen oder wenige Abnehmer und damit in eher kleineren Mengen. An die Stelle von Kunden- bzw. Kundenprozesskenntnis (SI) bzw. Kostenführerschaft (SSP) treten als Haupt-Erfolgsfaktor Schnelligkeit, Flexibilität sowie spezifische Prozesskompetenzen. ³⁴⁴	
Beziehungen zu	Marktleistung	Ein ESP bezieht Marktleistungen von anderen Service Providers und erbringt individualisierte Marktleistungen für Service Integrators oder Service Providers.

³⁴¹ Vgl. Winter (2003c), S. 54.

³⁴² Vgl. Heinrich/Winter (2004), S. 7.

³⁴³ Vgl. Heinrich/Winter (2004), S. 7.

³⁴⁴ Vgl. Winter (2003c), S. 53.

Kennzahl		
Definition	Eine Kennzahl dient der Objektivierung und Visualisierung von Unternehmenszielen und Verbesserungsbemühungen. Sie aggregiert mehrere in einem Unternehmen vorhandene Zahlen zu einem Wert. ³⁴⁵	
Beziehungen zu	Erfolgsfaktor	Eine Kennzahl operationalisiert genau einen Erfolgsfaktor.
	Perspektive	Eine Kennzahl ist genau einer Perspektive zugeordnet.
Kontaktpräferenz		
Definition	Die Kontaktpräferenz beschreibt die Art und Weise, in der der Kunde oder eine bestimmte Kundengruppe bevorzugt mit dem Unternehmen in Kontakt tritt.	
Beziehungen zu	Kunde	Eine Kontaktpräferenz ist einem oder mehreren Kunden zugeordnet.
Kritischer Erfolgsfaktor (KEF)		
Definition	Ein kritischer Erfolgsfaktor (KEF) bezeichnet eines der wenigen Merkmale, die den Erfolg eines Unternehmens ausmachen. ³⁴⁶	
Beziehungen zu	Kennzahl	Ein KEF wird durch keine, eine oder mehrere Kennzahl/en operationalisiert.
	Ziel	Ein KEF ist für die Erreichung eines oder mehrerer Unternehmensziels/-ziele erfolgsentscheidend.
	Perspektive	Ein KEF ist genau einer Perspektive zugeordnet.
Kunde		
Definition	Ein Kunde bezeichnet eine Person oder Organisation, die Produkte oder Leistungen eines Unternehmens bzw. einer Geschäftseinheit bezieht.	
Beziehungen zu	Kundensegment	Ein Kunde gehört einem oder mehreren Kundensegment/en an.
	Kundenprozess	Ein Kunde hat keinen, einen oder mehrere Kundenprozess/e.
	Kontaktpräferenz	Ein Kunde hat eine oder mehrere Kontaktpräferenz/en.
Kundenaktivität		
Definition	Eine Kundenaktivität stellt einen Teilschritt innerhalb des gesamten Kundenprozesses dar. ³⁴⁷	
Beziehungen zu	Serviceaktivität	Eine Kundenaktivität bedarf einer oder mehrerer Serviceaktivität/en.
	Kundenprozess	Eine Kundenaktivität ist Teil eines Kundenprozesses.
Kundenprozess		
Definition	Als Kundenprozess wird die Gesamtheit der Teilschritte und deren Reihenfolge bezeichnet, die einem Kunden zugeordnet werden können.	
Beziehungen zu	Kunde	Ein Kundenprozess ist einem oder mehreren Kunden zugeordnet.
	Marktleistung	Ein Kundenprozess bedarf einer oder mehrerer Leistung/en.
	Kundenaktivität	Ein Kundenprozess besteht aus einer oder mehreren Kundenaktivität/en

³⁴⁵ Vgl. Reichmann (1997), S. 19.

³⁴⁶ In Anlehnung an die Definition eines KEF des Prozessmanagements. Vgl. Brecht (2002), S. 159.

³⁴⁷ Vgl. Choinowski et al. (2003), S. 24.

Kundensegment		
Definition	Wird eine heterogene Gesamtmenge von Kunden in disjunkte, homogene Mengen aufgeteilt, so spricht man von Kundensegmenten. Die Untergliederung kann dabei unter verschiedenen Gesichtspunkten erfolgen.	
Beziehungen zu	Kunde	Ein Kundensegment umfasst einen oder mehrere Kunden.
Leistungsspezifikation		
Definition	Eine Leistungsspezifikation beschreibt eine oder mehrere Marktleistungen, die von einem Unternehmen bzw. einer Geschäftseinheit angeboten werden.	
Beziehungen zu	Dimensionsausprägung	Eine Leistungsspezifikation wird durch eine oder mehrere Dimensionsausprägung/en definiert.
	Marktleistung	Eine Leistungsspezifikation spezifiziert eine oder mehrere Marktleistung/en.
	Prozessleistung	Eine Leistungsspezifikation beeinflusst die Definition einer oder mehrerer Prozessleistung/en auf Organisationsebene.
Marktleistung		
Definition	Eine Marktleistung ist ein Produkt oder eine Dienstleistung, die ein Unternehmen oder eine Geschäftseinheit am Markt bzw. im Geschäftsnetzwerk anbietet. Sie wird durch das Zusammenwirken eines Bündels von Leistungen erbracht. ³⁴⁸ Eine Marktleistung kann standardisiert oder individualisiert sein.	
Beziehungen zu	Prozessleistung	Eine Marktleistung umfasst eine oder mehrere Prozessleistung/en auf Organisationsebene.
	Kundenprozess	Eine Marktleistung wird für die Unterstützung keines, eines oder mehrerer Kundenprozesse/s verwendet.
	Rolle	Eine Marktleistung wird von einem oder mehreren Unternehmen bzw. einer oder mehreren Geschäftseinheit/en im Geschäftsnetzwerk verwendet bzw. erbracht.
	Serviceprozess	Eine Marktleistung ist Teil keines, eines oder mehrerer Serviceprozesse/s.
	Leistungsspezifikation	Eine Marktleistung wird durch eine oder mehrere Leistungsspezifikation/en spezifiziert.
Massnahme		
Definition	Eine Massnahme bezeichnet eine Tätigkeit zur Realisierung eines bestimmten Zieles.	
Beziehungen zu	Ziel	Eine Massnahme realisiert ein oder mehrere Ziel/e.
	OE	Eine Massnahme wird von einer oder mehreren Organisationseinheit/en durchgeführt.
Organisationseinheit (OE)		
Definition	Eine Organisationseinheit ist ein eigenständiger, permanenter Teil der organisatorischen Struktur eines Unternehmens (z.B. Abteilung, Unternehmensbereich, Stelle). Sie kann mehr oder weniger aggregiert sein. ³⁴⁹ Neben weiteren Organisationseinheiten und Mitarbeitern umfasst sie auch Geschäftsobjekte. Eine Organisationseinheit kann auf Strategieebene eine Geschäftseinheit oder das Unternehmen selbst darstellen.	
Beziehungen zu	Rolle	Eine OE nimmt eine oder mehrere Rolle/n im Geschäftsnetzwerk ein.

³⁴⁸ Vgl. Glossar in *IMG* (1997).

³⁴⁹ Vgl. Glossar in *IMG* (1997).

	Massnahme	Eine OE ist für die Durchführung keiner, einer oder mehrerer Massnahm/en verantwortlich.
	Strategie	Eine OE verfolgt eine bestimmte Strategie.
	Serviceaktivität	Eine OE ist für die Durchführung keiner, einer, oder mehrerer Serviceaktivität/en verantwortlich.
	Geschäftsprozess	Eine OE ist für einen oder mehrere Geschäftsprozess/e auf Organisationsebene verantwortlich.
Perspektive		
Definition	Der Erfolg eines Unternehmens bei der Strategieumsetzung wird anhand von vier möglichst gleich gewichteten Perspektiven beurteilt. Die zentralen strategischen Erfolgsfaktoren eines Unternehmens werden in diese Perspektiven systematisiert und anschliessend durch die Bildung perspektivenorientierter Kennzahlen operationalisiert. ³⁵⁰	
Beziehungen zu	KEF	Eine Perspektive umfasst einen oder mehrere Erfolgsfaktor/en.
	Kennzahl	Eine Perspektive umfasst eine oder mehrere Kennzahl/en.
	Ziel	Eine Perspektive umfasst ein oder mehrere Ziel/e.
Public Service (PS)		
Definition	Ein PS stellt Leistungskomponenten bereit, die keinen spezifischen fachlichen Charakter haben, aber zur Abwicklung der Leistungserstellung unverzichtbar sind. Im engeren Sinne handelt es sich dabei um Dienste mit quasi-hoheitlicher Funktion, wie z.B. Zertifizierung oder Authentifizierung. Im weiteren Sinne können auch andere elektronische Dienste, wie z.B. Informationsdienste, Applikationsbetrieb, Katalogdienste, Zahlungsabwicklung und Logistikabwicklung, darunter subsumiert werden. ³⁵¹	
Beziehungen zu	Marktleistung	Ein Public Service erbringt eine oder mehrere standardisierte Marktleistung/en.
Rolle (Geschäftsnetzwerk)		
Definition	Eine Rolle definiert einen kompetenzbasierten Typ von Organisationseinheit (Unternehmen oder Geschäftseinheit) im Geschäftsnetzwerk. ³⁵² Es handelt sich dabei um eine abstrakte Klasse, von der keine Objekte erzeugt werden können.	
Beziehungen zu	BCI	Eine Rolle kann die BCI nutzen.
	Dimension	Eine Rolle wird durch eine oder mehrere Dimension/en beschrieben.
	Marktleistung	Eine Rolle verwendet oder erzeugt eine oder mehrere Marktleistung/en.
	OE	Eine Rolle wird durch ein oder mehrere Unternehmen bzw. eine oder mehrere Geschäftseinheiten wahrgenommen.
Serviceaktivität		
Definition	Eine Serviceaktivität stellt einen Teilschritt innerhalb des gesamten Serviceprozesses dar. Für jede Serviceaktivität kann festgelegt werden, ob diese selbst oder durch einen Kooperationspartner im Geschäftsnetzwerk durchführt werden soll. ³⁵³	

³⁵⁰ Vgl. Kaplan/Norton (1997).

³⁵¹ Vgl. Winter (2003c), S. 53.

³⁵² Vgl. Winter (2003c), S. 53.

³⁵³ Vgl. Heinrich (2002b), S. 87.

Beziehungen zu	Kundenaktivität	Eine Serviceaktivität unterstützt eine oder mehrere Kundenaktivität/en.
	Serviceprozess	Eine Serviceaktivität ist Teil eines Serviceprozesses oder mehrerer Serviceprozesse.
	Geschäftsprozess	Eine Serviceaktivität wird durch einen oder mehrere Geschäftsprozess/e (bzw. dessen Aktivitäten) auf Organisationsebene unterstützt.
	OE	Eine Serviceaktivität ist genau einer (internen oder externen) Organisationseinheit (Unternehmen bzw. Geschäftseinheit) zugeordnet.
Service Integrator (SI)		
Definition	Ein SI integriert Leistungskomponenten, um damit einen bestimmten Kundenprozess ganzheitlich zu unterstützen. Kernkompetenz des Service Integrator ist deshalb die umfassende Kenntnis von Kundenbedürfnissen bzw. Kundenprozessen. ³⁵⁴	
Beziehungen zu	Marktleistung	Ein SI bezieht Marktleistungen von Service Providern und erbringt Marktleistungen für einen bestimmten Kundenprozess.
Serviceprozess		
Definition	Ein Serviceprozess beschreibt die einzelnen Serviceaktivitäten sowie deren Reihenfolge, die ein Unternehmen oder eine Geschäftseinheit durchführen muss, um einen bestimmten Kundenprozess und dessen Kundenaktivitäten zu unterstützen. ³⁵⁵	
Beziehungen zu	Serviceaktivität	Ein Serviceprozess besteht aus einer oder mehreren Serviceaktivität/en.
	Marktleistung	Ein Serviceprozess umfasst eine oder mehrere Marktleistung/en
Shared Service Provider (SSP)		
Definition	Ein SSP produziert Leistungskomponenten für viele andere Service Providers oder Service Integrators in grossen Mengen. Haupterfolgskriterien des SSP sind folglich die Kostenführerschaft und effiziente Leistungserstellung. ³⁵⁶	
Beziehungen zu	Marktleistung	Ein SSP bezieht Leistungen von anderen Service Providern und erbringt standardisierte Leistungen für Service Integrators oder Service Providers.
Strategie		
Definition	Eine Strategie beschreibt die geplanten Verhaltensweisen eines Unternehmens oder einer Geschäftseinheit zur Erreichung ihrer Ziele. Sie legt somit fest, auf welche Art und Weise ein mittelfristiges oder langfristiges Ziel erreicht werden soll. ³⁵⁷	
Beziehungen zu	Ziel	Eine Strategie definiert ein oder mehrere Ziel/e.
	OE	Eine Strategie ist genau einer Organisationseinheit zugeordnet.

³⁵⁴ Vgl. Winter (2003c), S. 53.

³⁵⁵ Vgl. Heinrich (2002b), S. 85.

³⁵⁶ Vgl. Winter (2003c), S. 53.

³⁵⁷ Vgl. hierzu z.B. Mintzberg (1987).

Vertriebsweg		
Definition	Der Vertriebsweg kennzeichnet den Weg, auf dem Produkte oder Dienstleistungen zum Kunden gelangen. Typische Beispiele für Vertriebswege sind Filiale, Internet und Versandhandel.	
Beziehungen zu	Serviceprozess	Ein Vertriebsweg kann einen oder mehrere Serviceprozess/e beeinflussen.
Ziel		
Definition	Das Ziel definiert ein bestimmtes Ergebnis, das in der Zukunft erreicht werden soll.	
Beziehungen	KEF	Ein Ziel wird durch einen oder mehrere Erfolgsfaktor/en konkretisiert.
	Perspektive	Ein Ziel ist genau einer Perspektive zugeordnet.
	Strategie	Ein Ziel wird von einer Strategie definiert.

Tabelle 8: Metaentitätstypen der Strategieebene

4.1.2.6 Festlegung der verwendeten Symbole auf Strategieebene

Die folgende Tabelle 9 zeigt die für die zuvor definierten Metaentitäts- und Beziehungstypen verwendeten Notationselemente (Symbole).

Symbole der Strategieebene		
BCI	Dimension	Dimensionsausprägung
Kennzahl	Exclusive Service Provider	Kritischer Erfolgsfaktor
Kunde	Kundenaktivität	Kundenprozess
Kundensegment	Massnahme	Organisationseinheit
Perspektive	Public Service	Serviceaktivität

Service Integrator	Serviceprozess	Shared Service Provider
Strategie	Ziel	
bezieht Leistung (direkt) (von Rolle zu Rolle)	bezieht Leistung (über BCI) (von Rolle zu Rolle)	gehört zu (von Kunde zu Kundensegment)
bedarf (von Kundenakt. zu Serviceakt.)	beeinflusst (von Erfolgsfaktor zu Strategie)	hat (von Ziel zu Erfolgsfaktor)

Tabelle 9: Symbole der Strategieebene

Die Notationselemente sind getrennt von den Metaentitäts- und Beziehungstypen definiert und stellen lediglich einen Vorschlag dar. Dadurch ist eine Anpassung an ein spezifisches Umfeld und an individuelle Präferenzen möglich.

Für die Metaentitätstypen „Marktleistung“, „Leistungsspezifikation“, „Kontaktpräferenz“ und „Vertriebsweg“ wurden keine eigenen Symbole definiert, da sie durch die generischen Metaentitätstypen „Dimension“ und „Dimensionsausprägung“ repräsentiert werden können. Der Metaentitätstyp „Rolle (Geschäftsnetzwerk)“ stellt eine abstrakte Klasse im Sinne der UML-Spezifikation dar. Von einer abstrakten Klasse können nicht direkt Objekte erzeugt werden. Sie besitzt somit kein Notationselement. Durch Vererbung können allerdings von einer abstrakten Klasse Unterklassen abgeleitet werden. Objekte werden dann von den Unterklassen („Service Integrator“, „Shared Service Provider“ etc.) erzeugt.

Bei Beziehungstypen, denen kein Notationselement zugeordnet ist, handelt es sich entweder um eine modellübergreifende Beziehung oder um eine Aggregationsbeziehung, die in Form einer Schachtelung von Symbolen dargestellt wird (z.B. die Beziehung zwischen „Serviceprozess“ und „Serviceaktivität“).

4.2 Modellierung auf Organisationsebene

Die Gestaltung der Organisationsebene stellt das Bindeglied zwischen Strategieentwicklung und Systementwicklung dar (vgl. Abschnitt 2.3.2). Wesentliche Anforderungen an einen Ansatz für die Modellierung der Organisationsebene sind:³⁵⁸

³⁵⁸ Vgl. Winter (2004a), Kühn/Karagiannis (2005), S. 1495 und Abschnitt 2.3.2.

- Abbildung einer Übersicht der Leistungs-, Unterstützungs- und Führungsprozesse eines Unternehmens bzw. einer Geschäftseinheit,
- transparente Definition der Geschäftsprozesse anhand des Kontroll- und Informationsflusses,
- Beschreibung der in einem Geschäftsprozess bearbeiteten Objekte, eingesetzten Ressourcen sowie beteiligten Rollen bzw. Organisationseinheiten und
- Sicherstellung der Führbarkeit.

Für die Identifikation von Schlüsselementen auf der Organisationsebene sowie deren Zusammenhänge bieten die in dieser Arbeit bewerteten Methoden zur Unternehmensmodellierung (vgl. Abschnitt 3.2) bereits einige wesentliche Anhaltspunkte und Konzepte. Zudem wird auf domänenspezifische Ansätze/Methoden des Prozessmanagements und zur Gestaltung der Aufbauorganisation zurückgegriffen (vgl. Tabelle 7).³⁵⁹

Auf der Organisationsebene können folgende Ergebnisdokumententypen unterschieden werden:

- Die **Prozesslandkarte** gibt einen Überblick über die relevanten Prozesse sowie deren Leistungs- und Informationsbeziehungen zu anderen Prozessen innerhalb und ausserhalb des Unternehmens bzw. der Geschäftseinheit.³⁶⁰
- Die **Prozessvision** beschreibt die grundlegenden Prinzipien zur langfristigen Gestaltung eines bestimmten Prozesses.³⁶¹
- Die **Prozessführung** bestimmt die Steuerung des Prozesses. Sie plant Soll-Werte, erhebt Ist-Werte, legt Erfolgsfaktoren und Führungsgrössen fest und löst notwendige Massnahmen aus.³⁶² Ziel der Prozessführung ist die Erhöhung der Effizienz und Effektivität des Prozesses durch permanente Weiterentwicklung.³⁶³
- Die **Ablauforganisation** bildet die konkreten Aktivitäten eines Prozesses sowie ihre Abfolge und ihre Zuordnung zu den Rollenträgern ab.³⁶⁴
- Die **Aufbauorganisation** beschreibt die organisatorischen Rollenträger³⁶⁵ eines Unternehmens mit den zwischen ihnen bestehenden Kommunikations- und Weisungsbeziehungen.³⁶⁶

³⁵⁹ Zur Identifikation von wesentlichen Metaentitätstypen der Organisationsebene wird primär auf folgende Quellen zurückgegriffen: *Österle* (1995), *Scheer* (2001), *Brecht* (2002), *IMG* (1997), *Rosemann/zur Mühlen* (1997).

³⁶⁰ Vgl. Ergebnisse in *IMG* (1997).

³⁶¹ Vgl. Techniken in *IMG* (1997).

³⁶² Vgl. *Österle* (1995), S. 54.

³⁶³ Vgl. Ergebnisse *IMG* (1997).

³⁶⁴ Vgl. hierzu die Modellierung der Steuerungssicht in *Scheer* (2001).

- Die **Informationslandkarte** identifiziert die in einem Unternehmen bzw. einer Geschäftseinheit zur Verfügung stehenden Informationen und stellt diese aus verschiedenen Blickwinkeln dar.³⁶⁷

4.2.1 Vorgehensmodell

Das Vorgehensmodell zur Gestaltung der Organisationsebene basiert in Teilen auf der Methode PROMET-BPR.³⁶⁸ Diese dient dazu, Prozesse in Unternehmen zu identifizieren, zu verbessern und neu zu gestalten. Sie besteht auf Level 1 aus sechs wesentlichen Aktivitäten (vgl. Abbildung 38).

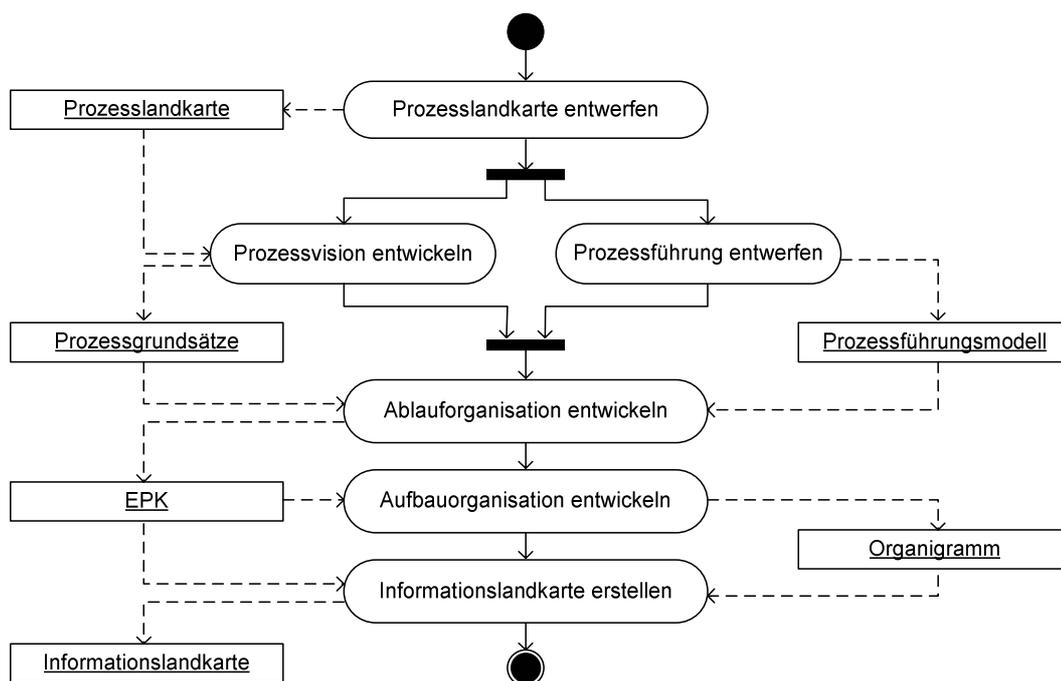


Abbildung 38: Vorgehensmodell auf Organisationsebene (Level 1)

Im ersten Schritt wird eine *Prozesslandkarte entworfen*. Die Zielsetzung einer Prozesslandkarte ist es, Notwendigkeiten, bestehende Abhängigkeiten und Schnittstellen näher darzulegen. Dabei sind unter dem Begriff Notwendigkeiten die Elemente einer Organisation zu verstehen, welche unverzichtbar für ein Bestehen am Markt sind (z. B. F&E, Personalmanagement). Die Abhängigkeiten und Schnittstellen sollen dabei den Zusammenhang und das Zusammenspiel der Prozesse verdeutlichen.³⁶⁹ Die Prozesslandkarte gibt somit einen Überblick über die wettbewerbsentscheidenden Prozesse des eigenen Unternehmens, die wichtigsten

³⁶⁵ Im folgenden bezeichnet der Oberbegriff organisatorischer Rollenträger in Anlehnung an die Definition eines organisatorischen Konstrukts nach ROSEMANN/ZUR MÜHLEN die Generalisierung von Rolle, Stelle, Stellentyp, Person und Organisationseinheit (vgl. Rosemann/zur Mühlen (1997)).

³⁶⁶ Vgl. Scheer (2001), S. 52.

³⁶⁷ Vgl. Strauch (2002), S. 180.

³⁶⁸ Vgl. IMG (1997) und <http://www.promet-web.com>.

³⁶⁹ Vgl. Zellner (2003), S. 275.

Prozesse der Geschäftspartner sowie den Leistungs- und Informationsaustausch zwischen den Prozessen. Die Identifikation der Leistungsprozesse ist überwiegend an die Leistungen, die das Unternehmen nach aussen abgibt, und an die ermittelten Produkt-/Marktkombinationen des Unternehmens angelehnt. Neben den Leistungsprozessen müssen auch die zum kontinuierlichen Betrieb der Prozesskoordination benötigten Unterstützungs- und Führungsprozesse definiert werden.³⁷⁰ Wenn alle Prozesstypen definiert sind und deren Leistungsaustausch bestimmt ist, kann als Ergebnisdokument die Prozesslandkarte angefertigt werden.

Im zweiten Schritt „Prozessvision entwickeln“ wird die *Prozessvision entwickelt* und die *Prozessführung entworfen*. Dabei soll die Suche nach neuen Lösungen für die Gestaltung von Prozessabläufen im Unternehmen unterstützt werden.³⁷¹ Es werden radikale Innovationen im Rahmen eines Abgleichs zwischen Geschäftsstrategie und Prozess angestrebt, d.h. aus der Geschäftsstrategie werden Vorgaben für die Prozesse abgeleitet, so dass ein realistischer und langfristiger Rahmen für die Gestaltung von Prozessen geschaffen werden kann.³⁷² Als Ergebnisdokument und Vorgabe für die Prozessgestaltung fungieren die Prozessgrundsätze. Umgekehrt können aber auch die im Rahmen der Prozessvision identifizierten, neuen Ideen bzw. Lösungen zu einer Anpassung von Teilen der Geschäftsstrategie führen.

Parallel zur Prozessvision kann die *Prozessführung entwickelt* werden. Die Prozessführung bestimmt die Steuerung eines Prozesses. Sie plant Soll-Werte, erhebt Ist-Werte, legt Führungsgrößen fest und löst notwendige Massnahmen aus. Input erhält die Prozessführung aus der Geschäftsstrategie in Form von Kennzahlen der BSC und aus Anregungen aus der Prozessvision.³⁷³ Bei der Entwicklung der Prozessführung werden zunächst mögliche Erfolgsfaktoren des Unternehmens bzw. der Geschäftseinheit (aus der BSC) und generelle Prozesserefolgsfaktoren (z.B. Zeit, Qualität, Kosten, Flexibilität usw.) identifiziert und im Anschluss die für den Unternehmenserfolg kritischen Faktoren benannt. Anschliessend erfolgt die Operationalisierung der Erfolgsfaktoren in Form von Führungsgrößen. Diese sind Indikatoren für die Effektivität und Effizienz eines Prozesses und dienen der Planung und Beurteilung der Prozessqualität.³⁷⁴ Auf Basis der Führungsgrößen werden Prozessziele (Zielwerte) festgelegt. Diese repräsentieren den erwünschten Zustand (Sollwerte) des Prozesses. Die Aktivitäten und Ergebnisse der kontinuierlichen Prozessführung sind mittels Berichtswesen (z.B. Prozessstatusbericht) festzuhalten. Zielabweichungen und Schwachstellen sollen so dokumentiert und eine graphische Übersicht über die Entwicklung der wichtigsten Führungsgrößen dargeboten werden. Zuletzt sind noch die Aufgaben und Verantwortlichkeiten festzulegen.

³⁷⁰ Vgl. Techniken in *IMG* (1997), S. 36-38.

³⁷¹ Vgl. Techniken in *IMG* (1997), S. 45-56.

³⁷² Vgl. *Hess* (1996), S. 179.

³⁷³ Vgl. zur Prozessführung Techniken in *IMG* (1997), S. 89ff.

³⁷⁴ Zur Messung des Erfolgsfaktors „Geschwindigkeit“ sind z.B. die Durchlaufzeit, Liegezeit und Kontrollzeit typische Führungsgrößen.

Nach dem Entwurf der Prozessführung kann die *Ablaufplanung entwickelt* werden. Die Aufgabe der Ablaufplanung besteht darin, sowohl die konkreten Funktionen bzw. Aktivitäten³⁷⁵ eines Prozesses und deren Abfolge als auch die Zuordnung der Aktivitäten zu den Rollenträgern festzulegen. Ferner werden die für die einzelnen Aktivitäten verwendeten Applikationen und Informationsobjekte festgelegt. Die Erstellung der Ablaufplanung basiert im Wesentlichen auf der EPK.³⁷⁶ Zunächst müssen die einzelnen Aktivitäten innerhalb eines Prozesses identifiziert werden. Dabei sind diejenigen Aktivitäten aus der Geschäftsstrategie abzuleiten, welche ein Unternehmen bzw. eine Geschäftseinheit effizient ausführen können muss. Die Auslöser einer Aktivität (Ereignisse) werden zu diesem Zeitpunkt noch nicht definiert, sondern nur die logische Folge der einzelnen Aktivitäten.³⁷⁷ Anschliessend müssen die an den jeweiligen Aktivitäten beteiligten Personen, Stellen oder Organisationseinheiten festgelegt und den Aktivitäten zugeordnet werden. Sind mehrere Organisationseinheiten an einer Aktivität beteiligt, so kann durch Angabe der Art der Beteiligung genauer differenziert werden. Zuletzt können die für den Geschäftsablauf relevanten Ereignisse in das Modell aufgenommen werden.³⁷⁸ Ereignisse lösen Nachrichten aus, welche wiederum Aktivitäten auslösen können. Eine Verknüpfung von Ereignissen und Aktivitäten wird durch die Verwendung von Operatoren (UND, Inklusiv-Oder, Exklusiv-Oder) unterstützt. Als Ergebnisdokument erhält man eine EPK, die den Ablauf eines bestimmten Prozesses darstellt.

Im nächsten Schritt können, ausgehend von den Prozessaktivitäten, die jeweiligen Rollenträger innerhalb der Ablaufplanung näher bestimmt und beschrieben werden. Diese stellen die Basis für die Erstellung der Stellenanforderung bzw. der Stellenbeschreibung dar. Durch Schaffung der beschriebenen Stellen, Stellentypen oder Organisationseinheiten und deren Beziehungen wird die *Aufbauorganisation konkretisiert* und *festgelegt*. Diese stellt die Organisationseinheiten in hierarchische Beziehungen zueinander. Untergeordnete Organisationseinheiten berichten an die ihnen übergeordnete Organisationseinheit und haben Weisungsbefugnis gegenüber den ihnen untergeordneten Organisationseinheiten.

Im letzten Schritt wird eine *Informationslandkarte erstellt*. Die Informationslandkarte stellt die in einem Unternehmen bzw. einer Geschäftseinheit zur Verfügung stehenden Informationen aus verschiedenen Blickwinkeln dar.³⁷⁹ Die einzelnen Informationen zur Erstellung der Informationslandkarte werden durch eine Inventur des Berichtswesens gesammelt. Dazu müssen die im Unternehmen zur Verfügung stehenden Berichte erfasst und ausgewertet werden.

³⁷⁵ Der Begriff Aktivität wird in dieser Arbeit synonym zu den Begriffen Funktion, Aufgabe, Vorgang und Tätigkeit verwendet.

³⁷⁶ EPK steht für Ereignisgesteuerte Prozesskette. Vgl. im Folgenden Scheer (2001) und Scheer/Thomas (2005). Auf die Gründe zur Wahl dieser Modellierungstechnik wird in Abschnitt 4.2.2.5 genauer eingegangen.

³⁷⁷ Vgl. Scheer (2001), S. 31.

³⁷⁸ Vgl. Scheer (2001), S. 124.

³⁷⁹ Für eine detaillierte Beschreibung der einzelnen Schritte zur Erstellung der Informationslandkarte vgl. Strauch (2002), Strauch/Winter (2002) und Winter/Strauch (2004).

4.2.2 Metamodell

Das Metamodell der Organisationsebene lässt sich entsprechend den bei der Organisationsgestaltung erstellten Ergebnisdokumententypen in die Sichten Prozesslandkarte, Prozessvision, Ablaufplanung, Prozessführung, Aufbauorganisation und Informationslandkarte einteilen.

4.2.2.1 Metamodell der Prozesslandkarte

Eine Prozesslandkarte ist eine grafische Übersicht über alle relevanten Prozesse mit ihren Leistungs- und Informationsbeziehungen zu anderen Prozessen innerhalb und ausserhalb des Unternehmens bzw. der Geschäftseinheit.³⁸⁰ Ein **Prozess** wird dabei als eine zusammengehörende Abfolge von Unternehmensverrichtungen zum Zweck einer Leistungserstellung definiert. Input bzw. Output eines Prozesses sind seine Leistungen, die von internen oder externen Kunden angefordert und abgenommen werden.³⁸¹

Jeder Prozess der Prozesslandkarte wird einem von drei Prozesstypen zugeordnet. Der Typ **Leistungsprozess** beschreibt Prozesse zur Erstellung und Vermarktung der Produkte und Dienstleistungen des Unternehmens. Die Leistungsprozesse werden auf dem Weg vom Erkennen eines Kundenbedürfnisses bis zu seiner Befriedigung durchlaufen.³⁸² Dies beinhaltet nicht nur die gesamte Auftragsabwicklung von der Anfrage über die Beschaffung und Produktion bis zur Auslieferung, sondern genauso die Akquisition der Kunden, die Pflege der Kundenbeziehungen oder auch die Betreuung des Kunden nach der Abwicklung eines Auftrags.³⁸³ Leistungsprozesse geben direkt Leistungen an Kunden ab und erzeugen damit unmittelbar Nutzen für ihre Abnehmer. In der Regel werden die Leistungsprozesse anhand der Produkte bzw. Leistungen, die ein Unternehmen seinen externen Kunden anbietet, definiert.³⁸⁴

Dem Prozesstyp **Unterstützungsprozess** werden alle Prozesse zugeordnet, die die Ressourcen zur Leistungserstellung aufbauen und pflegen.³⁸⁵ Sie ermöglichen die kontinuierliche Ausführung mehrerer Leistungsprozesse, insbesondere durch die Bereitstellung von Ressourcen (Personal, Material und Hilfsstoffe), die Bereitstellung und Pflege der Infrastruktur und die (Weiter-) Entwicklung der Produkte und Dienstleistungen.³⁸⁶

³⁸⁰ Vgl. *IMG* (1997), S. 17.

³⁸¹ Vgl. *Scheer* (1998), S. 3.

³⁸² Vgl. *Österle* (1995), S. 130.

³⁸³ Vgl. Glossar in *IMG* (1997).

³⁸⁴ Vgl. *Jonkers et al.* (2004), S. 268.

³⁸⁵ Vgl. *Österle* (1995), S. 130.

³⁸⁶ Vgl. Glossar in *IMG* (1997).

Der Prozesstyp **Führungsprozess** fasst alle Prozesse zusammen, die die übergreifende Planung, Steuerung und Kontrolle des zu führenden Objektes (z.B. andere Prozesse) gewährleisten. Ihre Ergebnisse gehen an die Leistungs- und Unterstützungsprozesse.³⁸⁷

Die von einem Prozess erzeugte Leistung (**Prozessleistung**) geht an interne oder externe Kunden bzw. an deren Prozesse. Empfänger einer Prozessleistung ist somit ein anderer Prozess innerhalb oder ausserhalb der Unternehmung bzw. Geschäftseinheit. Ist der Leistungsempfänger ein Prozess ausserhalb der eigenen Unternehmung, ist eine Prozessleistung Teil einer Marktleistung.³⁸⁸ Eine Prozessleistung kann materiell (Sachleistung) oder immateriell (Dienstleistung) sein und mit den zu ihrer Erstellung benötigten Kosten bewertet werden.³⁸⁹ Dienstleistungen können weiter in Informations- und sonstige Dienstleistungen unterteilt werden. Handelt es sich um eine Informationsleistung, die zwischen zwei Prozessen ausgetauscht wird, so besteht zwischen den Prozessen eine **Informationsbeziehung**. Andernfalls besteht zwischen diesen Prozessen eine **Leistungsbeziehung**. Die zwischen den Prozessen der Prozesslandkarte ausgetauschten Leistungen können in einem eigenen (Prozess-) Leistungsmodell in hierarchischer Form abgebildet werden.

Jeder Prozess der Prozesslandkarte lässt sich durch einen oder mehrere Prozesse spezialisieren (Generalisierungs-Spezialisierungs-Zusammenhang) oder in detailliertere (Teil-)Prozesse zerlegen (Aggregations-Dekompositions-Zusammenhang).³⁹⁰

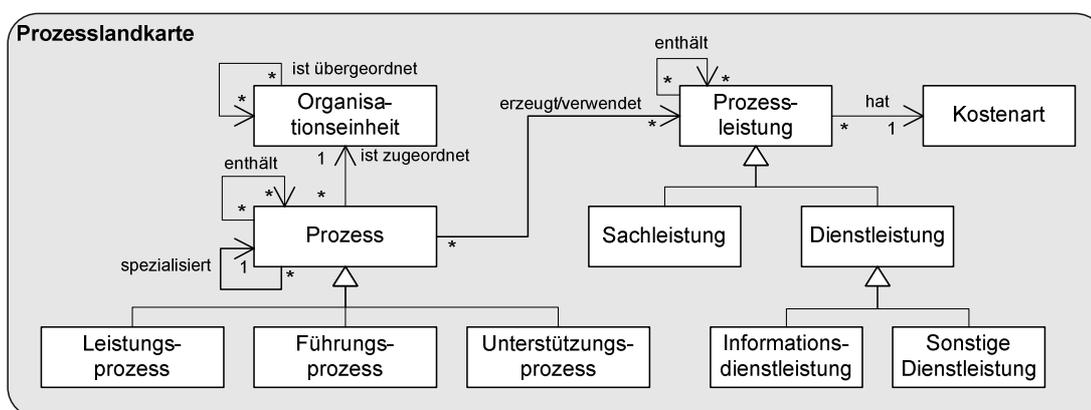


Abbildung 39: Metamodell der Prozesslandkarte

4.2.2.2 Metamodell der Prozessvision

Als wichtige Ideenquellen für neue Lösungen und damit als Grundlage für die Prozessvision können Mitarbeiter und deren Know-how, Beispiele vorhandener innovativer Lösungen, In-

³⁸⁷ Vgl. Gossar in *IMG* (1997)

³⁸⁸ Vgl. Brecht (2002), S. 170.

³⁸⁹ SCHEER differenziert den generellen Leistungs- bzw. Produktbegriff nach Sach- und Dienstleistungen. Vgl. Scheer (2001), S. 93.

³⁹⁰ Vgl. hierzu Abschnitt 2.2.1 und Winter (2003d), S. 90-91.

formationstechnologien und deren Potenziale sowie die Unternehmensstrategie bzw. Vorgaben aus dieser herangezogen werden.³⁹¹

Jedem Prozess der Prozesslandkarte kann eine Prozessvision zugeordnet werden. Die Prozessvision beschreibt anhand von **Prozessgrundsätzen**, wie der Prozess gestaltet sein sollte. Die Prozessgrundsätze werden unterteilt in die Dimensionen Leistungen, Ablauf, Erfolgsfaktoren und IT-Unterstützung. Dabei besitzt jede Dimension ein oder mehrere **Ausprägungen** und jede Ausprägung kann in der Regel auf genau eine Dimension zurückgeführt werden. Anhand der prozessspezifischen **IT-Unterstützung** können diejenigen Informationstechniken zusammengefasst werden, welche für die Prozessunterstützung als geeignet erscheinen. Die Dimension **Leistungen** beschreibt die von einem Prozess zu erzeugenden Leistungen sowie die Partner, die für die Leistungserstellung mit einzubeziehen sind. Die Dimension **Ablauf** fasst Hinweise aus der Strategiegestaltung zusammen, die bei der Modellierung des Prozessablaufs bzw. der Festlegung der einzelnen Aktivitäten sowie deren Reihenfolge zu berücksichtigen sind. Im Hinblick auf die Prozessführung werden Erfolgsfaktoren zur Bewertung und Steuerung des Prozesses in der Dimension **Prozessführung** zusammengefasst.

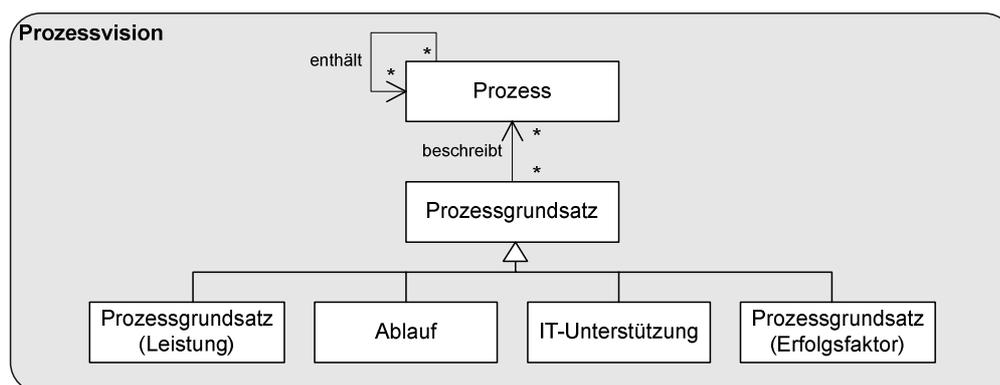


Abbildung 40: Metamodell der Prozessvision

4.2.2.3 Metamodell der Prozessführung

Die Steuerung des Prozesses wird durch die Prozessführung bestimmt. Sie plant Soll-Werte, erhebt Ist-Werte, legt Erfolgsfaktoren und Führungsgrößen fest und löst notwendige Massnahmen aus.³⁹² Input erhält die Prozessführung aus der Geschäftsstrategie in Form von aus der Balanced Scorecard abgeleiteten Kennzahlen und aus den Prozessgrundsätzen der Prozessvision.

Die folgende Beschreibung der Prozessführung enthält einige Begriffe, die auch auf Strategieebene Verwendung finden. Diese beziehen sich bei der Modellierung der Organisationsebene auf den Prozesskontext.

³⁹¹ Vgl. Techniken in *IMG* (1997) und *Hess* (1996), S. 183-189.

³⁹² Vgl. *Österle* (1995), S. 54.

Ein **kritischer Erfolgsfaktor** (KEF) der Organisationsebene beschreibt ein erfolgsentscheidendes Merkmal eines Prozesses. Er wird aus der Geschäftsstrategie (BSC) und der Prozessvision abgeleitet und durch eine oder mehrere Führungsgrößen operationalisiert.³⁹³ Typische Merkmale und damit kritische Erfolgsfaktoren eines Prozesses, die aus Kundensicht relevant sind, sind beispielsweise Qualität, Zeit, Kosten und Flexibilität.

Eine **Führungsgröße** beschreibt einen quantitativ messbaren, für die Führung eines Prozesses relevanten Sachverhalt. Sie operationalisiert einen oder mehrere Erfolgsfaktoren eines Prozesses und dient der Planung und Beurteilung der Prozessqualität im Sinne der kritischen Erfolgsfaktoren.³⁹⁴

Der Wert einer Führungsgröße, der für einen bestimmten Zeitpunkt angestrebt wird, wird durch das **Prozessziel** definiert. Es ist das Ergebnis der Planungsaktivitäten und verkörpert Vorgaben, auf die die Entscheidungen auszurichten sind.³⁹⁵ Sie bilden die Grundlage für die Soll-Ist-Vergleiche im Rahmen der Kontrolle und Neuausrichtung der Prozesse.

Die Prozessziele, kritischen Erfolgsfaktoren und Führungsgrößen werden in Berichten zusammengefasst. Ein **Bericht** stellt das Ausmass und die Ursachen von Abweichungen zwischen den geplanten und den effektiven Ergebnissen dar und geht an eine oder mehrere Organisationseinheiten.³⁹⁶ Im Hinblick auf eine informationstechnisch unterstützte Prozessführung kann jedem Prozess eine Applikation im Sinne eines **Messsystems** zugeordnet werden. Diese misst die für den Prozess definierten Führungsgrößen und erstellt aus den Messergebnissen in regelmässigen Abständen einen elektronischen Bericht, der an die verantwortliche(n) Organisationseinheit(en) geschickt wird.

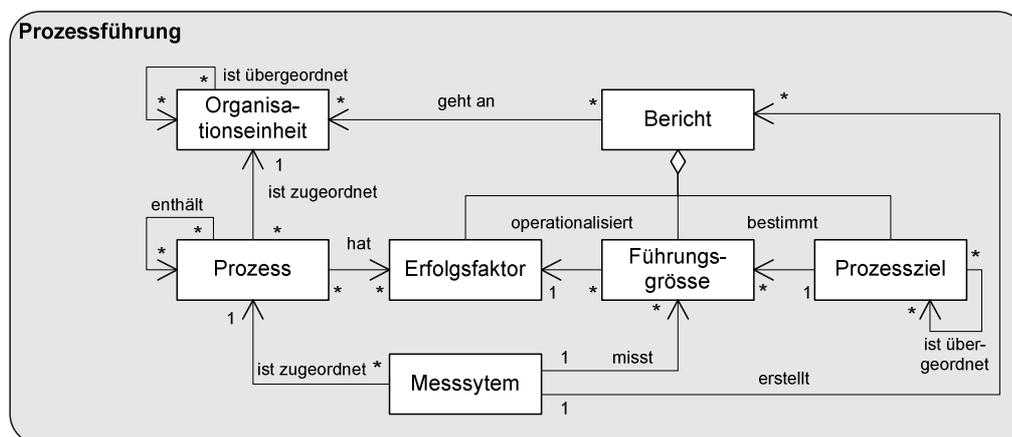


Abbildung 41: Metamodell der Prozessführung

³⁹³ Vgl. Brecht (2002), S. 159.

³⁹⁴ Vgl. Österle (1995), S. 54.

³⁹⁵ Vgl. Brecht (2002), S. 178.

³⁹⁶ Vgl. Brecht (2002), S. 149.

Eine **Organisationseinheit des Prozessmanagements** (OE des PM) stellt eine speziell für die Führung eines Prozesses eingerichtete organisatorische Einheit dar.³⁹⁷ Dadurch wird die Führung eines Prozesses Teil der Organisationsstruktur eines Unternehmens.

4.2.2.4 Metamodell der Aufbauorganisation

Das Modell der Aufbauorganisation beschreibt die Organisationseinheiten eines Unternehmens mit den zwischen ihnen bestehenden Kommunikations- und Weisungsbeziehungen.³⁹⁸ Es dient dazu, die komplexen Strukturen einer Organisation in vereinfachter Form darzustellen, indem gleichartige Aufgabenkomplexe zu Organisationseinheiten zusammengefasst werden. Abbildung 42 zeigt das Metamodell der Aufbauorganisation.

Eine **Organisationseinheit** bezeichnet einen eigenständigen, permanenten Teil der organisatorischen Struktur eines Unternehmens (z.B. Abteilung, Division, Geschäftseinheit).³⁹⁹ Sie kann mehr oder weniger aggregiert sein und eine Geschäftseinheit oder das Unternehmen selbst darstellen. Die Über- und Unterordnungsbeziehungen geben an, in welche organisatorischen Einheiten und Untereinheiten sich eine Organisation untergliedern lässt. Sie spiegeln gleichzeitig die Ergebnisse der Abteilungs- und Bereichsbildung wider.⁴⁰⁰ Auf der obersten Aggregationsstufe entspricht die organisatorische Einheit der gesamten Unternehmung. Unterstützungsbeziehungen bestehen z. B. zwischen den Stabs- und den Linienabteilungen.

Organisationseinheiten bilden die Grundstruktur der Unternehmung (Primärorganisation). Alternative Strukturen (z. B. organisatorische Einheiten auf Zeit oder für periodisch wiederkehrende Aufgaben) können diese Grundstruktur überlagern. Man spricht dann von temporären Organisationseinheiten. Eine temporär bestehende Organisationseinheit wird durch den Metaentitätstyp **sekundäre Organisationseinheit** repräsentiert. Sekundäre Organisationseinheiten sind in der Regel orthogonal zu den dauerhaften Organisationseinheiten, repräsentiert durch den Metaentitätstyp **primäre Organisationseinheit**, positioniert.

Die Führung von Prozessen führt in der Regel zu einer Überlagerung der Primärstruktur.⁴⁰¹ Daher wird zusätzlich das Konstrukt **Organisationseinheit des Prozessmanagement** (OE des PM) eingeführt. Eine OE des PM ist eine speziell für die Führung eines Prozesses eingerichtete Organisationseinheit.⁴⁰² Bei einer reinen Prozessorganisation ist die OE des PM nicht erforderlich, da die Prozesse zu primären organisatorischen Einheiten werden.

³⁹⁷ Vgl. Glossar in *IMG* (1997) oder *Brecht* (2002), S. 165.

³⁹⁸ Vgl. *Scheer* (2001), S. 52.

³⁹⁹ Vgl. Glossar in *IMG* (1997).

⁴⁰⁰ Vgl. *Brecht* (2002), S. 164.

⁴⁰¹ Vgl. *Brecht* (2002), S. 164.

⁴⁰² Vgl. Glossar in *IMG* (1997).

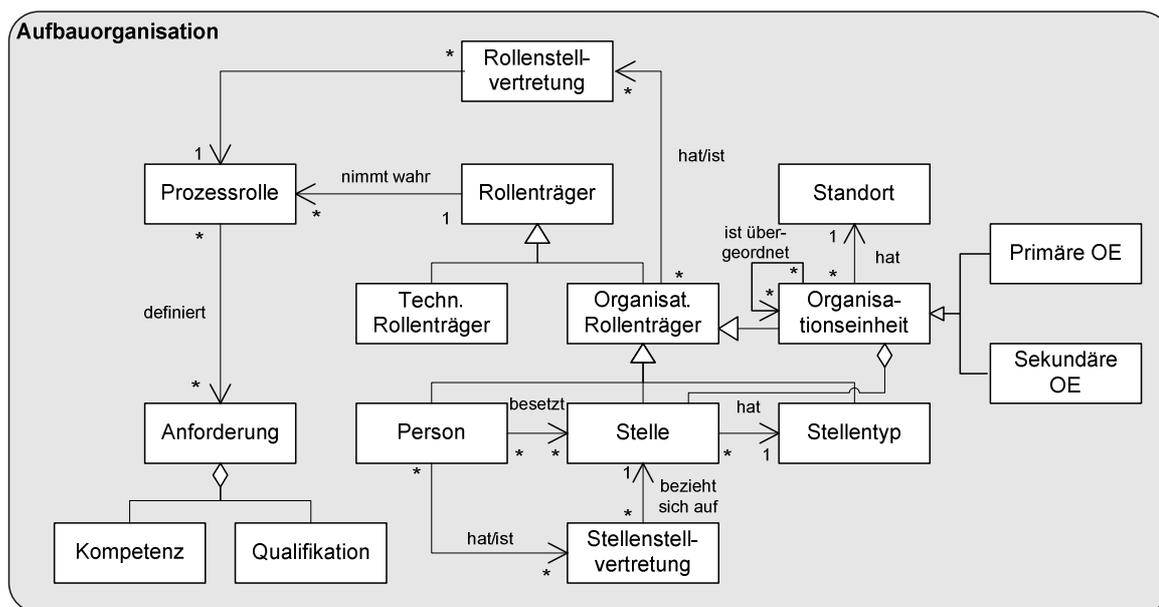


Abbildung 42: Metamodell der Aufbauorganisation

Eine Organisationseinheit besteht aus mehreren Stellen, wobei eine **Stelle** eine Zusammenfassung von Aufgaben beschreibt, die einer Person übertragen werden können und diese dauerhaft bei definierter, im Regelfall kontinuierlicher Arbeitszeit auslasten.⁴⁰³ Die Stelle stellt somit, im Gegensatz zur Rolle, ein dauerhaftes, langfristiges Konstrukt innerhalb der Aufbauorganisation dar. Stellen mit gleichen Kompetenzen können zu einem **Stellentyp** zusammengefasst werden.

Die **(Prozess-)Rolle** repräsentiert ein organisatorisches Konstrukt, das als Bindeglied zwischen Aufbau- und Ablauforganisation dient. Sie bündelt einzelne Aktivitäten der Ablauforganisation, die letztendlich von Menschen (organisatorischen Rollensträgern) oder Maschinen (technischen Rollensträgern) wahrgenommen werden. Darüber hinaus definiert die Prozessrolle zum einen die für die Ausführung einer Aktivität notwendige minimale **Qualifikation** (z. B. „spricht Spanisch“), zum anderen beschreibt eine Rolle **Kompetenzen**, welche einem Rollensträger übertragen werden (z. B. „ist zeichnungsberechtigt“).⁴⁰⁴

Eine Stelle kann von einer bestimmten **Person** oder aber auch im Sinne von Teilzeitstellen und Schichtarbeit von mehreren Personen besetzt werden. Zudem wird keine zwingende Zuordnung zwischen Person und Stelle gefordert, so dass auch freie Mitarbeiter im Modell der Aufbauorganisation erfasst werden können.⁴⁰⁵

Um eine Zuordnung zwischen einer Prozessrolle und unterschiedlichen organisatorischen oder technischen Ressourcen zu ermöglichen, enthält das Metamodell der Aufbauorganisation

⁴⁰³ Vgl. Rosemann/zur Mühlen (1997), S. 80.

⁴⁰⁴ Vgl. Rosemann/zur Mühlen (1997), S. 79.

⁴⁰⁵ Vgl. Rosemann/zur Mühlen (1997), S. 80.

den abstrakten Metaentitätstyp **Rollenträger**. Ein Rollenträger kann in einen **organisatorischen Rollenträger** und einen **technischen Rollenträger** differenziert werden.⁴⁰⁶ Zu den organisatorischen Rollenträgern gehören die Konstrukte **Stelle**, **Person**, **Stellentyp** und **Organisationseinheit**. Dadurch kann eine Rolle von einem dieser vier Konstrukte der Aufbauorganisation übernommen werden. Technische Rollenträger sind beispielsweise Maschinen oder Informationssysteme. Sie können dementsprechend weiter nach **materialverarbeitend** und **informationsverarbeitend** unterschieden werden.⁴⁰⁷

Ein organisatorischer Rollenträger kann in Bezug auf eine bestimmte Prozessrolle eine **Rollenstellvertretung** haben.⁴⁰⁸ Im Rahmen dieser Rollenstellvertretung wird ein Rolleninhaber durch einen oder mehrere Stellvertreter vertreten. Analog lässt sich eine **Stellenstellvertretung** definieren.

Die geographische Verteilung der Organisationseinheiten und Stellen wird durch den Metaentitätstyp **Standort** abgebildet. Ein Standort ist ein geographischer Punkt, an dem sich eine oder mehrere organisatorische Einheiten bzw. Stellen befinden.

4.2.2.5 Metamodell der Ablauforganisation

Die Ablauforganisation bildet, basierend auf der Prozessvision, die konkreten Aktivitäten⁴⁰⁹ eines Prozesses sowie ihre Abfolge und ihre Zuordnung zu den Rollenträgern ab.

Für die Modellierung von Prozessabläufen existieren bereits zahlreiche Modellierungssprachen. Vor allem wegen ihrer hohen Verbreitung und ihrer Ausdrucksmächtigkeit werden sehr häufig die ereignisgesteuerten Prozessketten (EPK)⁴¹⁰ oder auch UML-Aktivitätsdiagramme⁴¹¹ eingesetzt. Während die EPK speziell für die Modellierung von Geschäftsprozessen entwickelt wurde, fokussiert die UML dagegen auf die objektorientierte Entwicklung von Softwaresystemen. Die im Folgenden beschriebenen Metaentitätstypen und deren Repräsentationen für die Abbildung von Prozessabläufen basieren daher primär auf der EPK bzw. eEPK (erweiterte ereignisgesteuerte Prozesskette). Zudem wird mit der Verwendung der EPK als Grundlage für die Entwicklung des Metamodells der Ablauforganisation eine Modellierungssprache gewählt, die formal spezifiziert ist⁴¹² und die über eine ausreichende semantische Ausdrucksmächtigkeit verfügt.

⁴⁰⁶ Vgl. *Scheer* (2001), S. 56. SCHEER nimmt in der Organisationssicht von ARIS eine ähnliche Differenzierung vor. Er unterscheidet dabei zwischen primär menschlichen und primär technischen Leistungsträgern.

⁴⁰⁷ Vgl. z.B. *Scheer* (2001), S. 56.

⁴⁰⁸ Vgl. *Wortmann* (2005), S. 120.

⁴⁰⁹ Synonym werden oftmals die Begriffe Funktion, Vorgang, Tätigkeit oder Aufgabe verwendet.

⁴¹⁰ Vgl. z.B. *Becker et al.* (2002b) und *Scheer/Jost* (2002).

⁴¹¹ Vgl. z.B. *Jeckle et al.* (2004b).

⁴¹² Vgl. *Scheer* (2001), S. 170-171 und *Rosemann* (1996), S. 122-123.

Die EPK ist ein wesentliches Element des ARIS-Konzepts (Architektur integrierter Informationssysteme).⁴¹³ Mit EPKs können Geschäftsprozesse grafisch dargestellt werden. Wesentliche Elemente einer EPK sind Ereignisse, Funktionen und Verknüpfungsoperatoren.⁴¹⁴ Jede Funktion kann zusätzlich mit einem Informationsobjekt verbunden werden, dessen Information bei der Durchführung einer Funktion manipuliert wird. Reichert man die Funktionen der EPK mit zusätzlichen Informationen über ausführende Personen, unterstützende Systeme, verwendete Daten etc. an, erhält man eine so genannte erweiterte EPK (eEPK), mit der die Verbindung zu anderen Modellen und Sichten des ARIS-Konzepts hergestellt werden kann.⁴¹⁵

Da der in dieser Arbeit verfolgte Ansatz im Sinne einer Unternehmensarchitektur die wesentlichen Metaentitätstypen definieren soll, werden auch für die Modellierung der Ablauforganisation lediglich ausgewählte Metaentitätstypen der EPK verwendet und definiert. Diese sind in der Regel auch in anderen Modellierungssprachen in ähnlicher Form vorhanden. Im Mittelpunkt steht vor allem die Abfolge und Nebenläufigkeit von Aktivitäten sowie deren Zuordnung zu den Rollenträgern. Für ein umfassendes und detailliertes Prozessmanagement sollte aus diesem Grunde ein entsprechend spezialisiertes Metamodell verwendet werden.

Die wesentlichen Metaentitätstypen des Metamodells der Ablauforganisation sind im Ansatz dieser Arbeit, in Anlehnung an die EPK, Aktivitäten⁴¹⁶, Ereignisse und Verknüpfungsoperatoren (vgl. Abbildung 43).

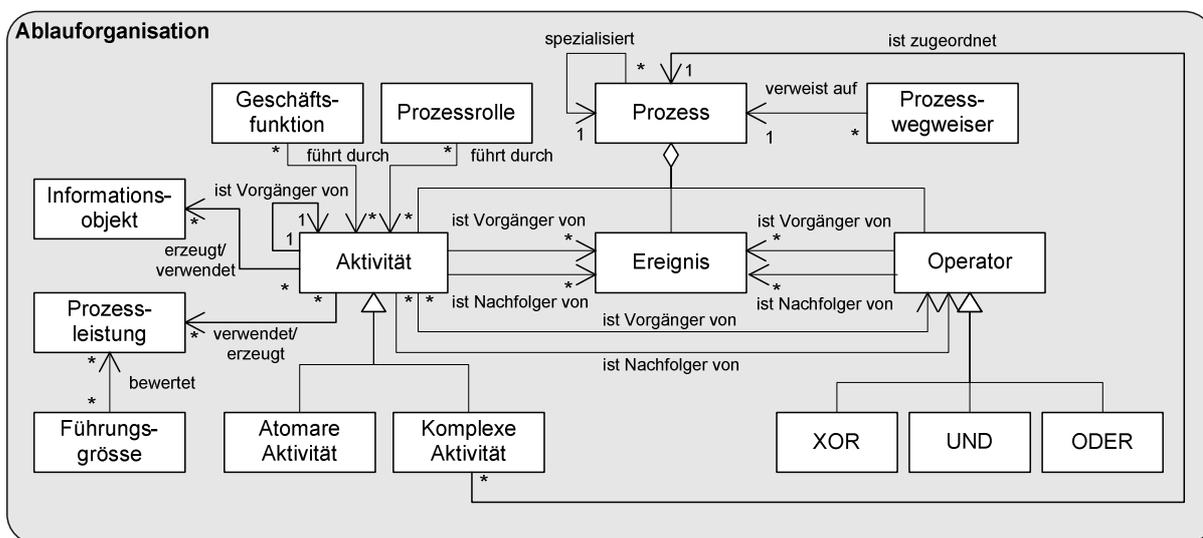


Abbildung 43: Metamodell der Ablauforganisation

Eine **Aktivität** ist eine betriebliche Tätigkeit mit einem definierten Handlungsziel, die ein Objekt manipuliert und von einer Rolle bzw. einem (organisatorischen oder technischen) Rol-

⁴¹³ Vgl. z.B. Scheer/Jost (2002).

⁴¹⁴ Vgl. hierzu auch Abschnitt 3.2.1.2.

⁴¹⁵ Vgl. hierzu die in Abschnitt 3.2.1 beschriebenen Metamodelle der unterschiedlichen Sichten von ARIS.

⁴¹⁶ Eine Funktion der EPK wird im Metamodell dieser Arbeit als Aktivität bezeichnet.

lenrtäger durchgeführt wird.⁴¹⁷ Sie stellen die aktiven Elemente der Ablauforganisation dar. Eine Aktivität wird von einer oder mehreren Geschäftsfunktionen durchgeführt. Eine **Geschäftsfunktion** ist eine Verarbeitungseinheit, die aus einer geschäftlichen (betriebswirtschaftlichen) Perspektive erhoben wurde.⁴¹⁸ Sie definiert eine bestimmte Funktionalität, die für die Durchführung einer oder mehrerer Aktivitäten benötigt wird. Die Geschäftsfunktionen stellen das Bindeglied zur Applikationsunterstützung von Aktivitäten der Geschäftsprozesse dar.⁴¹⁹ Durch die Definition von Geschäftsfunktionen wird somit festgelegt, welche Aktivitäten der Geschäftsprozesse automatisiert bzw. automatisierbar sind.

Im Gegensatz zu einer Aktivität, die ein zeitverbrauchendes Geschehen darstellt, ist ein **Ereignis** auf einen Zeitpunkt bezogen.⁴²⁰ Ereignisse sind die passiven Elemente der Ablauforganisation.

Eine **Kontrollflussbeziehung** verbindet Ereignisse und Aktivitäten.⁴²¹ Ereignisse können sowohl Aktivitäten auslösen als auch deren Ergebnis sein. Um auszudrücken, dass eine Aktivität durch ein Ereignis oder mehrere gestartet wird bzw. eine Aktivität ein Ereignis oder mehrere auslösen kann, wird der Metaentitätstyp **Verknüpfungsoperator** verwendet.⁴²² Verknüpfungsoperatoren können nach konjunktiven, adjunktiven und disjunktiven⁴²³ Verknüpfungen differenziert werden. Die entsprechenden Operatoren werden als **AND**-, **OR**- und **XOR**-Operatoren bezeichnet. Des Weiteren können die Verknüpfungsoperatoren dahingehend unterschieden werden, ob sie eine eingehende und mehrere ausgehende (**Split-Operator**) oder mehrere eingehende und eine ausgehende Kontrollflusskante (**Join-Operator**) besitzen.

Das Metamodell der EPK zwingt zu einer abwechselnden Folge von Aktivitäten (Funktionen) und Ereignissen. Dadurch wird der Ablauf unnötig verlängert, wodurch unter Umständen ein unverständliches und unübersichtliches Diagramm entsteht. Dies trifft insbesondere dann zu, wenn Aktivitäten mit einem Verb im Infinitiv beschrieben werden (z.B. „Beleg bearbeiten“) und die anschließenden Ereignisse lediglich das Verb in ein Partizip umsetzen (z.B. „Beleg bearbeitet“). Deshalb wird im Metamodell dieser Arbeit auf diese Einschränkung verzichtet. Eine Aktivität kann folglich auch ein direkter Nachfolger bzw. Vorgänger einer anderen Aktivität sein. Semantisch bedeutet dies, dass die nachfolgende Aktivität gestartet wird, sobald die

⁴¹⁷ Vgl. *Brecht* (2002), S. 147, *Österle* (1995), S. 50 und *Scheer* (2001), S. 22. BRECHT und ÖSTERLE verwenden synonym den Begriff Aufgabe, SCHEER verwendet dagegen die Bezeichnung Funktion. KELLER ET AL. setzen den Funktionsbegriff in der EPK mit dem der Aufgabe gleich. Vgl. *Keller et al.* (1992), S. 8.

⁴¹⁸ Vgl. *Schwinn* (2005), S. 19 und *Gutzwiller* (1994), S. 81.

⁴¹⁹ Vgl. *Jonkers et al.* (2004), S. 268.

⁴²⁰ Vgl. *Scheer/Thomas* (2005), S. 5.

⁴²¹ Vgl. *Scheer/Thomas* (2005), S. 6.

⁴²² Vgl. *Scheer/Thomas* (2005), S. 6.

⁴²³ In Anlehnung an die Terminologie der Aussagenlogik.

vorhergehende Aktivität beendet wurde. Nebenläufige Aktivitäten müssen weiterhin durch Verknüpfungsoperatoren modelliert werden.

Aus Gründen der Übersichtlichkeit und besseren Lesbarkeit wird der Ablauf von Prozessen in der Regel nicht durch ein einziges Modell repräsentiert, sondern in mehrere Teilmodelle zerlegt.⁴²⁴ Dabei ist sowohl eine Schachtelung als auch eine Zerlegung der Modelle bzw. Aktivitäten möglich. Die Schachtelung von Aktivitäten dient der **Hierarchisierung** bzw. Dekomposition von Ablaufmodellen. Bei der Dekomposition wird die übergeordnete Aktivität (**Komplexe Aktivität**) mit dem Namen des entsprechenden Detailmodells bezeichnet. Ein **Prozesswegweiser** ermöglicht dagegen die Verknüpfung von zwei Ablaufmodellen, die den gleichen Detaillierungsgrad besitzen. Dadurch kann ein Prozess in Teilprozesse zerlegt werden, ohne eine „künstliche“ Hierarchisierung erzeugen zu müssen. Der Prozesswegweiser wird, analog zur Dekomposition, mit dem Namen des Modells bezeichnet, auf das er verweist.

Modelle der Ablauforganisation dienen oftmals als Grundlage für die Entwicklung von Workflow-Modellen.⁴²⁵ Sie stellen Ergebnisdokumente dar, die gemäss den verfolgten Zwecken um workflowspezifische Informationen angereichert oder in neue Ergebnisdokumente transformiert werden. Um die Transformation in entsprechende Workflow-Modelle zu erleichtern, sollte sich die Konstruktion von Ablaufmodellen nach einer bestimmten, vorgegebenen Syntax richten. Regeln für die Konstruktion syntaktisch richtiger Modelle der Ablauforganisation definieren beispielsweise NÜTTGENS/RUMP.⁴²⁶

Die Ablauforganisation stellt, analog zur Steuerungssicht in ARIS (EPK)⁴²⁷, die zentrale Sicht der Organisationsebene dar, in der Metaentitätstypen aus anderen Sichten mit Aktivitäten in Beziehung gesetzt werden. So können beispielsweise Instanzen der Metaentitätstypen organisatorischer oder technischer Rollenträger, Prozessleistung, Informationsobjekt oder Applikation mit einer bestimmten Aktivität innerhalb eines Prozessablaufs assoziiert sein. Dadurch wird die Ablauforganisation mit anderen Sichten der Unternehmensarchitektur, wie z.B. der Aufbauorganisation oder Applikationsbestandsführung, verbunden. Im Unterschied zum Metamodell der EPK können zudem die von einer Aktivität erzeugten Prozessleistungen durch Führungsgrössen bewertet werden. Diese Führungsgrössen stellen wichtige Indikatoren für

⁴²⁴ Vgl. z.B. *Scheer/Thomas* (2005), S. 11 und *Jeckle et al.* (2004b), S. 216.

⁴²⁵ Vgl. hierzu z.B. *Holten et al.* (1997), S. 10-12.

⁴²⁶ Für eine detaillierte Beschreibung der Syntax und Semantik von EPKs siehe z.B. *Nüttgens/Rump* (2002).

⁴²⁷ Die EPK stellt die zentrale Modellierungssprache des ARIS-Sichtenkonzepts dar. An EPK-Funktionen können zusätzliche Sprachkonstrukte aus anderen Sichten, wie z.B. Umfelddaten, menschliche Arbeitsleistung, maschinelle Ressourcen, Anwendungssoftware, Leistungen, Finanzmittel, Organisationseinheiten oder Unternehmensziele annotiert werden. Dadurch kann neben dem Kontrollfluss auch zwischen Organisations-/Ressourcen-, Informations-, Leistungs- sowie Finanzmittelfluss unterschieden werden. Vgl. *Scheer/Thomas* (2005), S. 7-8.

die Effektivität und Effizienz eines Prozesses bzw. einer Aktivität dar.⁴²⁸ Sie verknüpfen somit die Ablauforganisation mit der Prozessführung.

Das Metamodell der EPK ermöglicht eine sehr detaillierte Spezifikation von Prozessabläufen. Dies widerspricht dem Anspruch der groben und aggregierten Darstellung der Zusammenhänge der Unternehmensarchitektur (vgl. Abschnitt 2.3.1). Deshalb wird alternativ zu dem in dieser Arbeit leicht vereinfachten Metamodell der EPK das Metamodell von Ablaufketten⁴²⁹ empfohlen. Dieses ermöglicht lediglich die Abbildung der groben Ablauffolge von Aktivitäten eines Prozesses sowie deren Zuordnung zu Rollenträgern und garantiert somit einen für die Modellierung der Unternehmensarchitektur angemesseneren Detaillierungsgrad.

4.2.2.6 Metamodell der Informationslandkarte

Das Ziel einer Informationslandkarte ist es, die in einem Unternehmen bzw. einer Geschäftseinheit zur Verfügung stehenden Informationen zu identifizieren und aus verschiedenen Blickwinkeln darzustellen.⁴³⁰ Für ihre Erstellung müssen Informationen zu den im Folgenden beschriebenen Metaentitätstypen gesammelt und ausgewertet werden.⁴³¹

Die einzelnen Informationen werden durch eine Inventur des Berichtswesens gesammelt. Dazu müssen die im Unternehmen zur Verfügung stehenden Berichte erfasst werden. Ein **Bericht** stellt einer oder mehreren Organisationseinheit(en) Informationen nach bestimmten Kriterien (Dimensionen) zur Verfügung.⁴³² Er kann von einem bestimmten **Messsystem** elektronisch erzeugt und hierarchisch strukturiert sein. So kann ein Bericht beispielsweise auf bereits zuvor definierten Berichten basieren. Ein Bericht im Sinne der Prozessführung fasst Prozessziele, kritische Erfolgsfaktoren und Führungsgrößen eines bestimmten **Prozesses** zusammen und stellt das Ausmass und die Ursachen von Abweichungen zwischen den geplanten und den effektiven Ergebnissen des Prozesses dar.⁴³³

Für jeden Bericht sind dessen Konsumenten und Produzenten zu identifizieren. Der Metaentitätstyp **Konsument** beschreibt die Stelle (und damit indirekt die Person), die den Bericht nachfragt, sowie die Organisationseinheit, in welcher die Person arbeitet. Dementsprechend gibt der Metaentitätstyp **Produzent** die Person an, die für die Produktion des Berichts zuständig ist sowie die Organisationseinheit, in welcher die Person arbeitet. Darüber hinaus müssen die in dem Bericht aufgelisteten Informationsobjekte (Berichtspositionen)⁴³⁴ erfasst werden.

⁴²⁸ Vgl. Glossar in *IMG* (1997).

⁴²⁹ Vgl. z.B. Ablaufplanung in *IMG* (1997) und *Brecht* (2002), S. 138 und S. 362f.

⁴³⁰ Vgl. *Strauch* (2002), S. 180. Vgl. im Folgenden auch *Strauch/Winter* (2002) und *Winter/Strauch* (2004).

⁴³¹ Vgl. *Strauch* (2002), S. 181-182.

⁴³² Vgl. *Strauch* (2002), S. 139.

⁴³³ Vgl. Abschnitt 4.2.2.3.

⁴³⁴ STRAUCH verwendet anstelle des Begriffs Informationsobjekte den Begriff Berichtsposition, den er synonym mit dem Begriff Information verwendet. Vgl. *Strauch* (2002).

Für die Strukturierung bzw. Auflistung der einzelnen Informationsobjekte können bestimmte **Dimensionen** definiert werden.

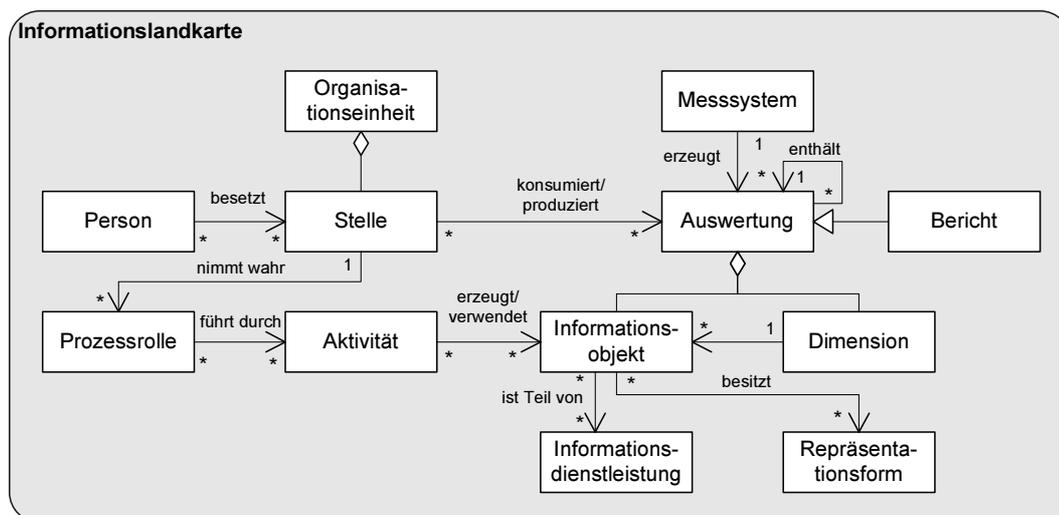


Abbildung 44: Metamodell der Informationslandkarte

Ein **Informationsobjekt** stellt eine Sammlung von Informationen dar, auf die bei der Durchführung von Aktivitäten wiederholt zurückgegriffen wird.⁴³⁵ Jedem Informationsobjekt können eine oder mehrere **Repräsentationsformen** zugeordnet werden.⁴³⁶ Diese lassen sich beispielsweise nach dem verwendeten Medium (z.B. Papier, elektronisch, Audio) oder dem verwendeten Format (z.B. PDF, HTML) unterscheiden. Informationsobjekte können Teil der Prozessleistung sein, wenn diese als Informationsdienstleistungen an nachfolgende Prozesse weitergegeben werden.⁴³⁷ Im Rahmen der Prozessführung umfassen Informationsobjekte Informationen über Prozessziele, kritische Erfolgsfaktoren oder Führungsgrößen. Auf der Organisationsebene können elektronische (durch Informationssystem unterstützte) und konventionelle Informationsobjekte unterschieden werden.⁴³⁸ Für die nachfolgende Gestaltung der Applikationsebene sind lediglich die elektronisch erfassbaren bzw. erfassten Informationsobjekte relevant. Diese werden auf Applikationsebene bestimmten Applikationen zugeordnet, die sie verwalten und/oder manipulieren.

Auf Basis der erfassten Ausprägungen der oben genannten Metaentitätstypen lassen sich anschliessend Auswertungen durchführen.⁴³⁹ So können beispielsweise Informationsobjekte und Organisationseinheiten bzw. Personen in Beziehung gesetzt werden. Dadurch wird für jedes im Unternehmen bzw. in der Geschäftseinheit zur Verfügung gestellte Informationsobjekt ersichtlich, welche Organisationseinheit bzw. Person dieses Informationsobjekt produziert

⁴³⁵ In Anlehnung an die Definition eines Informationsobjektes auf Applikationsebene. Vgl. *Schwinn* (2005), S. 138.

⁴³⁶ Vgl. *Jonkers et al.* (2004), S. 269.

⁴³⁷ Vgl. hierzu z.B. auch *Scheer* (2001), S. 70.

⁴³⁸ Vgl. hierzu *Frank* (1995), S. 5 und *Scheer* (2001), S. 69.

⁴³⁹ Vgl. *Strauch* (2002), S. 185-186.

bzw. verwendet. Zudem können potenzielle homonyme oder synonyme Informationsobjekte identifiziert werden, indem aufgezeigt wird, in welche Organisationseinheit die entsprechende Information geliefert wird. Sind es mehrere Organisationseinheiten, welche die Information nachfragen, liegt die Vermutung nahe, dass dieselbe Information unterschiedlich interpretiert wird. Die Auflistung potenzieller Homonyme und Synonyme unterstützt die spätere auf Applikationsebene erfolgende Erstellung des Informationsobjektmodells und die anschliessende Durchführung der Homogenisierung.

In Verbindung mit dem Modell der Ablaufplanung lässt sich darüber hinaus nachvollziehen, welche Organisationseinheiten oder Personen bei der Durchführung bestimmter Aktivitäten innerhalb der Geschäftsprozesse welche Informationsobjekte verwenden und/oder erzeugen. Dadurch wird ersichtlich, für die Erfüllung welcher Aktivität(en) die jeweilige Information wichtig ist und umgekehrt, welche Informationen für die Erfüllung einer bestimmten Aktivität benötigt werden. Folglich lässt sich sowohl der Informationsfluss zwischen Prozessen bzw. Aktivitäten als auch zwischen Organisationseinheiten bzw. Stellen/Personen auswerten. Da der Informationsfluss zwischen Prozessen und Aktivitäten bereits in der Prozesslandkarte und im Modell der Ablaufplanung abgebildet ist, steht bei der Informationslandkarte der Informationsfluss zwischen Organisationseinheiten und Stellen im Vordergrund.

4.2.2.7 Definition der Metaentitätstypen auf Organisationsebene

Die folgende Tabelle 10 spezifiziert die zuvor erwähnten Metaentitätstypen sowie deren Beziehungen im Detail.

Aktivität		
Definition	Eine Aktivität (synonym verwendeter Begriff für Aufgabe, Funktion, Vorgang) ist eine betriebliche Tätigkeit mit einem definierten Handlungsziel, die ein Objekt manipuliert und von einer Rolle bzw. einem Rollenträger durchgeführt wird. ⁴⁴⁰	
Beziehungen zu	Prozess	Eine Aktivität ist Bestandteil von einem oder mehreren Prozess/en.
	(Prozess-)Rolle	Eine Aktivität wird von keiner, einer oder mehreren Rolle/n durchgeführt.
	Informationsobjekt	Eine Aktivität manipuliert ein oder mehrere Informationsobjekt/e.
	Geschäftsfunktion	Eine Aktivität wird von einer oder mehreren Geschäftsfunktion/en durchgeführt.
	Applikation	Eine Aktivität wird von keiner, einer oder mehreren Applikation/en unterstützt.
	Ereignis	Eine Aktivität wird von einem oder mehreren Ereignis/sen ausgelöst.
Bericht		
Definition	Ein Bericht stellt einer oder mehreren Organisationseinheiten Informationen nach bestimmten Kriterien (Dimensionen) zur Verfügung. ⁴⁴¹ Ein Bericht der Prozessführung stellt das Ausmass und die Ursachen von Abweichungen zwischen den geplanten und den	

⁴⁴⁰ Vgl. Brecht (2002), S. 147, Österle (1995), S. 50 und Scheer (2001), S. 22.

⁴⁴¹ Vgl. Strauch (2002), S. 139.

	effektiven Ergebnissen eines Prozesses dar. Er gibt einen Überblick über die Entwicklung der Führungsgrößen. ⁴⁴²	
Beziehungen zu	Organisationseinheit	Ein Bericht geht an eine oder mehrere Organisationseinheit/en.
	Erfolgsfaktor	Ein Bericht enthält einen oder mehrere Erfolgsfaktor/en.
	Führungsgrösse	Ein Bericht enthält eine oder mehrere Führungsgrösse/n.
	Prozessziel	Ein Bericht umfasst ein oder mehrere Prozessziel/e.
	Stelle	Ein Bericht wird von einer oder mehreren Stelle/n konsumiert/produziert.
	Messsystem	Ein Bericht wird von einem Messsystem erzeugt.
	Informationsobjekt	Ein Bericht enthält ein oder mehrere Informationsobjekt/e.
Ereignis		
Definition	Ein Ereignis ist definiert als ein punktuell Geschehen, das einen Tatbestand enthält (Was) und zu einem Zeitpunkt stattfindet (Wann). ⁴⁴³ Eine Bedingung legt fest, unter welchen Voraussetzungen ein Ereignis relevant ist. Die Aktion beschreibt, wie auf das Eintreten eines Ereignisses bei Erfüllung der entsprechenden Bedingung reagiert werden soll.	
Beziehungen zu	Aktivität	Ein Ereignis wird von einer oder mehreren Aktivität/en ausgelöst.
	Operator	Ein Ereignis hat keinen, einen oder mehrere Operator/en als Vorgänger/Nachfolger.
Führungsgrösse		
Definition	Eine Führungsgrösse ist ein operationalisiertes Merkmal eines Prozesses. Führungsgrößen dienen der Planung und Beurteilung der Prozessqualität im Sinne der kritischen Erfolgsfaktoren, insbesondere der Zeit, Qualität, Kosten und Flexibilität. ⁴⁴⁴ Sie sind die Messgrößen für die kritischen Erfolgsfaktoren und stellen wichtige Indikatoren für die Effektivität und Effizienz eines Prozesses dar. ⁴⁴⁵	
Beziehungen zu	KEF	Eine Führungsgrösse operationalisiert einen oder mehrere KEF/en eines Prozesses.
	Prozessziel	Für eine Führungsgrösse werden ein oder mehrere Ziel/e definiert.
	Bericht	Eine Führungsgrösse ist Teil eines Berichts.
	Applikation	Eine Führungsgrösse wird von keiner oder einer Applikation gemessen.
	Prozessleistung	Eine Führungsgrösse bewertet eine oder mehrere Prozessleistung/en.
Führungsprozess		
Definition	Führungsprozesse gewährleisten die prozessübergreifende Planung, Steuerung und Kontrolle des Gesamtunternehmens. Ihre Leistungen gehen an die Leistungs- und Unterstützungsprozesse. Im Rahmen der Führungsprozesse werden Aufgaben wie die Kontrolle der Finanzen, die Entwicklung der Unternehmensstrategie sowie die Überwachung von deren Umsetzung ausgeführt.	
Geschäftsfunktion		
Definition	Eine Geschäftsfunktion ist eine Verarbeitungseinheit, die aus einer geschäftlichen (betriebswirtschaftlichen) Perspektive erhoben wurde. ⁴⁴⁶	

⁴⁴² Vgl. Brecht (2002), S. 149.

⁴⁴³ Vgl. Scheer (2001), S. 124.

⁴⁴⁴ Vgl. Österle (1995), S. 54.

⁴⁴⁵ Vgl. Österle (1995), S. 116 und Glossar in IMG (1997).

⁴⁴⁶ Vgl. Schwinn (2005), S. 19 und Gutzwiller (1994), S. 81.

Beziehungen zu	Aktivität	Eine Geschäftsfunktion führt eine Aktivität aus.
	Applikationsfunktion	Eine Geschäftsfunktion kann durch eine oder mehrere Applikationsfunktion/en auf Applikationsebene realisiert werden.
Informationsobjekt		
Definition	Ein Informationsobjekt der Organisationsebene stellt eine Sammlung von Informationen dar, auf die bei der Durchführung von Aktivitäten wiederholt zurückgegriffen wird. ⁴⁴⁷ Es können elektronische und konventionelle Informationsobjekte differenziert werden. ⁴⁴⁸	
Beziehungen zu	Aktivität	Ein Informationsobjekt wird von einer oder mehreren Aktivität/en manipuliert.
	Bericht	Ein Informationsobjekt kann Teil eines Berichtes sein.
	Informationsdienstleistung	Ein Informationsobjekt kann Teil einer oder mehrerer Informationsdienstleistung/en sein.
Kritischer Erfolgsfaktor (KEF)		
Definition	Ein kritischer Erfolgsfaktor bezeichnet eines der wenigen Merkmale, die den Erfolg eines Prozesses ausmachen. Die kritischen Erfolgsfaktoren helfen der Prozessführung, sich auf das Wesentliche zu konzentrieren. ⁴⁴⁹	
Beziehungen zu	Führungsgröße	Ein KEF wird durch keine, eine oder mehrere Führungsgröße/n operationalisiert.
	Prozess	Ein KEF ist für einen oder mehrere Prozess/e erfolgsentscheidend.
Leistungsprozess		
Definition	Gegenstand von Leistungsprozessen ist die Erstellung und der Vertrieb von Produkten und Dienstleistungen. ⁴⁵⁰ Mittels dieses Prozessstyps besitzt das Unternehmen Kontakt zum Kunden, da die Prozessleistung direkt an diesen abgegeben wird.	
Messsystem		
Definition	Ein Messsystem dient der informationstechnischen Unterstützung der Prozessführung. Ein Messsystem erhebt den Ist-Zustand eines Prozesses, indem es die im Rahmen der Prozessführung definierten Führungsgrößen misst und Berichte erstellt, die die Messergebnisse zusammenfassen. ⁴⁵¹	
Beziehungen zu	Applikation	Ein Messsystem wird durch eine Applikation auf Applikationsebene realisiert.
	Bericht	Ein Messsystem erzeugt einen oder mehrere Bericht/e.
	Prozess	Ein Messsystem ist einem oder mehreren Prozess/en zugeordnet.
	Führungsgröße	Ein Messsystem misst keine, eine oder mehrere Führungsgröße/n.
Operator		
Definition	Ein Operator stellt eine logische Verknüpfung zwischen Ereignissen und Aktivitäten dar. Typische logische Verknüpfungen sind UND, ODER (Inklusiv-Oder) und XOR (Exklusiv-Oder). ⁴⁵²	
Beziehungen zu	Aktivität	Ein Operator ist der Vorgänger/Nachfolger von einer oder mehreren Aktivität/en.
	Ereignis	Ein Operator ist der Vorgänger/Nachfolger von einem oder mehreren Ereignis/sen.

⁴⁴⁷ In Anlehnung an die Definition eines Informationsobjektes auf Applikationsebene. Vgl. *Schwinn* (2005), S. 138.

⁴⁴⁸ Vgl. *Scheer* (2001), S. 69.

⁴⁴⁹ Vgl. *Brecht* (2002), S. 159.

⁴⁵⁰ Vgl. *Österle* (1995), S. 130.

⁴⁵¹ Vgl. hierzu auch *Brecht* (2002), S. 162.

⁴⁵² Vgl. *Scheer* (2001), S. 125.

Organisationseinheit des Prozessmanagements (OE des PM)		
Definition	Eine organisatorische Einheit des Prozessmanagements ist eine speziell für die Führung eines Prozesses eingerichtete Organisationseinheit. ⁴⁵³	
Beziehungen zu	Organisationseinheit	Eine OE des PM ist eine Organisationseinheit.
	Prozess	Eine OE des PM ist für das Management genau eines Prozesses verantwortlich.
	Prozessgrundsatz	Eine OE des PM pflegt einen Prozessgrundsatz oder mehrere.
Prozess		
Definition	Der Prozess wird als eine zusammengehörende Abfolge von Unternehmensverrichtungen zum Zweck einer Leistungserstellung definiert. Input bzw. Output eines Prozesses sind seine Leistungen, die von internen oder externen Kunden angefordert und abgenommen werden. ⁴⁵⁴	
Beziehungen zu	Prozessleistung	Ein Prozess erbringt oder bezieht keine, eine oder mehrere Prozessleistung/en.
	Organisationseinheit	Ein Prozess ist einer oder mehreren Organisationseinheit/en (des PM) zugeordnet.
	Serviceaktivität	Ein Prozess korrespondiert mit keiner oder einer Serviceaktivität auf Strategieebene.
	KEF	Für einen Prozess sind ein oder mehrere kritische Erfolgsfaktor/en erfolgsentscheidend.
	Applikationsdomäne	Ein Geschäftsprozess kann durch eine oder mehrere Applikationsdomäne/n auf Applikationsebene unterstützt werden.
	Aktivität	Ein Geschäftsprozess beinhaltet eine oder mehrere Aktivität/en.
	Applikation	Einem Prozess ist keine oder eine Applikation als Messsystem zugeordnet.
Prozessgrundsatz		
Definition	Ein Prozessgrundsatz beschreibt ein grundlegendes, langfristig gültiges Prinzip bei der Weiterentwicklung eines Prozesses. ⁴⁵⁵ Ein Prozessgrundsatz bezieht sich beispielsweise auf die von einem Prozess erzeugte Leistung, seinen Ablauf, seine Erfolgsfaktoren oder seine IT-Unterstützung.	
Beziehungen zu	Prozess	Ein Prozessgrundsatz beschreibt die Gestaltung eines Prozesses oder mehrerer Prozesse.
	Organisationseinheit	Ein Prozessgrundsatz wird von einer verantwortlichen OE gepflegt.
Prozessleistung		
Definition	Prozessleistungen sind das Ergebnis (der Output) der Leistungserstellung eines Prozesses, das an interne oder externe Kunden geht. Empfänger einer Leistung ist ein anderer Prozess innerhalb oder ausserhalb der Unternehmung. Eine Prozessleistung kann materiell oder immateriell sein. Ist der Leistungsempfänger ein Prozess ausserhalb der eigenen Unternehmung, ist eine Prozessleistung Teil einer Marktleistung. ⁴⁵⁶	
Beziehungen zu	Prozess	Eine Prozessleistung wird von einem oder mehreren Prozess/en erzeugt und von einem oder mehreren Prozess/en entgegengenommen.
	Marktleistung	Eine Prozessleistung ist Teil keiner, einer oder mehrerer Marktleistung/en auf Strategieebene.

⁴⁵³ Vgl. Glossar in *IMG* (1997).

⁴⁵⁴ Vgl. *Scheer* (1998), S. 3.

⁴⁵⁵ Vgl. *Brecht* (2002), S. 168.

⁴⁵⁶ Vgl. *Brecht* (2002), S. 170.

	Informationsobjekt	Eine Prozessleistung kann als Informationsdienstleistung ein oder mehrere Informationsobjekt/e umfassen.
	Führungsgrösse	Eine Prozessleistung wird durch eine oder mehrere Führungsgrösse/n bewertet.
Prozessrolle		
Definition	Eine Prozessrolle repräsentiert zum einen die für die Ausführung einer Aktivität notwendige minimale Qualifikation (z. B. „spricht Spanisch“), zum anderen beschreibt eine Rolle Kompetenzen, welche einem Rollenträger übertragen werden (z. B. „ist zeichnungsberechtigt“). ⁴⁵⁷ Die Prozessrolle dient als Bindeglied zwischen der Aufbau- und Ablauforganisation.	
Beziehungen zu	Aktivität	Eine Rolle führt eine oder mehrere Aktivität/en innerhalb eines Prozesses aus.
	Anforderung	Eine Prozessrolle definiert eine oder mehrere Anforderung/en.
	Rollenträger	Eine Prozessrolle wird von einem Rollenträger wahrgenommen.
Prozessziel		
Definition	Das Prozessziel definiert den Wert einer Führungsgrösse, der für einen bestimmten Zeitpunkt angestrebt wird. ⁴⁵⁸	
Beziehung zu	Führungsgrösse	Ein Prozessziel bestimmt eine oder mehrere Führungsgrösse/n.
Rollenträger		
Definition	Organisationseinheiten bzw. die ihnen zugeordneten Mitarbeiter (organisatorische Rollenträger) und/oder Maschinen (technische Rollenträger) führen im Rahmen einer Prozessrolle Aktivitäten aus. ⁴⁵⁹	
Beziehungen zu	Rolle	Ein Rollenträger nimmt eine oder mehrere (Prozess-)Rolle/n wahr.
	Benutzerkonto	Ein Rollenträger verfügt über kein/ein/mehrere Konto/Konten auf Systemebene.
Stelle		
Definition	Eine Stelle bezeichnet eine Zusammenfassung von Aufgaben, die eine derartige Kapazitätsnachfrage bilden, dass sie einer Person übertragen werden können und diese dauerhaft bei definierter, im Regelfall kontinuierlicher Arbeitszeit auslasten. ⁴⁶⁰	
Beziehungen zu	Person	Eine Stelle kann durch eine oder mehrere Person/en besetzt werden, z.B. aufgrund von Schichtarbeitsplätzen oder Teilzeit.
	Bericht	Eine Stelle konsumiert/produziert keinen, einen oder mehrere Bericht/e.
Stellentyp		
Definition	Ein Stellentyp fasst Stellen mit gleichen Kompetenzen zusammen (z. B. Stellentyp „Sekretärin“ im Gegensatz zur Stelle „Sekretärin von Prof. Winter“). ⁴⁶¹	
Beziehungen zu	Stelle	Ein Stellentyp fasst eine oder mehrere Stelle/n zusammen.
Stellvertretung		
Definition	Eine Stellenstellvertretung stellt einen begrenzten oder unbegrenzten Einsatz eines oder mehrerer Stelleninhaber (Person) für einen anderen Stelleninhaber dar, sowohl in räumlicher, funktionsmässiger als auch in zeitlicher Hinsicht. ⁴⁶² Analog lässt sich eine Rollenstellvertretung für einen Rollenträger definieren.	
Beziehungen zu	Person	Eine Stellvertretung ist einer oder mehreren Person/en zugeordnet.
	Stelle	Eine Stellvertretung bezieht sich auf eine oder mehrere Stelle/n.

⁴⁵⁷ Vgl. Rosemann/zur Mühlen (1997), S. 79.

⁴⁵⁸ Vgl. Brecht (2002), S. 178.

⁴⁵⁹ Vgl. Scheer (2001), S. 56 und Metamodell in IMG (1997).

⁴⁶⁰ Vgl. Rosemann/zur Mühlen (1997), S. 80.

⁴⁶¹ Vgl. Rosemann/zur Mühlen (1997), S. 80.

⁴⁶² Vgl. Wortmann (2005), S. 122.

Unterstützungsprozess	
Definition	Unterstützungsprozesse ermöglichen die kontinuierliche Ausführung mehrerer Leistungsprozesse, insbesondere durch die Bereitstellung von Ressourcen (Personal, Material und Hilfsstoffe), die Bereitstellung und Pflege der Infrastruktur und die (Weiter-) Entwicklung der Produkte und Dienstleistungen. ⁴⁶³

Tabelle 10: Metaentitätstypen der Organisationsebene

4.2.2.8 Festlegung der verwendeten Symbole auf Organisationsebene

Die folgende Tabelle 11 gibt einen Überblick über die Notationselemente (Symbole) für die auf Organisationsebene definierten Metaentitäts- und Beziehungstypen. Metaentitätstypen ohne zugeordnetes Notationselement werden lediglich textuell (z.B. in Form von Listen) repräsentiert.

Symbole der Organisationsebene		
Atomare Aktivität	Ereignis	Führungsgröße
Führungsprozess	Geschäftsfunktion	Informationsobjekt
Komplexe Aktivität	Kritischer Erfolgsfaktor	Leistungsprozess
Operator	Organisationseinheit	Prozessleistung
Prozessrolle	Prozessstart	Prozesswegweiser

⁴⁶³ Vgl. Glossar in IMG (1997).

Prozessziel	Stelle	Unterstützungsprozess
 Prozessziel	 Stelle	 Unterstützungsprozess
erbringt/bezieht Information (von Prozess zu Prozess)	bestimmt (von Prozessziel zu Führungsgrösse)	erbringt/bezieht Leistung (von Prozess zu Prozess)
		
führt aus (von Rollenträger zu Aktivität)	ist Input für (von IO zu Aktivität)	ist Output von (von IO zu Aktivität)
		
unterstützt (von Applikation zu Aktivität)	ist zugeordnet (von Führungsgrösse zu Aktivität)	ist übergeordnet (von OE zu OE)
		
nimmt wahr (von Rollenträger zu Rolle)	gehört zu (von Stelle zu OE)	kommuniziert mit (von OE zu OE)
		
ist Teil von (von Leistung zu Leistung)	ist Nachfolger/Vorgänger von (von Ereignis zu Aktivität/Operator)	ist Stellvertretung für (von Person zu Stelle)
		

Tabelle 11: Symbole der Organisationsebene

Bei Beziehungstypen, denen kein Notationselement zugeordnet ist, handelt es sich entweder um eine modellübergreifende Beziehung oder um eine Aggregationsbeziehung. Aggregationsbeziehungen werden in Form einer Schachtelung von Symbolen (z.B. bei der Beziehung zwischen „Erfolgsfaktor“ und „Führungsgrösse“) dargestellt. Modellübergreifende Beziehungen werden später bei der softwaretechnischen Implementierung durch Referenzen abgebildet (vgl. hierzu Abschnitt 5.2.4).

4.3 Modellierung auf Informationssystemebene

Die Systemebene ist die der Organisationsebene logisch nachfolgende Gestaltungsebene (vgl. Abschnitt 2.3.2). Sie wird weiter unterteilt in die Applikationsebene sowie Softwarekomponenten- und Datenstrukturebene.

Auf der Informationssystemebene werden Integrationsbereiche innerhalb des Informationssystems festgelegt, ausgehend von den zuvor spezifizierten Organisationsstrukturen, Informationsbedarfen und Prozessabläufen. Im Normalfall wird ein eng gekoppelter Integrationsbereich durch eine Applikation realisiert, während lose Kopplungen zwischen Integrationsbereichen durch Schnittstellen zwischen Applikationen realisiert werden. Wichtige Ergebnisse der

Applikationsgestaltung sind gesamthaft die Integrationsarchitektur und auf detaillierter Ebene Fachkonzepte für einzelne Applikationen bzw. Informationssystem-Komponenten.

Wesentliche Anforderungen an einen Ansatz für die Modellierung der Systemebene sind:⁴⁶⁴

- Unterstützung bei der Ableitung einer effektiven und effizienten IT-Unterstützung für die Prozessebene,
- Unterstützung bei der optimalen Strukturierung von Applikationen und ihrer Integration unter Vermeidung ungewollter Redundanzen oder Lücken.

Auf der Systemebene können folgende Ergebnisdokumententypen unterschieden werden:

- Mit Hilfe des **Informationsobjekte- bzw. Geschäftsfunktionenmodells** werden die in Bezug auf die Prozesse relevanten Informationsobjekte (z.B. Kunde, Partner, Produkt) und Geschäftsfunktionen (z.B. Kunde aufnehmen) ermittelt, miteinander verglichen und in einem Homogenisierungsschritt hinsichtlich Synonymen und Homonymen vereinheitlicht und dokumentiert.⁴⁶⁵ Sie dienen als Basis für die Ableitung von neuen Applikationen im Rahmen der Definition der Integrationsarchitektur.
- Die **Applikationslandschaft** (Applikationsbestandsführung) dokumentiert den Gesamtbestand an Applikationen, die sich in einem Unternehmen bzw. in einer Geschäftseinheit in Betrieb oder in Planung befinden.⁴⁶⁶
- Die **Integrationsarchitektur** beschreibt eine applikatorische Beziehungsstruktur. Diese wird aus den Beziehungselementen Informationsobjekte, Geschäftsfunktionen, Prozesse bzw. Organisationseinheiten und deren unterschiedlichen Beziehungen untereinander hergeleitet.⁴⁶⁷ Den daraus abgeleiteten Integrationsbereichen können neu zu entwickelnde Applikationen zugeordnet werden.
- Das **Softwarekomponenten- und Plattformenmodell** dokumentiert die einzelnen Komponenten ausgewählter Applikationen sowie die von ihnen manipulierten Datenobjekte und die Plattformen, auf denen sie betrieben werden. Zudem werden wichtige Schnittstellen und Datenspeicher abgebildet. Die Abbildung der Softwarekomponenten, Plattformen, Schnittstellen und Datenstrukturen stellt vor allem im Hinblick auf zukünftige Applikationsintegrationsprojekte eine wichtige Grundlage dar.⁴⁶⁸

⁴⁶⁴ Vgl. *Winter* (2004c), *Kühn/Karagiannis* (2005) sowie Abschnitt 2.3.1 und 2.3.2.

⁴⁶⁵ Vgl. *Choinowski et al.* (2003), S. 76.

⁴⁶⁶ Vgl. *Choinowski et al.* (2003), S. 72.

⁴⁶⁷ Der Ansatz geht auf WINTER zurück, nach welchem die beiden Methoden der Applikationsarchitekturmodellierung „Business System Planning“ (vgl. IBM 1984) und Promet@STP (System- und Technologieplanung) (vgl. IMG 2000) zu einer integrierten Verfahrensweise kombiniert wurden. Vgl. *Winter* (2003a).

⁴⁶⁸ Vgl. hierzu z.B. die von SCHWINN entwickelte Methode für Applikationsintegrationsprojekte *Schwinn* (2005).

- Das **Autorisierungsmodell** ordnet den menschlichen Rollenträgern für die Durchführung der informationstechnisch unterstützten Aufgaben entsprechende Systemberechtigungen zu. Diese Berechtigungen regeln den Zugriff auf die Methoden und Datenobjekte der existierenden Softwarekomponenten und werden über Benutzerkonten den Rollenträgern zugeordnet.⁴⁶⁹
- Das **Datenmodell** bildet die detaillierte Datenstruktur eines fachlichen Anwendungsbereichs ab.⁴⁷⁰ Als Grundlage für die Erstellung des Datenmodells dienen die auf Applikations- sowie Softwarekomponenten- und Datenstrukturebene identifizierten Informationsobjekte bzw. Datenobjekte. Das noch aus fachkonzeptioneller Sicht entworfene Datenmodell stellt eine hilfreiche Grundlage für die objektorientierte Softwareentwicklung dar.

Im folgenden Abschnitt wird das Vorgehen zur Erstellung der zuvor genannten Ergebnisdokumente beschrieben.

4.3.1 Vorgehensmodell

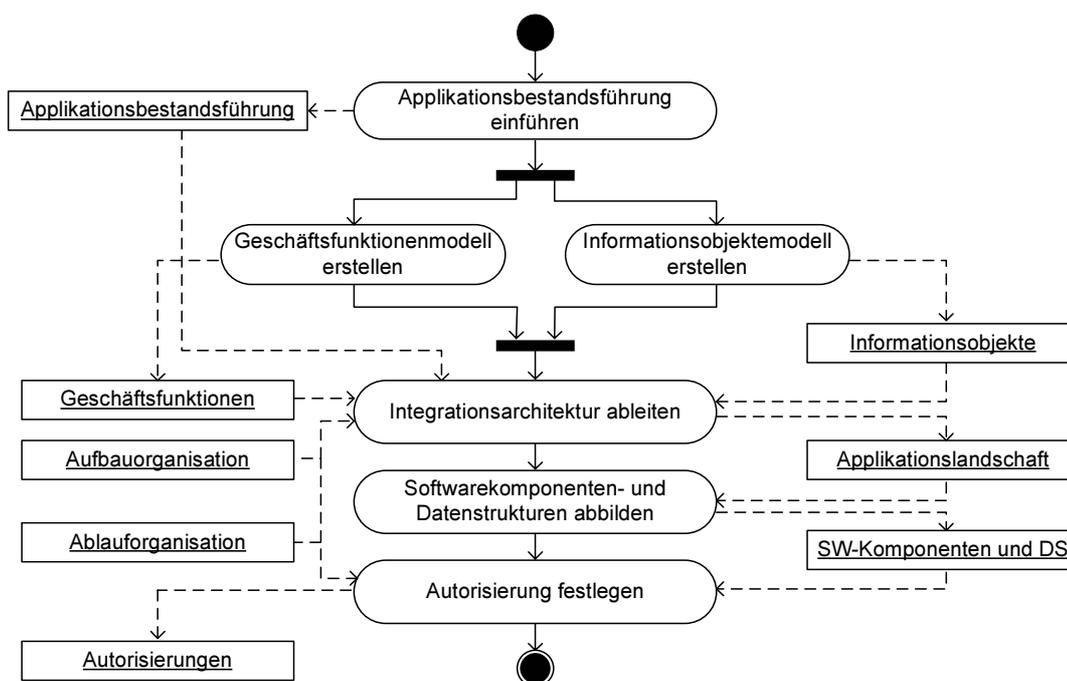


Abbildung 45: Vorgehensmodell auf Informationssystemebene (Level 1)

Als erstes wird eine *Applikationsbestandsführung eingeführt*, da aus ihr die Integrationsarchitektur abgeleitet wird.⁴⁷¹ Die Applikationsbestandsführung soll den aktuellen Gesamtbestand an geplanten und realisierten Applikationen anhand vordefinierter Attribute dokumentieren.

⁴⁶⁹ Vgl. hierzu z.B. die von Wortmann entwickelte Methode für die unternehmensweite Autorisierung Wortmann (2005).

⁴⁷⁰ SCHEER definiert beispielsweise eine Mikrosicht auf die Datenobjekte, die mit Hilfe eines erweiterten ERM abgebildet wird (vgl. Scheer (2001), S. 70).

⁴⁷¹ Vgl. Choinowski et al. (2003), S. 69.

Das Vorgehensmodell zur Erstellung der Applikationsbestandsführung besteht aus vier Schritten:⁴⁷² Zuerst wird das Modell der Bestandsführung in Absprache mit den unterschiedlichen Fachbereichen und deren Informationsbedarf erstellt und im zweiten Schritt verabschiedet. Im dritten Schritt wird ein entsprechendes Modell für die Bestandsführung in einer Datenbank angelegt, auf die über das Intranet zugegriffen werden kann. Im vierten Schritt werden die notwendigen Verantwortlichkeiten und involvierten Organisationseinheiten festgelegt. Im letzten Schritt folgt die unternehmensinterne Implementierung.

Nach der Einführung der Applikationsbestandsführung werden parallel ein *Informationsobjekte- und Geschäftsfunktionenmodell* erstellt. Mit Hilfe des Informationsobjektmodells werden die auf Organisationsebene nach unterschiedlichen Konventionen abgebildeten Prozesse hinsichtlich der verwendeten Begriffe für Informationsobjekte harmonisiert. Es werden die in Bezug auf die Prozesse relevanten Informationsobjekte, wie z.B. Kunde, Partner, Produkt, ermittelt, miteinander verglichen und in einem Homogenisierungsschritt hinsichtlich Synonymen und Homonymen vereinheitlicht und in einem Modell dokumentiert. Die Ableitung der Informationsobjekte für das unternehmens- bzw. geschäftsbereichsweite Informationsobjektmodell erfolgt anhand von zwei Quellen: erstens aus den Ergebnisdokumenten der Organisationsebene und zweitens aus der Applikationsbestandsführung. Das Vorgehensmodell besteht aus drei Schritten:⁴⁷³ Zuerst werden die Informationsobjekte anhand der Dokumentation der Prozessabläufe und der Organisationsstruktur ermittelt, harmonisiert und in einem Informationsobjektmodell dokumentiert. Dabei werden lediglich die als elektronisch bzw. informationstechnisch-unterstützt klassifizierten Informationsobjekte betrachtet. Danach erfolgt die Synchronisation mit den Informationsobjekten der Applikationsbestandsführung und als drittes die entsprechende Dokumentation mit Weisung an die Verantwortlichen der Applikationsbestandsführung, die entsprechenden Anpassungen in der Ist-Dokumentation vorzunehmen. Als Ergebnis liegt ein harmonisiertes, unternehmensweites Informationsobjektmodell mit den entsprechenden Begriffen, Definitionen und Hinweisen auf Synonyme und Homonyme vor.

Ebenso wie die Informationsobjekte werden auch die Geschäftsfunktionen ermittelt und in einem Geschäftsfunktionenmodell harmonisiert.

Aus der Applikationsbestandsführung (Applikationslandschaft) sowie dem Geschäftsfunktionen- und Informationsobjektmodell wird die *Integrationsarchitektur abgeleitet*. Diese besteht aus einer Ist- und Soll-Integrationsarchitektur. Beide werden mittels eines doppelten Clusteranalyseverfahrens ermittelt, welches pro Verfahrensindikation dieselben Variablen über unterschiedliche Grundmengen verwendet. Der Ansatz geht auf WINTER zurück, nach welchem die beiden Methoden der Applikationsarchitekturmodellierung „Business System

⁴⁷² Vgl. Choinowski et al. (2003), S. 73.

⁴⁷³ Vgl. Choinowski et al. (2003), S. 76.

Planning“ (BSP)⁴⁷⁴ und Promet® STP (System- und Technologieplanung)⁴⁷⁵ zu einer integrierten Verfahrensweise kombiniert wurden.⁴⁷⁶ Als Variablen werden Informationsobjekt, Geschäftsfunktion und Organisationseinheit verwendet, die im doppelten Matrixverfahren mit vordefinierten Werten einander gegenübergestellt und durch die Bündelung gleicher Wertebezüge zu Integrationsbereichen ausgewertet werden. Für die Ist-Integrationsarchitektur wird als Grundmenge die Gesamtmenge der im Informationsobjektmodell, Geschäftsfunktionsmodell und Aufbauorganisationsmodell dokumentierten Entitäten genommen. Für die Soll-Integrationsarchitektur gilt als Grundmenge die entlang der neuen Soll-Prozessabläufe ermittelte Menge an neuen Informationsobjekten, Geschäftsfunktionen und Organisationseinheiten. Die ermittelten Integrationsbereiche über Ist und Soll werden danach miteinander verglichen und ausgewertet, sowohl hinsichtlich eines Redundanz- und Lückennachweises an bestehender Applikationsunterstützung als auch hinsichtlich der Gestaltungsvorgaben aus den Gestaltungsgrundsätzen der Prozessebene. Eine Priorisierung der Integrationsbereiche hinsichtlich der Geschäftsanforderungen wird während dieser Entwurfsphase noch nicht durchgeführt, sondern erfolgt später während der IT-Strategiefestlegung unter dem Punkt „Soll-Ist-Konzeption definieren“. Als Ergebnisdokument liegen die aufeinander abgestimmten Integrationsbereiche aus Ist und Soll auf Mikroebene vor, inklusive eines Lücken- und Redundanznachweises hinsichtlich bestehender Applikationsunterstützung. Integration auf Mikroebene bedeutet, dass eine optimale und aufeinander abgestimmte Integration der Daten-, Funktions- und Organisationseinheiten hergeleitet worden ist.

Im nächsten Schritt werden die für die Implementierung ausgewählter Applikationen benötigten *Implementierungsbausteine abgebildet*, wie beispielsweise Softwarekomponenten, Middlewarekomponenten, Schnittstellen sowie Plattformen, auf denen die Softwarekomponenten installiert sind.⁴⁷⁷

Ausgangsbasis zur *Erstellung des Datenstrukturmodells* bilden die im Informationsobjektmodell sowie im Modell der Softwarekomponenten identifizierten Informations- bzw. Datenobjekte. Die detaillierte Datenstruktur, d.h. die Datenobjekte sowie deren Beziehungen, wird mit Hilfe einer objektorientierten Entwurfsmethode (z.B. UML⁴⁷⁸ und ERM⁴⁷⁹) in einem oder mehreren Klassendiagramm/en abgebildet. Diese können später als Vorlage für die softwaretechnische Implementierung verwendet werden.

⁴⁷⁴ Vgl. *IBM-Corporation* (1984).

⁴⁷⁵ Vgl. *IMG* (1999).

⁴⁷⁶ Vgl. *Winter* (2003a).

⁴⁷⁷ Für eine detaillierte Beschreibung des Vorgehens zur Abbildung der Implementierungsbausteine, insbesondere im Hinblick auf zukünftige Applikationsintegrationsprojekte, vgl. *Schwinn* (2005).

⁴⁷⁸ Vgl. *OMG* (2005).

⁴⁷⁹ Vgl. *Chen* (1976).

Basierend auf den Modellen der Aufbau- und Ablauforganisation der Organisationsebene sowie dem Modell der Softwarekomponenten und Datenstrukturen kann in einem letzten Schritt die *Autorisierung festgelegt* werden. Diese verwaltet und kontrolliert die Zugriffsrechte der einzelnen Rollenträger auf die Softwarekomponenten und Datenobjekte und stellt somit ein wichtiges Konzept für die Gewährleistung der Sicherheit eines Informationssystems dar. Für die Festlegung der Autorisierung muss jedem Rollenträger ein Benutzerkonto zugeordnet werden, das entsprechende Berechtigungen umfasst, die den Zugriff auf bestimmte Softwarekomponenten und Datenobjekte definieren.⁴⁸⁰

4.3.2 Metamodell

Das Metamodell der Systemebene lässt sich entsprechend den bei der Systemgestaltung erstellten Ergebnisdokumententypen in die Sichten Informationsobjekte und Geschäftsfunktionen, Applikationsbestandsführung (Applikationslandschaft), Integrationsarchitektur, Softwarekomponenten und Plattformen, Autorisierung sowie Datenstrukturen einteilen.

4.3.2.1 Metamodell der Applikationslandschaft

Die Applikationslandschaft (Bestandsführung) enthält den aktuellen Gesamtbestand an geplanten und realisierten Applikationen und dokumentiert diese anhand vordefinierter Attribute. Für die Ist-Integrationsarchitektur sind die Informationen zu Informationsobjekten, Geschäftsfunktionen und zu den involvierten Organisationseinheiten von Bedeutung und müssen pro Applikation erfasst werden. Sie stellen die Vergleichsbasis mit der Soll-Integrationsarchitektur dar, welche die entsprechenden Informationsobjekte, Geschäftsfunktionen und involvierten Organisationseinheiten aus der Prozessmodellierung ableitet. Zusätzliche Informationen können ebenso in der Applikationsbestandsführung erfasst werden, wie z.B. Informationen über den Technologielebenszyklus einer Applikation, die in Bezug auf zu tätige Investitionen von Relevanz sein können. Die Konzeptualisierung der Applikationsbestandsführung fällt unternehmensindividuell aus, je nach erwünschtem Informationsgrad. Im Laufe der Zeit kann sie um Attribute erweitert werden, da es sich um eine laufende Dokumentation handelt. In dem hier präsentierten Ansatz werden für die Applikationsbestandsführung lediglich die Angaben zu Informationsobjekten, Geschäftsfunktionen und den involvierten Organisationseinheiten vorausgesetzt, weiterführende Attribute sind unternehmensindividuell zu definieren.

Eine **Applikation** besteht aus Applikationsfunktionen und deren Operationen auf Informationsobjekten. Applikationen können zur Beherrschung der Komplexität in einer **Applikationsdomäne** logisch zusammengefasst werden. Diese kann auf Organisationsebene einen oder mehrere Prozesse unterstützen. Jeder Applikation ist ein **Owner** zugeordnet, der aus einer

⁴⁸⁰ Für eine detaillierte Beschreibung des Vorgehens zur Festlegung der Autorisierung vgl. *Wortmann (2005)*.

bestimmten Organisationseinheit stammt und für die Applikation und deren Dokumentation verantwortlich ist. Ein Owner kann eine Stelle, eine Person oder eine Organisationseinheit sein.

Eine **Applikationsfunktion** realisiert eine oder mehrere Geschäftsfunktionen mit Hilfe von Informationstechnologie.⁴⁸¹ Sie kann als Input Informationsobjekte erhalten und/oder als Output neue oder veränderte Informationsobjekte erzeugen.

Ein **Informationsobjekt** stellt eine Sammlung dauerhaft gespeicherter Informationen dar, auf die (von Applikationsfunktionen) wiederholt zurückgegriffen wird. Es ist einer verwaltenden und einer oder mehreren operierenden Applikation(en) zugeordnet und wird auf der Ebene der Softwarekomponenten und Datenstrukturen als **Datenobjekt** repräsentiert.⁴⁸² Jedes Datenobjekt ist letztendlich einem bestimmten **Datenspeicher** zugeordnet. SCHWINN definiert zusätzlich den Metaentitätstyp „Datenspeicherobjekt“, welcher eine Instanz eines Datenobjekts in dem Datenspeicher repräsentiert. Im Sinne der Unternehmensarchitektur und im Hinblick auf einen angemessenen Detaillierungsgrad wird auf die Verwendung dieses Metaentitätstyps in diesem Ansatz verzichtet.

Abbildung 46 zeigt das Metamodell der Applikationslandschaft mit den zuvor definierten Metaentitätstypen sowie deren Beziehungen.

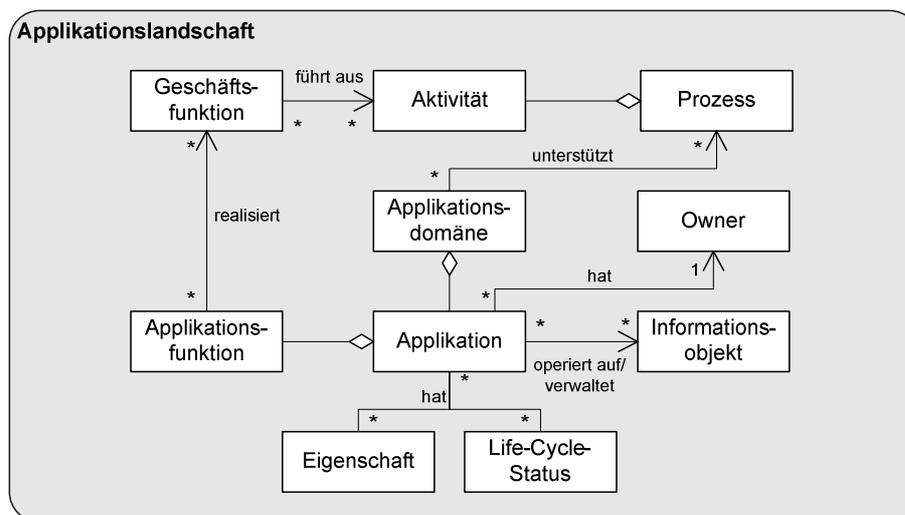


Abbildung 46: Metamodell der Applikationslandschaft

Dank der durchgängigen Dokumentation der Applikationen bzw. Applikationsdomänen und der unterstützten Geschäftsprozesse können Aussagen über betroffene Applikationen bei Veränderung oder Wegfall vorhandener Prozesse getroffen werden. Typische Fragestellungen wären dabei „Welche Applikationen sind bei der Veränderung eines Prozesses betroffen?“

⁴⁸¹ Vgl. Schwinn (2005), S. 20.

⁴⁸² Vgl. Schwinn (2005), S. 138.

oder „Welche Applikationen (oder Teile davon) können aufgrund weggefallener Prozesse stillgelegt werden?“.

4.3.2.2 Metamodell des Geschäftsfunktions- und Informationsobjektmodells

Jedes Informationsobjekt bzw. jede Geschäftsfunktion verfügt über eine **Begrifflichkeit**, die hinsichtlich eines festzulegenden **Standard-Begriffs** und seiner **Synonyme** und **Homonyme** mit Hilfe von **Gestaltungskriterien** harmonisiert wird (vgl. Abbildung 47). Als **Erhebungsquelle** kommen Prozesse, Organisationseinheiten und die Applikationsbestandsführung in Frage. Die Applikationsbestandsführung verfügt über bereits begrifflich standardisierte Informationsobjekte und Geschäftsfunktionen, die anderen Erhebungsquellen jedoch nicht. Die Informationsobjekte bzw. Geschäftsfunktionen und ihre Beschreibung werden in einem Geschäftsfunktions- bzw. Informationsobjektmodell dokumentiert. Für das Modell und den Harmonisierungsprozess wird ein **Owner** definiert, welcher die Verantwortung für das Informationsobjekt- bzw. Geschäftsfunktionsmodell übernimmt. Die (unternehmensweit) standardisierten Informationsobjekte sind Bezugsgröße und Voraussetzung für die Erstellung der Integrationsarchitektur sowie für zukünftige Projekte der Applikationsintegration. Die Dokumentation der Informationsobjekte und Geschäftsfunktionen ermöglicht zukünftig die einfachere Durchführung von Analysen, die für Integrationsfragestellungen relevant sind.⁴⁸³ Beispielsweise können bei Veränderungen von Informationsobjekten oder Geschäftsfunktionen (z. B. ein zusätzliches Attribut im Informationsobjekt „Kunde“) Rückschlüsse auf betroffene Applikationen gezogen werden. Daneben können Redundanzen leichter aufgedeckt werden.

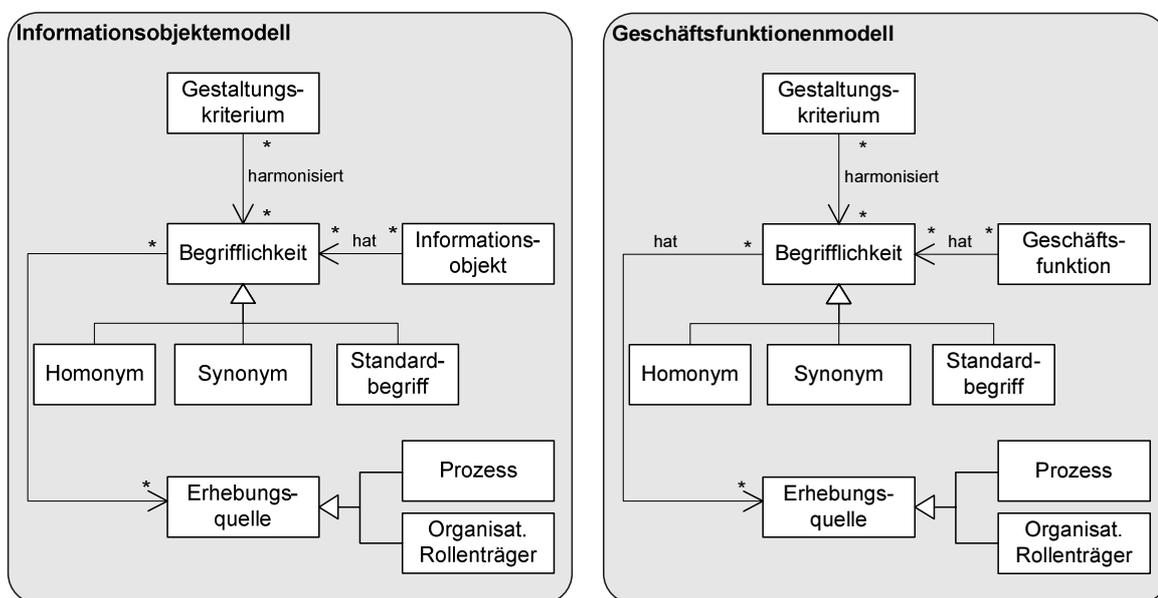


Abbildung 47: Metamodell des Geschäftsfunktions- und Informationsobjektmodells

⁴⁸³ Vgl. hierzu Schwinn (2005), S. 199.

4.3.2.3 Metamodell der Integrationsarchitektur

Die Integrationsarchitektur dokumentiert die Beziehungsstrukturen zwischen den Entitäten Informationsobjekt, Geschäftsfunktion und Organisationseinheit der Reihe nach in Bezug auf die unterschiedlichen Beziehungsarten wie beispielsweise Informationsfluss, Datenfluss, Leistungsfluss und Controllingfluss, um die relevanten Integrationsbereiche hervorzuheben. Diesen Integrationsbereichen können bestehende oder neu zu entwickelnde Applikationen zugeordnet werden.

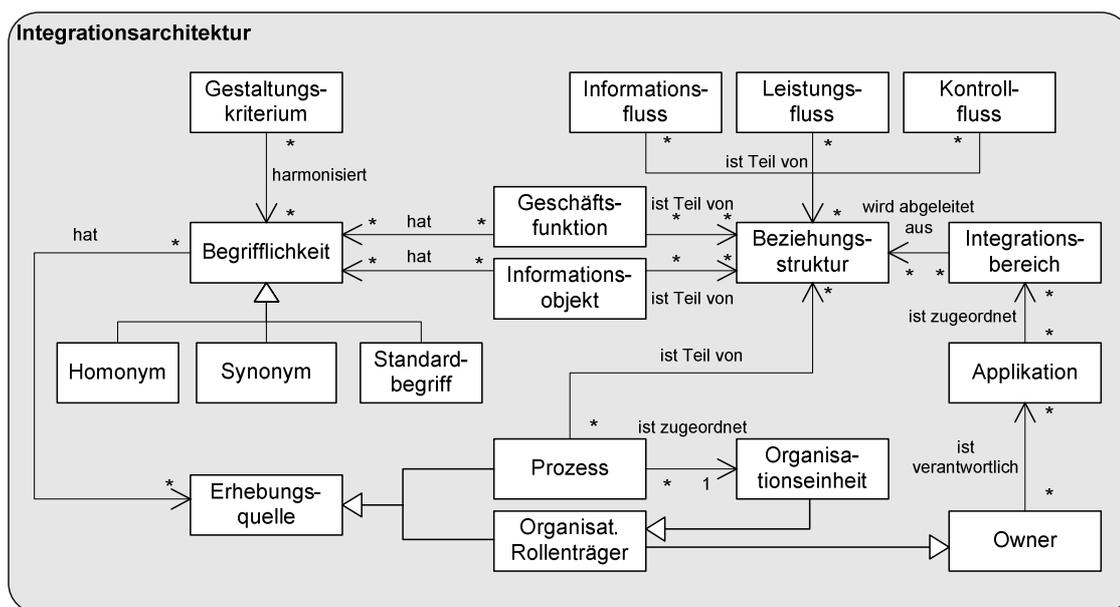


Abbildung 48: Metamodell der Integrationsarchitektur

4.3.2.4 Metamodell der Softwarekomponenten und Plattformen

Das in dieser Arbeit vorgeschlagene Metamodell der Softwarekomponenten- und Datenstrukturebene richtet sich in Teilen nach dem von SCHWINN definierten Metamodell für Applikationsintegrationsprojekte.⁴⁸⁴

Die Arbeit von SCHWINN konzentriert sich ausschliesslich auf Integrationsaspekte der Applikations- sowie Softwarekomponenten- und Datenstrukturebene. In der Realität existieren allerdings meist keine reinen Integrationsprojekte oder Integrationsarchitekturen, sondern sie sind Bestandteil eines umfassenden Projekts oder einer Unternehmensarchitektur. Zudem ist in der Regel von einem Integrationsprojekt die Gestaltung der Organisations- und in manchen Fällen auch der Strategieebene betroffen. Deshalb sollten die in seiner Arbeit vorgeschlagenen Methodenelemente in vorhandene Methoden bzw. Ansätze zur Gestaltung der Unternehmensarchitektur eingebunden werden.⁴⁸⁵ Da der Fokus des in der vorliegenden Arbeit präsentierten Ansatzes auf der Definition eines umfassenden Metamodells zur Abbildung der Unter-

⁴⁸⁴ Vgl. Schwinn (2005).

⁴⁸⁵ Vgl. Schwinn (2005), S. 242.

nehmensarchitektur liegt, werden lediglich die wesentlichen Metaentitätstypen des Metamodells seiner Methode übernommen und integriert.⁴⁸⁶ Diese und weitere Metaentitätstypen sowie deren Beziehungen werden im Folgenden beschrieben (vgl. Abbildung 49).

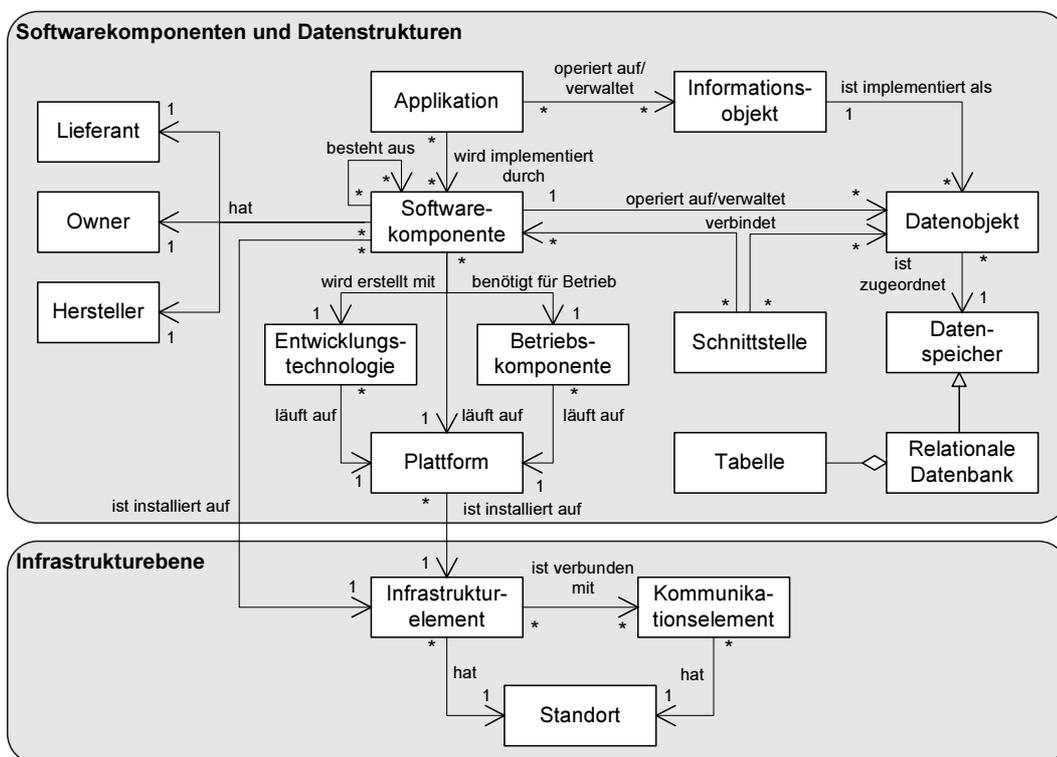


Abbildung 49: Metamodell der Softwarekomponenten und Datenstrukturen

Eine Applikation auf Applikationsebene wird durch eine oder mehrere Softwarekomponenten implementiert.⁴⁸⁷ Eine **Softwarekomponente** ist ein abgeschlossener Softwarebaustein, der eine bestimmte Menge an Applikationsfunktionen einer Applikation anbietet. Sie kann als Ganzes konfiguriert und ausgeliefert werden und bedarf der Wartung.⁴⁸⁸ Zusätzlich besitzt sie einen bestimmten **Owner**, **Hersteller** sowie **Lieferanten**. Softwarekomponenten operieren auf einem/mehreren Datenobjekt(en) oder verwalten ein/mehrere Datenobjekt(e), wobei ein **Datenobjekt** eine Kategorie von auf Dauer gespeicherten Daten darstellt, auf die wiederholt von Softwarekomponenten zurückgegriffen wird. Ein Datenobjekt korrespondiert mit genau einer Klasse eines UML-Klassendiagramms oder mit einem Datenobjekt einer anderen Datenmodellierungssprache, wie beispielsweise ER-Diagramme.⁴⁸⁹ Auf der Softwarekomponenten- und Datenstrukturebene repräsentiert ein Datenobjekt ein oder mehrere Informationsobjekte oder Teile davon. Es ist genau einem **Datenspeicher** zugeordnet und wird in diesem auf Dauer gespeichert. Ein Datenspeicher kann beispielsweise eine relationale oder objektorien-

⁴⁸⁶ Vgl. im Folgenden Schwinn (2005), S. 127-142 und Schelp/Schwinn (2005).

⁴⁸⁷ Vgl. Jonkers et al. (2004), S. 270.

⁴⁸⁸ Vgl. Schwinn (2005), S. 85.

⁴⁸⁹ Vgl. hierzu z.B. auch Jonkers et al. (2004), S. 271.

tierte Datenbank sein. Jede Softwarekomponente benötigt für den Betrieb eine **Betriebskomponente**.⁴⁹⁰ Sie stellt die unmittelbare Run-Time-Umgebung einer Softwarekomponente dar. Für die Erstellung einer Softwarekomponente wird eine bestimmte **Entwicklungstechnologie** (z.B. J2EE oder .NET) benötigt. Softwarekomponenten, Betriebskomponenten sowie Entwicklungstechnologien laufen auf einer bestimmten **Plattform**, wie beispielsweise Windows oder Unix. Softwarekomponenten sowie Datenobjekte können durch eine **Schnittstelle** miteinander verbunden sein.

Die detaillierte Abbildung der IT-Infrastruktur (z.B. Server, Router etc.) ist nicht von zentraler Bedeutung für die Unternehmensarchitektur. Stattdessen sollten dafür bereits existierende Ansätze und Werkzeuge für das IT-Portfoliomanagement zum Einsatz kommen.⁴⁹¹ Da allerdings zu diesen Werkzeugen und deren zugrunde liegenden Konzepten Anknüpfungspunkte definiert sein müssen, um beispielsweise die Konsistenz der Informationen gewährleisten sowie Impact-Analysen durchführen zu können, wird in Abschnitt 4.4 ein Infrastrukturmodell mit den Metaentitätstypen „Infrastrukturelement“ und „Kommunikationselement“ eingeführt. Für ausgewählte Softwarekomponenten und Plattformen kann somit dokumentiert werden, auf welchen Infrastrukturelementen diese installiert sind. Dadurch lassen sich die Auswirkungen eines Ausfalls eines bestimmten Infrastruktur- oder Kommunikationselements analysieren und vorhersagen.

4.3.2.5 Metamodell des Datenstrukturmodells

Informationsobjekte werden in Form von Datenobjekten implementiert. Für die Modellierung der Datenobjekte sowie deren Beziehungen, d.h. der detaillierten Datenstruktur eines fachlichen Anwendungsbereichs, werden in der Regel objektorientierte Klassendiagramme, wie beispielsweise UML-Klassendiagramme⁴⁹², oder ER-Diagramme⁴⁹³ verwendet.

In diesem Ansatz kommen UML-Klassendiagramme für die Abbildung der Datenstrukturen zum Einsatz, da sich die UML als führender Notationsstandard in Praxis und Wissenschaft etabliert hat.⁴⁹⁴ Ausserdem werden auch bereits die Metamodelle und Vorgehensmodelle dieser Arbeit mit Hilfe der UML dargestellt. Sie kommt somit im Rahmen dieser Arbeit sowohl als Modellierungs- als auch als Metamodellierungssprache zum Einsatz.

Wie bereits in Abschnitt 2.2.4 beschrieben, bietet ein UML-Klassendiagramm die Möglichkeit, die statische Struktur eines Systems darzustellen. Es zeigt dessen wesentliche statische Eigenschaften sowie ihre Beziehungen zueinander. Das zentrale Konzept sind Klassen. Eine

⁴⁹⁰ Vgl. *Schwinn* (2005), S. 85.

⁴⁹¹ Vgl. hierzu z.B. *Adams* (2005).

⁴⁹² Für eine detaillierte Darstellung des Metamodells von UML-Klassendiagrammen vgl. z.B. *OMG* (2005).

⁴⁹³ Für eine detaillierte Darstellung des Metamodells von ER-Diagrammen vgl. z.B. *Chen* (1976) und *Scheer* (2001), S. 72.

⁴⁹⁴ Vgl. z.B. *Balzert* (2000), S. V.

Klasse wird in der UML 2 als eine Sammlung von Exemplaren bzw. Objekten definiert, die über gemeinsame Eigenschaften, Einschränkungen und eine gemeinsame Semantik verfügen.⁴⁹⁵ Jede Klasse stellt demnach einen abstrahierten Sammelbegriff für eine Menge gleichartiger Dinge bzw. Objekte dar.⁴⁹⁶ Jede Klasse besitzt Attribute, Operationen und Beziehungen (vgl. Abbildung 50).

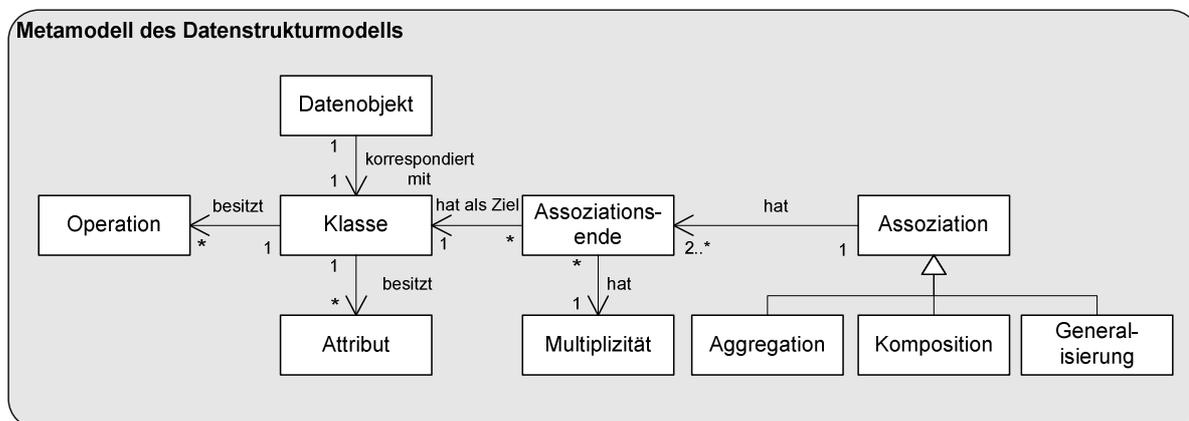


Abbildung 50: Vereinfachtes Metamodell des UML-Klassendiagramms

Attribute beschreiben bestimmte Charakteristika einer Klasse. Eine **Operation** definiert Name, Typ und Parameter für den Aufruf des internen Verhaltens von Objekten einer Klasse und bietet somit den Anknüpfungspunkt zu den Verhaltensdiagrammen der UML, die die dynamischen Konstrukte eines Systems beschreiben (z.B. Aktivitäten, Interaktionen etc.). Klassen können über Assoziationen miteinander verbunden sein.

Eine **Assoziation** beschreibt eine Menge semantisch gleichartiger Beziehungen zwischen zwei oder mehreren Klassen. Eine Assoziation wird weiter spezialisiert in eine **Aggregation**, **Komposition** und **Generalisierung**. Jede Assoziation besitzt zwei oder mehrere Assoziationsenden, für die eine **Multiplizität** definiert sein kann.

Die in Form eines oder mehrerer objektorientierter Klassendiagramme detailliert abgebildete Datenstruktur eines fachlichen Anwendungsbereichs kann als Vorlage für die softwaretechnische Implementierung dienen.

4.3.2.6 Metamodell der Autorisierung

Durch die zunehmende softwaretechnische Unterstützung von Geschäftsprozessen kann ein Ausfall von Informationssystemen oder der unerwünschte Zugriff auf unternehmensinterne Daten zu erheblichen Geschäftseinbußen führen. Die Informationssysteme und deren Daten müssen daher entsprechend geschützt werden. Die Autorisierung stellt ein wichtiges Konzept

⁴⁹⁵ Vgl. *OMG* (2005), S. 45.

⁴⁹⁶ Vgl. *Jeckle et al.* (2004b), S. 32.

für die Gewährleistung der Sicherheit von Informationssystemen in einem Unternehmen dar und sollte daher auch Bestandteil einer Unternehmensarchitektur sein.⁴⁹⁷

Der Begriff Autorisierung bezeichnet die Verwaltung und Kontrolle von Zugriffsrechten.⁴⁹⁸ Synonym wird oftmals der Begriff Zugriffskontrolle verwendet. Die Kontrolle bzw. Überprüfung von Zugriffsrechten umfasst die Behandlung von Zugriffen auf Informationssysteme und Daten und deren Gewährung oder Zurückweisung entsprechend den definierten Rechten.⁴⁹⁹ Die Verwaltung von Zugriffsrechten umfasst dagegen das Erteilen, Entziehen und Pflegen von Zugriffsrechten.⁵⁰⁰

Die zuvor definierten Modelle der Aufbau- und Ablauforganisation auf Organisationsebene sowie die Modelle der Softwarekomponenten und Datenstrukturen bilden die Basis für die Vergabe von Zugriffsberechtigungen. Anknüpfungspunkte des Metamodells der Autorisierung zu diesen Modellen sind die Metaentitätstypen menschlicher Rollenträger und Prozessrolle der Aufbauorganisation, Aktivität der Ablauforganisation sowie die Metaentitätstypen Softwarekomponente, Applikationsfunktion und Datenobjekt des SW-Komponenten- und Datenstrukturmodells. Diese Metaentitätstypen werden nun im Autorisierungsmodell mit den Metaentitätstypen Benutzerkonto, Ressource, Berechtigung und Berechtigungskonzept in Beziehung gesetzt.⁵⁰¹ Das Metamodell der Autorisierung definiert somit eine ebenenübergreifende Sicht, die Elemente der Informationssystem- und Organisationsebene enthält (vgl. Abbildung 51).

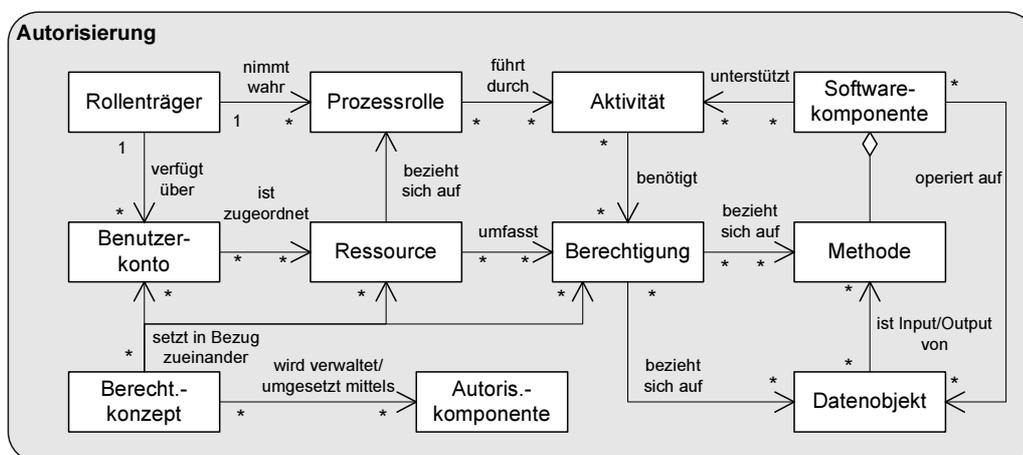


Abbildung 51: Metamodell der Autorisierung⁵⁰²

⁴⁹⁷ Vgl. im Folgenden Wortmann (2005), S. 123-125.

⁴⁹⁸ Vgl. Jonscher/Dittrich (1994), S. 27.

⁴⁹⁹ Vgl. Samarati/de Capitani di Vimercati (2002), S. 137.

⁵⁰⁰ Vgl. Rupprecht (2002), S. 22.

⁵⁰¹ Vgl. Wortmann (2005), S. 123.

⁵⁰² Vgl. Wortmann (2005), S. 123.

Damit die menschlichen Rollenträger die informationstechnisch unterstützten Aktivitäten eines Prozesses durchführen können, müssen sie über entsprechende (System-)Berechtigungen verfügen.⁵⁰³ Diese Berechtigungen regeln den Zugriff auf die Methoden und Datenobjekte der existierenden Softwarekomponenten und werden über Benutzerkonten den Rollenträgern zugeordnet. Eine **Berechtigung** ermöglicht somit die Ausführung einer bestimmten **Methode** auf einem bestimmten geschützten **Datenobjekt**.⁵⁰⁴ Jeder **Rollenträger** verfügt über kein, ein oder mehrere Benutzerkonto/en, wobei jedem **Benutzerkonto** eine oder mehrere Ressource/n zugeordnet sein können.

Ein Benutzerkonto repräsentiert eine Person im Informationssystem.⁵⁰⁵ Eine **Ressource** ist ein intermediäres Konstrukt, das für die Zuweisung der Berechtigungen zu Benutzerkonten verwendet wird.⁵⁰⁶ Das **Berechtigungskonzept** setzt die einzelnen Berechtigungen miteinander in Beziehung und legt damit fest, welcher Benutzer auf welche Methode bzw. auf welches Datenobjekt zugreifen darf. Die im Berechtigungskonzept definierten Regelungen werden informationstechnisch in einer **Autorisierungskomponente** abgebildet und verwaltet.

4.3.2.7 Definition der Metaentitätstypen auf Informationssystemebene

Die folgende Tabelle 12 spezifiziert die im vorhergehenden Abschnitt beschriebenen Metaentitätstypen sowie deren Beziehungen im Detail.

Applikation		
Definition	Eine Applikation enthält Applikationsfunktionen, die eine oder mehrere Funktion/en auf Basis von Informationstechnologie realisieren. Die Zusammenfassung mehrerer Applikationsfunktionen und ihrer Operationen auf Informationsobjekten bildet die Applikation. ⁵⁰⁷	
Beziehungen zu	Informationsobjekt	Eine Applikation verwaltet/operiert auf einem oder mehreren Informationsobjekt/en.
	Applikationsfunktion	Eine Applikation beinhaltet eine oder mehrere Applikationsfunktion/en.
	Softwarekomponente	Eine Applikation wird durch eine oder mehrere Softwarekomponente/n implementiert.
	Applikationsdomäne	Eine Applikation ist genau einer Applikationsdomäne zugeordnet.
	Owner	Eine Applikation besitzt eine Organisationseinheit als Owner.
Applikationsdomäne		
Definition	Eine Applikationsdomäne ist eine logische Zusammenfassung von Applikationen. Applikationsdomänen werden zur Beherrschung der Komplexität gebildet und sind unabhängig von konkreten Applikationen und Softwarekomponenten. ⁵⁰⁸	

⁵⁰³ Vgl. Wortmann (2005), S. 123.

⁵⁰⁴ Vgl. Ferraiolo et al. (2001), S. 233.

⁵⁰⁵ Vgl. Kern et al. (2002), S. 46.

⁵⁰⁶ Für eine genaue Definition und Begründung für die Einführung dieses Metaentitätstyps vgl. Wortmann (2005), S. 123.

⁵⁰⁷ Vgl. Schwinn (2005), S. 130.

⁵⁰⁸ Vgl. Schwinn (2005), S. 74. Synonym für eine Applikationsdomäne werden häufig die Begriffe „logische Einheit“ (vgl. Lankes et al. (2005), S. 1445) oder „Building Block“ (Vgl. Jung (2004), S. 313) verwendet.

Beziehungen zu	Applikation	Eine Applikationsdomäne umfasst eine oder mehrere Applikation/en.
	Prozess	Eine Applikationsdomäne unterstützt einen oder mehrere Geschäftsprozess/e auf Organisationsebene.
	OE	Einer Applikationsdomäne ist eine verantwortliche Organisationseinheit zugeordnet.
Applikationsfunktion		
Definition	Eine Applikationsfunktion ist eine von einer bestimmten Applikation realisierte Funktion.	
Beziehungen zu	Geschäftsfunktion	Eine Applikationsfunktion realisiert eine oder mehrere Geschäftsfunktion/en auf Organisationsebene.
	Applikation	Eine Applikationsfunktion ist Teil einer Applikation.
Assoziation		
Definition	Eine Assoziation beschreibt eine Menge semantisch gleichartiger Beziehungen zwischen zwei oder mehreren Klassen. ⁵⁰⁹	
Beziehungen	Assoziationssende	Eine Assoziation besitzt zwei oder mehrere Assoziationssenden.
Autorisierungskomponente		
Definition	Eine Autorisierungskomponente ist eine Softwarekomponente, die Funktionalitäten zur Verwaltung und/oder Kontrolle von Berechtigungen bereitstellt. ⁵¹⁰	
Beziehungen zu	Berechtigungskonzept	Eine Autorisierungskomponente implementiert und verwaltet ein oder mehrere Berechtigungskonzept/e.
Benutzerkonto		
Definition	Ein Benutzerkonto repräsentiert eine bestimmte Person, Stelle oder Rolle im Informationssystem. ⁵¹¹	
Beziehungen zu	Rollenträger	Ein Benutzerkonto ist einem bestimmten Rollenträger auf Organisationsebene zugeordnet.
Berechtigung		
Definition	Eine Berechtigung ermöglicht die Ausführung einer Methode auf einem geschützten Datenobjekt. ⁵¹²	
Beziehungen zu	Aktivität	Eine Berechtigung wird für die Durchführung einer oder mehrerer Aktivität/en benötigt.
	Datenobjekt	Eine Berechtigung bezieht sich auf ein oder mehrere geschützte Datenobjekt/e.
	Methode	Eine Berechtigung bezieht sich auf eine oder mehrere geschützte Methode/n.
Berechtigungskonzept		
Definition	Ein Berechtigungskonzept legt fest, welcher Benutzer auf welche Methoden und/oder Datenobjekte zugreifen darf. ⁵¹³	
Beziehungen zu	Autorisierungskomponente	Ein Berechtigungskonzept wird in einer oder mehreren Autorisierungskomponente/n umgesetzt und verwaltet.
	Berechtigung	Ein Berechtigungskonzept setzt mehrere Berechtigungen in Bezug zueinander.

⁵⁰⁹ Vgl. *OMG* (2005), S. 36.

⁵¹⁰ Vgl. *Wortmann* (2005), S. 124.

⁵¹¹ Vgl. *Kern et al.* (2002), S. 46.

⁵¹² Vgl. *Ferraiolo et al.* (2001), S. 233.

⁵¹³ Vgl. *Seufert* (2001), S. 31f.

Betriebskomponente		
Definition	Eine Betriebskomponente ist ein technisches System (Subsystem), welches die unmittelbare Run-Time-Umgebung einer Softwarekomponente bildet.	
Beziehungen zu	Plattform	Eine Betriebskomponente läuft auf einer bestimmten Plattform.
	Softwarekomponente	Eine Betriebskomponente unterstützt den Betrieb einer oder mehrerer Softwarekomponente/n.
Beziehungsstruktur		
Definition	Eine Beziehungsstruktur beschreibt den Leistungs-, Informations- und Kontrollfluss zwischen Geschäftsfunktionen, Informationsobjekten und Prozessen/Organisationseinheiten/Leistungstypen. ⁵¹⁴	
Beziehungen zu	Integrationsbereich	Aus einer Beziehungsstruktur werden ein oder mehrere Integrationsbereich/e abgeleitet.
Datenobjekt		
Definition	Ein Datenobjekt stellt eine Kategorie von auf Dauer gespeicherten Daten dar, auf die wiederholt von Softwarekomponenten zurückgegriffen wird. ⁵¹⁵	
Beziehungen zu	Informationsobjekt	Ein Datenobjekt repräsentiert auf der Softwarekomponenten- und Datenstrukturebene ein oder mehrere Informationsobjekt/e oder Teile davon.
	Softwarekomponente	Ein Datenobjekt kann von einer oder mehreren Softwarekomponente/n manipuliert werden.
	Datenspeicher	Ein Datenobjekt ist genau einem Datenspeicher zugeordnet.
	Methode	Eine Methode verwendet/erzeugt ein oder mehrere Datenobjekt/e.
	Klasse	Ein Datenobjekt korrespondiert mit genau einer Klasse.
Datenspeicher		
Definition	Ein Datenspeicher ermöglicht die dauerhafte Speicherung von Datenobjekten.	
Beziehungen zu	Datenspeicherobjekt	Ein Datenspeicher enthält mehrere dauerhaft gespeicherte Datenobjekte.
Entwicklungstechnologie		
Definition	Eine Entwicklungstechnologie, z. B. J2EE/.NET, eignet sich für die Erstellung von Softwarekomponenten.	
Beziehungen zu	Plattform	Eine Entwicklungstechnologie läuft auf einer Plattform.
	Softwarekomponente	Eine Entwicklungsplattform unterstützt die Erstellung einer oder mehrerer Softwarekomponente/n.
Informationsobjekt		
Definition	Ein Informationsobjekt stellt eine Sammlung von auf Dauer gespeicherten Informationen dar, auf die wiederholt zurückgegriffen wird. ⁵¹⁶	
Beziehungen zu	Applikation	Ein Informationsobjekt ist einer verwaltenden und einer oder mehreren operierenden Applikation/en zugeordnet.
	Datenobjekt	Ein Informationsobjekt wird auf der Softwarekomponenten- und Datenstrukturebene als ein oder mehrere Datenobjekt/e repräsentiert.

⁵¹⁴ Es wird dabei vorausgesetzt, dass Leistungstypen in einem prozessorientiert organisierten Unternehmen mit Organisationseinheiten bzw. Prozessen korrespondieren. Vgl. hierzu *Winter* (2006), S. 12.

⁵¹⁵ Vgl. *Schwinn* (2005), S. 23.

⁵¹⁶ Vgl. *Schwinn* (2005), S. 138.

Integrationsbereich		
Definition	Ein Integrationsbereich umfasst Informationsobjekte, Geschäftsfunktionen und Organisationseinheiten/Prozesse, die eine enge Beziehungsstruktur besitzen. ⁵¹⁷	
Beziehungen zu	Beziehungsstruktur	Ein Integrationsbereich wird aus einer oder mehreren Beziehungsstruktur/en abgeleitet.
	Applikation	Einem Integrationsbereich ist eine Applikation zugeordnet.
Klasse		
Definition	Eine Klasse repräsentiert eine Sammlung von Exemplaren bzw. Objekten, die über gemeinsame Eigenschaften, Einschränkungen und eine gemeinsame Semantik verfügen. ⁵¹⁸	
Beziehungen zu	Attribut	Eine Klasse besitzt ein oder mehrere Attribut/e.
	Operation	Eine Klasse besitzt eine oder mehrere Operation/en.
	Datenobjekt	Eine Klasse korrespondiert mit genau einem Datenobjekt.
	Assoziationsende	Eine Klasse ist das Ziel eines Assoziationsendes oder mehrerer.
Methode		
Definition	Eine Methode implementiert eine bestimmte Funktionalität einer Softwarekomponente und stellt diese nach aussen hin zur Verfügung. ⁵¹⁹	
Beziehungen zu	Softwarekomponente	Eine Methode ist Teil einer Softwarekomponente.
	Datenobjekt	Eine Methode verwendet/erzeugt ein oder mehrere Datenobjekt/e.
	Berechtigung	Auf eine Methode bezieht/beziehen sich eine oder mehrere Berechtigung/en.
	Operation	Eine Methode ruft eine oder mehrere Operation/en zur Manipulation von Datenobjekten auf.
Operation		
Definition	Eine Operation definiert das Verhalten einer Klasse bzw. deren Instanzen. Sie wird durch ihre Signatur (Operationsname, Parameter, Rückgabewert) beschrieben. ⁵²⁰	
Beziehungen zu	Klasse	Eine Operation ist genau einer Klasse zugeordnet.
	Methode	Eine Operation wird von einer oder mehreren Methode/n aufgerufen.
Owner		
Definition	Ein Owner ist eine Organisationseinheit, eine Stelle oder eine Person, die für ein bestimmtes Element der Systemebene (z.B. Applikation, Softwarekomponente, Informationsobjekt, Datenspeicher) verantwortlich ist.	
Beziehungen zu	Applikation	Ein Owner ist für eine oder mehrere Applikation/en verantwortlich.
	Softwarekomponente	Ein Owner ist für eine oder mehrere Softwarekomponente/n verantwortlich.
	Datenspeicher	Ein Owner ist für einen oder mehrere Datenspeicher verantwortlich.

⁵¹⁷ Vgl. *Winter* (2006), S. 17-20.

⁵¹⁸ Vgl. *OMG* (2005), S. 45.

⁵¹⁹ Vgl. *Schwinn* (2005), S. 22.

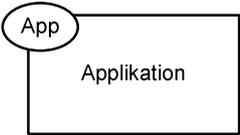
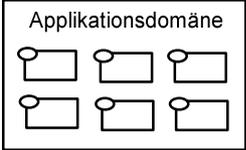
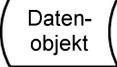
⁵²⁰ Vgl. *OMG* (2005), S. 99.

Plattform		
Definition	Betriebskomponenten und Entwicklungstechnologien laufen auf Plattformen. Beispiele für Plattformen sind OS/390, Windows oder Unix.	
Beziehungen zu	Softwarekomponente	Auf einer Plattform laufen eine oder mehrere Softwarekomponente/n.
	Infrastrukturelement	Eine Plattform ist auf einem Infrastrukturelement installiert.
Schnittstelle		
Definition	Eine Schnittstelle beschreibt, wie von aussen auf Softwarekomponenten, Applikationsfunktionen oder Datenobjekte zugegriffen werden kann.	
Beziehungen zu	Applikationsfunktion	Eine Schnittstelle kann von einer Applikationsfunktion angeboten werden.
	Softwarekomponente	Eine Schnittstelle kann von einer Softwarekomponente angeboten werden.
	Datenobjekt	Eine Schnittstelle kann von einem Datenobjekt angeboten werden.
Softwarekomponente		
Definition	Eine Softwarekomponente ist ein abgeschlossener Softwarebaustein, der eine bestimmte Menge an Applikationsfunktionen einer Applikation anbietet.	
Beziehungen zu	Betriebskomponente	Eine Softwarekomponente benötigt für den Betrieb eine Betriebskomponente.
	Entwicklungstechnologie	Eine Softwarekomponente wird mit einer Entwicklungstechnologie entwickelt.
	Applikation	Eine Softwarekomponente ist Teil einer oder mehrerer Applikation/en.
	Datenobjekt	Eine Softwarekomponente operiert auf/verwaltet Datenobjekte/n.
	Methode	Eine Softwarekomponente enthält eine oder mehrere Methode/n, die Datenobjekte verwenden/erzeugen.
	Plattform	Eine Softwarekomponente läuft auf einer Plattform.

Tabelle 12: Metaentitätstypen der Informationssystemebene

4.3.2.8 Festlegung der verwendeten Symbole auf Informationssystemebene

Die folgende Tabelle 13 zeigt die Notationselemente (Symbole) für die auf Organisationsebene definierten Metaentitäts- und Beziehungstypen. Metaentitätstypen ohne zugeordnetes Notationselement werden wiederum lediglich textuell erfasst.

Symbole der Informationssystemebene		
Applikation	Applikationsdomäne	Applikationsfunktion
		
Benutzerkonto	Betriebskomponente	Datenobjekt
		

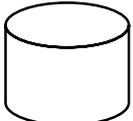
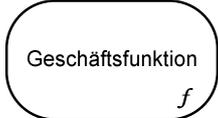
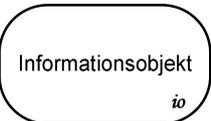
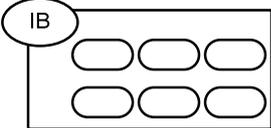
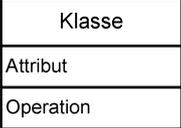
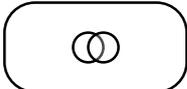
Datenspeicher	Entwicklungstechnologie	Geschäftsfunktion
 Datenspeicher	 Entwicklungs- technologie	 Geschäftsfunktion <i>f</i>
Informationsobjekt	Integrationsbereich	Klasse
 Informationsobjekt <i>io</i>	 IB	 Klasse Attribut Operation
Plattform	Schnittstelle	Softwarekomponente
 Plattform		 SW-Komponente
Datenfluss (zw. SW-Komponente/Datenspeicher und Schnittstelle)	Informationsfluss (von Applikation zu Applikation)	ist zugeordnet (von Integrationsbereich zu Applikation)
		
Generalisierung (von Klasse zu Klasse)	Assoziation (von Klasse zu Klasse)	Aggregation (von Klasse zu Klasse)
		

Tabelle 13: Symbole der Informationssystemebene

Analog zur Strategie- und Organisationsebene handelt es sich bei Beziehungstypen ohne zugeordnetes Notationselement um eine modellübergreifende Beziehung (z.B. zwischen „Applikation“ und „Informationsobjekt“) oder um eine Aggregationsbeziehung (z.B. zwischen „Applikationsdomäne“ und „Applikation“).

4.4 Modellierung auf Infrastrukturebene

Die IT-Infrastruktur eines Unternehmens ist der Ausschnitt der Informationstechnik (Computer- und Kommunikationstechnik), der in einem Unternehmen zum Einsatz kommt.⁵²¹ Sie ist im engeren Sinne nicht mehr Betrachtungsgegenstand des Business Engineering (vgl. Abschnitt 2.3.2). Die im Grundlagenteil durchgeführte Analyse der Literatur zum Thema Unternehmensarchitektur (vgl. Abschnitt 2.3.1) sowie die Diskussion existierender Bezugsrahmen in der Praxis (vgl. Abschnitt 3.3) haben allerdings gezeigt, dass die eingesetzten Infrastrukturelemente (z.B. Server, Workstations, Netzwerkkomponenten etc.) sowie deren Beziehungen einen wichtigen Bestandteil einer Unternehmensarchitektur darstellen. Gerade im Hinblick auf die spätere Durchführung so genannter Impact-Analysen

⁵²¹ Vgl. Brenner (1994).

ist die Dokumentation der auf einem Infrastrukturelement installierten Softwarekomponenten und Plattformen erforderlich. Dadurch lässt sich beispielsweise feststellen, welche anderen Elemente und Bereiche der Unternehmensarchitektur durch den Ausfall eines bestimmten Infrastrukturelements betroffen sind.

Für die detaillierte Abbildung der zugrunde liegenden Infrastruktur existieren bereits zahlreiche Modelle, Methoden und Werkzeuge (z.B. für das IT-Portfoliomanagement⁵²²). Diese sollten auch weiterhin im Unternehmen zum Einsatz kommen. Um einen Anknüpfungspunkt zu den Detailmodellen herzustellen, werden auf der Infrastrukturebene die beiden Metaentitätstypen „Infrastrukturelement“ und „Kommunikationselement“ eingeführt (vgl. Abbildung 52).

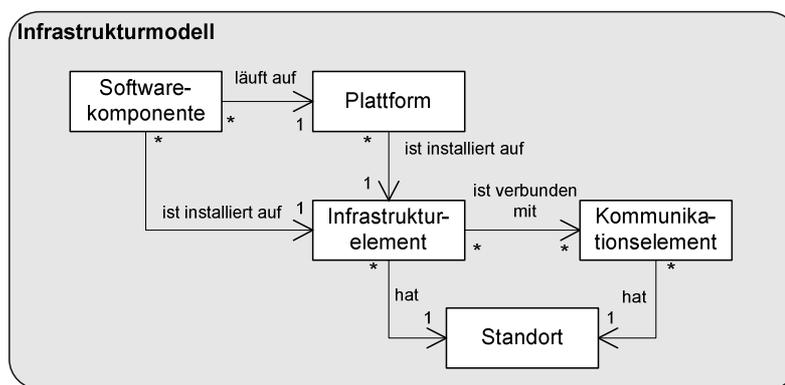


Abbildung 52: Metamodell des Infrastrukturmodells

Ein **Infrastrukturelement** bezeichnet einen Computer bzw. alle festen Bestandteile eines Computers inklusive der Peripherie. Unter Peripheriegeräten werden alle Geräte, die an einen Computer angeschlossen werden können, verstanden. Hierzu zählen z.B. Drucker, Monitor, Tastatur, Maus, Modem, Scanner etc. Auf einem Infrastrukturelement können mehrere Softwarekomponenten und Plattformen installiert sein. Infrastrukturelemente sind über Kommunikationselemente miteinander verbunden. Ein **Kommunikationselement** bezeichnet ein Netzwerkelement, das die Schnittstelle zwischen Infrastrukturelementen (z.B. zwischen Clients und Servern) bzw. zwischen räumlich getrennten Netzwerken bildet. Zu übermittelnde Daten werden über Kommunikationselemente und Datenleitungen an den Empfänger (Infrastrukturelement) versandt. Die folgende Tabelle 14 fasst die Metaentitätstypen der Infrastrukturebene sowie deren Beziehungen zusammen.

Infrastrukturelement	
Definition	Ein Infrastrukturelement bezeichnet einen Computer bzw. alle festen Bestandteile eines Computers inklusive der Peripherie. Unter Peripheriegeräten werden alle Geräte, die an einen Computer angeschlossen werden können, verstanden. Hierzu zählen z.B. Drucker, Monitor, Tastatur, Maus, Modem, Scanner usw.
Beziehungen zu	Kommunikationselement Ein Infrastrukturelement ist mit keinem, einem oder mehreren Kommunikationselement/en verbunden.

⁵²² Für einen Überblick über Werkzeuge für das IT-Portfoliomanagement vgl. z.B. Adams (2005).

	IT-Service	Ein Infrastrukturelement ist Teil eines oder mehrerer IT-Services. ⁵²³
	Plattform	Auf einem Infrastrukturelement ist/sind eine oder mehrere Plattform/en installiert.
	Softwarekomponente	Auf einem Infrastrukturelement ist/sind eine oder mehrere Softwarekomponent/en installiert.
	Standort	Ein Infrastrukturelement hat einen bestimmten Standort.
Kommunikationselement		
Definition	Ein Kommunikationselement bezeichnet ein Netzwerkelement, das die Schnittstelle zwischen Computern (z.B. zwischen Clients und Servern) bzw. zwischen räumlich getrennten Netzwerken bildet. Zu übermittelnde Daten werden über Kommunikationselemente und Datenleitungen an den Empfänger (Computer) versandt.	
Beziehungen zu	Infrastrukturelement	Ein Kommunikationselement ist mit keinem, einem oder mehreren Infrastrukturelement/en verbunden.
	IT-Service	Ein Kommunikationselement ist Teil eines oder mehrerer IT-Services. ⁵²⁴
	Standort	Ein Kommunikationselement hat einen bestimmten Standort.

Tabelle 14: Metaentitätstypen der Infrastrukturebene

Für die Darstellung der auf Infrastrukturebene definierten Metaentitäts- und Beziehungstypen können die in Tabelle 15 dargestellten Symbole verwendet werden.

Symbole der Infrastrukturebene		
Infrastrukturelement	Kommunikationselement	ist verbunden mit (von I-Element zu K-Element)
 Infrastrukturelement	 Kommunikationselement	

Tabelle 15: Symbole der Infrastrukturebene

4.5 Vorschlag zur Integration serviceorientierter Konzepte

Die Kundenorientierung stellt für heutige Betreiber der IT eine der wichtigsten strategischen Ausrichtungen im Rahmen des strategischen IT-Managements dar.⁵²⁵ Sowohl externe IT-Betreiber als auch interne IT-Abteilungen treten nicht mehr als reine Technologieanbieter auf, sondern positionieren sich zunehmend als Dienstleister gegenüber Anwendern von IT. Diese Anwender fragen als Dienstnehmer im Rahmen der Durchführung ihrer Aktivitäten innerhalb der Geschäftsprozesse nach informationsverarbeitender Funktionalität mit einer definierten Qualität.⁵²⁶ Die Kooperation zwischen IT-Dienstleister und Dienstnehmer wird dabei durch so

⁵²³ Vgl. hierzu Abschnitt 4.5.
⁵²⁴ Vgl. hierzu Abschnitt 4.5.
⁵²⁵ Vgl. Hochstein/Hunziker (2003), S. 46.
⁵²⁶ Vgl. Mayerl et al. (2005), S. 1.

genannte IT-Services gestaltet, für die bestimmte Leistungs- und Qualitätsmerkmale verhandelt und in Form von Service Level Agreements (SLA) vertraglich festgelegt werden. Die vertragliche Festlegung von SLAs ist unabhängig davon, ob der IT-Anwender bzw. Dienstnehmer die Leistungen von einer internen IT-Abteilung oder einem externen IT-Dienstleistungsunternehmen bezieht.

Dieser Wandel von einem technologieorientierten IT-Betreiber zu einem kundenorientierten IT-Dienstleister mit einer methodischen Gestaltung der internen IT-Prozesse kann nur mit Hilfe eines serviceorientierten IT-Management vollzogen werden.⁵²⁷ Mittlerweile existiert eine Reihe von serviceorientierten Referenzmodellen für das IT-Management. Die wohl umfangreichste Sammlung von Vorgehensbeschreibungen im Bereich IT Service Management (ITSM) und den internationalen De-facto-Standard für IT-Dienstleister stellt die IT Infrastructure Library (ITIL) dar.⁵²⁸ Dabei handelt es sich um ein generisches Referenzmodell, mit dem IT-Leistungen geplant, überwacht und gesteuert werden können.⁵²⁹ Die wichtigsten Komponenten sind „Service Delivery“ und „Application Management“ auf der taktischen Ebene sowie „Service Support“ und „ICT Infrastructure Management“ auf der operativen Ebene. Beiträge für die Bereitstellung von IT-Services liefert ITIL mit den Empfehlungen zum Service Support und Service Delivery, und hier insbesondere mit der Definition des Prozesses für das Service Level Management (SLM).

Eine strukturiert entwickelte und klar dokumentierte Unternehmensarchitektur bildet eine wertvolle Grundlage für das IT-Service-Management.⁵³⁰ IT-Verantwortliche können mit ihrer Hilfe einen klaren Blick auf die IT-Infrastruktur und Applikationen, die unterstützten Geschäftsprozesse sowie auf die unterschiedlichen Abhängigkeiten zwischen diesen Bereichen erhalten. Davon profitieren nahezu alle in ITIL definierten Kernprozesse. Um effiziente und effektive Betriebsprozesse gewährleisten zu können, sollten deshalb die Schlüsselemente des ITSM, wie beispielsweise IT-Services, SLAs und IT-Service-Prozesse sowie deren Beziehungen zu anderen Bereichen und Elementen der Unternehmensarchitektur, dokumentiert sein. Im Folgenden wird daher ein Metamodell für eine IT-Servicesicht vorgeschlagen (vgl. Abbildung 53) und mit den anderen in dieser Arbeit definierten Sichten auf die Unternehmensarchitektur integriert (vgl. Abbildung 54). In Tabelle 16 werden die Metaentitätstypen der IT-Servicesicht sowie deren Beziehungen genauer spezifiziert.

Ein **IT-Service** stellt eine Menge zusammengehörender Dienstleistungen dar, die von einem IT-System oder der IT-Abteilung bzw. einem externen IT-Dienstleister bereitgestellt werden, um die Geschäftsprozesse des Unternehmens oder Kunden zu unterstützen.⁵³¹ IT-Services

⁵²⁷ Vgl. Hochstein/Hunziker (2003), S. 46 und Hochstein et al. (2004), S. 68.

⁵²⁸ Vgl. Commerce (2000).

⁵²⁹ Vgl. Hochstein/Hunziker (2003), S. 50.

⁵³⁰ Vgl. Lankhorst (2005), S. 17-18 und IFEAD (2005), S. 17.

⁵³¹ Vgl. z.B. Commerce (2000) und Zarnekow et al. (2005).

variieren in ihrer Ausprägung sehr stark. Sie können sowohl aus einzelnen **Softwarekomponenten** (bzw. deren Funktionalitäten) als auch aus kompletten Bündeln bestehen. Diese Bündel enthalten heterogene Servicekomponenten, wie beispielsweise Softwarekomponenten, **Infrastrukturelemente** und entsprechende **Zusatzdienste**. Die Zusatzdienste umfassen in der Regel Informations-, Beratungs-, Schulungs-, Problembehebungs- sowie Änderungsdurchführungsleistungen und werden durch definierte Betriebsprozesse (**IT-Service-Prozesse**) innerhalb der IT-Abteilung bzw. des externen IT-Dienstleisters erbracht.⁵³² Ein IT-Service-Prozess stellt somit, im Sinne von zu erbringenden Dienstleistungen, eine Teilkomponente des IT-Services dar. Die von einem IT-Dienstleister angebotenen IT-Services werden in einem **Servicekatalog** mit entsprechenden Kostenangaben erfasst.⁵³³ Analog zu den Geschäftsprozessen und Applikationen ist jedem IT-Service eine verantwortliche **Organisationseinheit** zugeordnet.

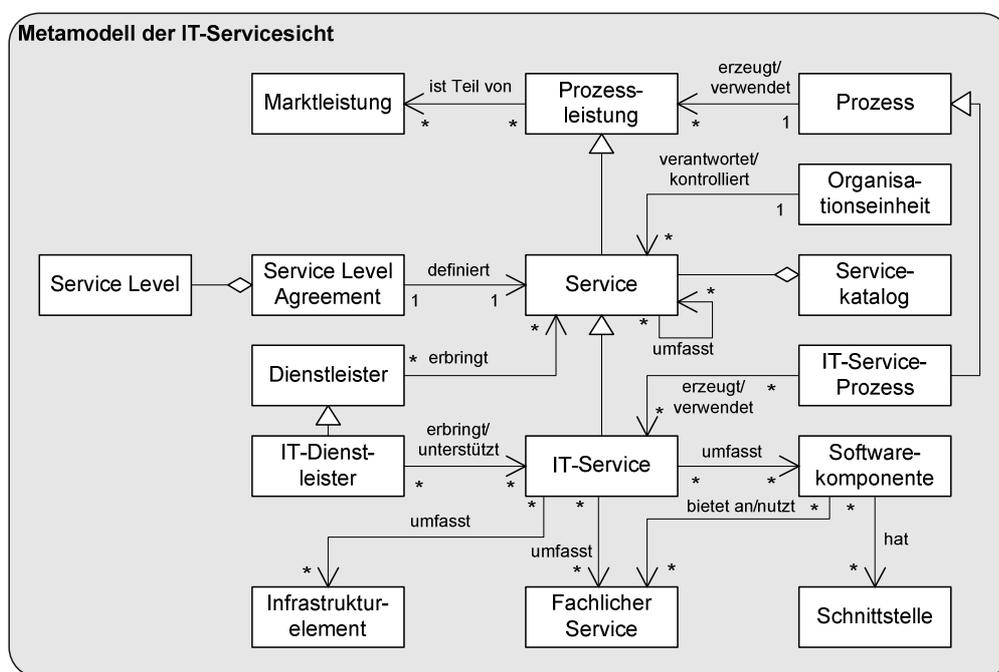


Abbildung 53: Metamodell der IT-Servicesicht

Ein **Service Level Agreement (SLA)** ist eine schriftliche Vereinbarung zwischen dem Service-Lieferanten und dem Kunden, welche die für den IT-Service vereinbarten Service Level dokumentiert.⁵³⁴ Ein **Service Level** ist wiederum ein (Qualitäts-)Merkmal eines bereitgestellten IT-Services (wie bspw. Service-Zeiten, Verfügbarkeitsrate und Antwortzeiten). Nach Abschluss von SLAs steht ein IT-Dienstleister in der Pflicht, die definierten Services mit der vereinbarten Qualität (Service Levels) den jeweiligen Dienstnehmern zur Verfügung zu stel-

⁵³² Vgl. z.B. Hegering et al. (1999).

⁵³³ Vgl. Mayerl et al. (2005).

⁵³⁴ Vgl. Dan et al. (2004), S. 136.

len.⁵³⁵ Nach ITIL können für verschiedene beteiligte Partner neben SLAs weitere vertragliche Vereinbarungen, wie Operational Level Agreement (OLA) und Underpinning Contract, unterschieden werden.

IT-Service		
Definition	Ein IT-Service stellt eine Menge zusammengehörender Dienstleistungen dar, die von einem IT-System oder der IT-Abteilung bzw. einem externen IT-Dienstleister bereitgestellt werden, um die Geschäftsprozesse des Unternehmens zu unterstützen. ⁵³⁶	
Beziehungen zu	Infrastrukturelement	Ein IT-Service umfasst ein oder mehrere Infrastrukturelement/e.
	IT-Dienstleister	Ein IT-Service wird von einem oder mehreren IT-Dienstleister/n erbracht/unterstützt.
	IT-Service-Prozess	Ein IT-Service wird von einem oder mehreren IT-Service-Prozess/en erzeugt/verwendet.
	Marktleistung	Ein IT-Service kann Teil einer oder mehrerer Marktleistung/en sein.
	Organisationseinheit	Ein IT-Service wird von einer Organisationseinheit verantwortet/kontrolliert.
	Service Level Agreement (SLA)	Ein IT-Service wird von einem SLA definiert.
	Softwarekomponente	Ein IT-Service umfasst eine oder mehrere Softwarekomponent/en.
IT-Service-Prozess		
Definition	Ein IT-Service-Prozess beschreibt einen Betriebsprozess innerhalb der IT-Abteilung bzw. des IT-Dienstleisters zur Erzeugung bestimmter Zusatzdienste eines IT-Services.	
Beziehungen zu	IT-Service	Ein IT-Service-Prozess erzeugt/verwendet einen oder mehrere IT-Service/s.
	IT-Dienstleister	Ein IT-Service-Prozess wird von einem externen IT-Dienstleister bzw. der internen IT-Abteilung durchgeführt.
Servicekatalog		
Definition	Ein Servicekatalog erfasst die von einem IT-Dienstleister angebotenen IT-Services mit entsprechenden Kostenangaben. ⁵³⁷	
Beziehungen zu	IT-Service	Ein Servicekatalog umfasst mehrere IT-Services.
	Organisationseinheit	Ein Servicekatalog wird von einer Organisationseinheit verwaltet.
Service Level		
Definition	Ein Service Level beschreibt ein (Qualitäts-)Merkmal eines bereitgestellten IT-Services. ⁵³⁸	
Beziehungen zu	Service Level Agreement	Ein Service Level ist Teil eines Service Level Agreements.

⁵³⁵ Vgl. Mayerl et al. (2005).

⁵³⁶ Vgl. Commerce (2000).

⁵³⁷ Vgl. Mayerl et al. (2005).

⁵³⁸ Vgl. Dan et al. (2004), S. 84.

Service Level Agreement (SLA)		
Definition	Ein Service Level Agreement (SLA) ist eine schriftliche Vereinbarung zwischen dem Service-Lieferanten (externer IT-Dienstleister oder IT-Abteilung) und dem Kunden, die die für den IT-Service vereinbarten Service Levels dokumentiert. ⁵³⁹	
Beziehungen zu	IT-Service	Ein SLA definiert einen bestimmten IT-Service.
	Service Level	Ein SLA umfasst mehrere Service Levels.

Tabelle 16: Metaentitätstypen der IT-Servicesicht

Ein IT-Service wird durch einen IT-Service-Prozess erbracht und stellt folglich eine **Prozessleistung** im Sinne der bereits in der Ablaufplanung definierten Prozessleistung dar. Prozessleistungen sind das Ergebnis (der Output) der Leistungserstellung eines Prozesses, in diesem Fall eines IT-Service-Prozesses, das an interne oder externe Kunden geht. Empfänger einer Prozessleistung ist ein anderer Prozess innerhalb oder ausserhalb der Unternehmung. Ist der Leistungsempfänger ein Prozess ausserhalb der eigenen Unternehmung, ist eine Prozessleistung Teil einer Marktleistung.⁵⁴⁰ Somit können die von einer IT-Abteilung erbrachten IT-Services auch Teil von Marktleistungen eines Unternehmens sein, wenn beispielsweise die IT-Abteilung ihre Leistungen im Sinne eines Insourcings auch externen Unternehmen anbietet. IT-Service-Prozesse können dementsprechend auch nach Unterstützungsprozessen oder Leistungsprozessen differenziert werden. Ein IT-Service-Prozess stellt einen Leistungsprozess dar, wenn dessen IT-Leistungen Teil der Marktleistungen sind und somit direkt an externe Kunden gehen.

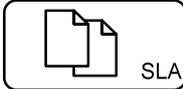
Symbole der IT-Servicesicht		
Infrastrukturelement	IT-Service	IT-Service-Prozess
 Infrastrukturelement	 IT-Service	 IT-Service-Prozess
Service Level Agreement (SLA)	definiert (von SLA zu IT-Service)	erzeugt/verwendet (von IT-Service-Prozess zu IT-Service)
		

Tabelle 17: Symbole der IT-Servicesicht

Grundsätzlich lässt sich die IT-Servicesicht der Informationssystemebene zuordnen, da Softwarekomponenten Teil von IT-Services sind. Bei der Definition der IT-Services spielt aber auch die Orientierung an den fachlichen Anforderungen der Geschäftsseite des Unternehmens

⁵³⁹ Vgl. Dan et al. (2004), S. 136.

⁵⁴⁰ Vgl. Brecht (2002), S. 170.

eine wichtige Rolle. Deshalb enthält die IT-Servicesicht einige wesentliche Verknüpfungen und Metaentitätstypen der anderen Ebenen und Sichten der Unternehmensarchitektur, so dass sie im Unterschied zu den zuvor definierten Sichten eher als ebenenübergreifende Sicht betrachtet werden kann. Im Wesentlichen lassen sich dabei die folgenden sechs Verknüpfungen der IT-Servicesicht mit den übrigen Sichten identifizieren, welche in Abbildung 54 überblicksartig dargestellt sind:

- (1) Eine grundlegende Beziehung besteht zwischen der IT-Servicesicht und der auf Organisationsebene definierten Prozesslandkarte. IT-Services stellen Prozessleistungen der IT-Service-Prozesse dar und dienen der Unterstützung von Prozessen der Prozesslandkarte. Darüber hinaus können die in den IT-Services enthaltenen IT-Service-Prozesse in einer eigenen IT-Prozesslandkarte des IT-Bereichs dokumentiert sein.

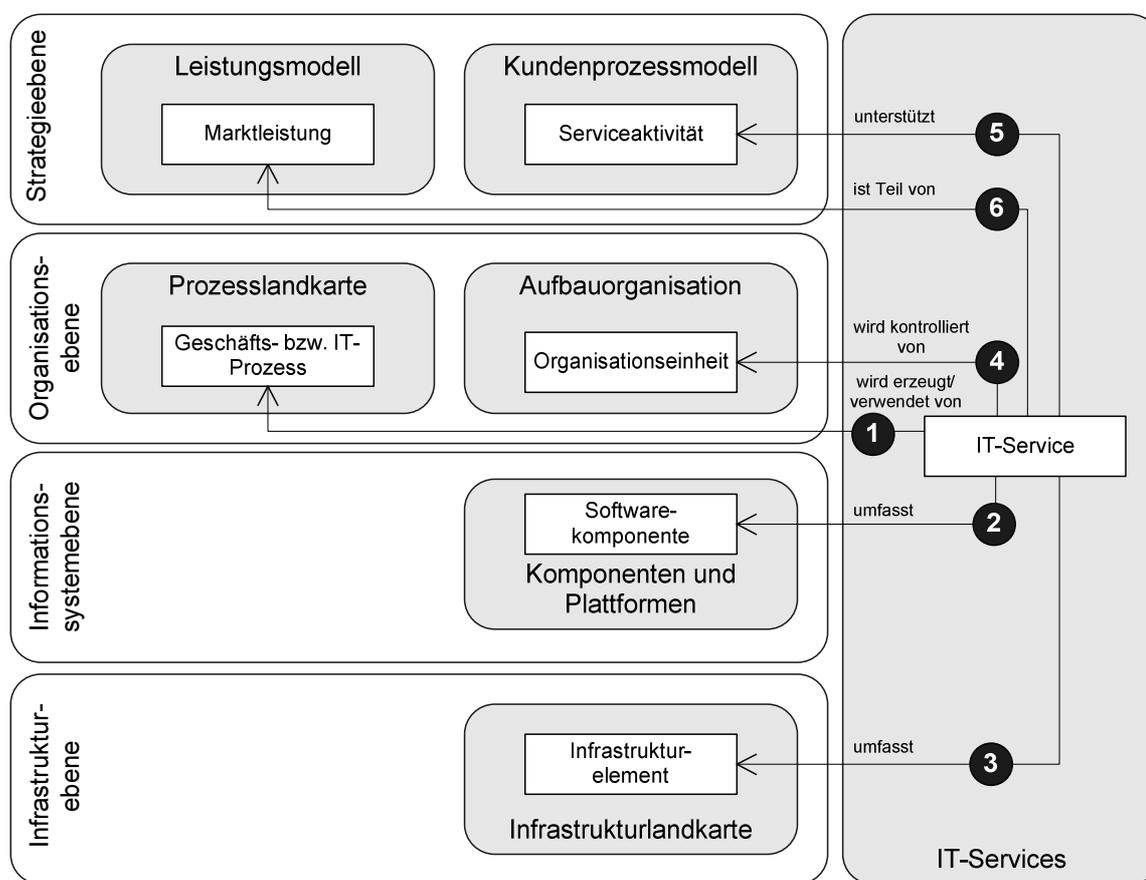


Abbildung 54: Integration der IT-Servicesicht

- (2) Eine weitere Abhängigkeit existiert zwischen dem Komponentenmodell und der IT-Servicesicht. Jeder IT-Service umfasst eine oder mehrere Softwarekomponenten, deren Funktionalitäten über Schnittstellen und mit entsprechenden Zusatzdiensten (IT-Service-Prozess) angeboten werden. Die von einer Softwarekomponente angebotenen Funktionalitäten können im Sinne einer serviceorientierten Architektur (SOA) ebenfalls als Services (Fachlicher Service) definiert werden, sofern sie gekapselt und ihre standardisierten

Schnittstellen in einem Serviceverzeichnis publiziert sind (vgl. Abbildung 55).⁵⁴¹ Dadurch können sie flexibel genutzt und wieder verwendet werden. Die eigentliche Implementierung der Komponente ist somit unabhängig von der Schnittstelle, so dass die Implementierung jederzeit geändert werden kann, ohne die Konsumenten (andere Softwarekomponenten) zu beeinflussen.

- (3) Die Technologie- und Infrastrukturebene ist im engeren Sinne nicht Betrachtungsgegenstand des Business Engineering.⁵⁴² Für die Definition und Integration einer IT-Servicesicht sind aber auch die eingesetzten Technologien und Infrastrukturelemente von Bedeutung, da sie wesentliche Bestandteile der IT-Services darstellen.
- (4) Wie bereits zuvor erwähnt wurde, ist jedem IT-Service analog zu den Geschäftsprozessen eine verantwortliche Organisationseinheit im Sinne eines Owner zuzuordnen. Dementsprechend steht die IT-Servicesicht auch mit der Aufbauorganisation des Unternehmens bzw. der Geschäftseinheit in Beziehung.
- (5) Des Weiteren können die zu einem IT-Service gehörenden IT-Service-Prozesse Serviceaktivitäten zur Unterstützung der Kundenaktivitäten im Kontext des Kundenprozessmodells darstellen bzw. diese unterstützen.
- (6) Die Einführung einer Serviceorientierung wirkt sich offensichtlich nicht nur auf die Organisation und die zugrunde liegenden Informationssysteme eines Unternehmens aus, sondern beeinflusst auch dessen Geschäftsstrategie. IT-Services können, wie zuvor bereits beschrieben, Teil von Marktleistungen sein und beziehen sich somit auf das Leistungsmodell eines Unternehmens oder einer Geschäftseinheit.

In Verbindung mit dem zuvor beschriebenen serviceorientierten IT-Management und der Gestaltung der Kooperation zwischen IT-Dienstleister und Dienstnehmer durch so genannte IT-Services wird auch das Konzept der serviceorientierten Architektur (SOA) sowohl in der Praxis als auch in der Wissenschaft stark diskutiert.⁵⁴³ Bisher liegt dem SOA-Konzept allerdings kein allgemein akzeptiertes Verständnis zugrunde. Ursprünglich stammt der Begriff SOA aus der Softwareentwicklung. Die W3C definiert beispielsweise SOA als eine Art mehrschichtige, verteilte Systemarchitektur, die aus einer Menge von gekapselten Funktionalitäten besteht, deren standardisierte Schnittstellen publiziert sind.⁵⁴⁴ Dadurch können sie in Softwaresystemen flexibel genutzt und gut wiederverwendet werden. Prinzipiell spielt es dabei keine Rolle, ob Services von internen Softwarekomponenten bereitgestellt oder extern bezogen werden.

⁵⁴¹ Vgl. hierzu z.B. *Winter/Schelp* (2005), S. 241 und *Schemm et al.* (2006), S. 24-30.

⁵⁴² Vgl. *Schelp/Schwinn* (2005), S. 1334.

⁵⁴³ Vgl. z.B. *Arsanjani* (2004), *Dan et al.* (2004), *W3C* (2004a), *Winter/Schelp* (2005), *Schemm et al.* (2006), *Böhmman et al.* (2002).

⁵⁴⁴ Vgl. *W3C* (2004a).

Wenn die Serviceorientierung tatsächlich so vorteilhaft ist, wie allgemein vermutet wird, dann sollte dieses Konzept nicht nur aus technischer Sicht, sondern auch auf den „oberen“ Ebenen der Unternehmensarchitektur aus fachlicher Sicht angewandt werden.⁵⁴⁵ Eine serviceorientierte Architektur wird deshalb in dieser Arbeit als eine mehrschichtige, verteilte Integrationsarchitektur verstanden, die Teile der Applikationsarchitektur als fachliche Services kapselt und das Bindeglied zwischen Organisations- und Informationssystemebene darstellt.⁵⁴⁶ Ein fachlicher Service stellt dabei ein abstraktes Software-Element bzw. eine Schnittstelle dar, die anderen Applikationen unter Nutzung von verbreiteten Standards stabile, wieder verwendbare Software-Funktionalität auf einer anwendungsorientierten, fachlichen Granularitätsstufe anbietet.⁵⁴⁷ Das SOA-Konzept wird dementsprechend in den Bezugsrahmen des Ansatzes der vorliegenden Arbeit als horizontale Servicesicht der Applikationsebene eingeordnet (vgl. Abbildung 32). Diese Servicesicht kapselt Fachfunktionalität von unterschiedlichen Applikationen, um diese anschliessend für die Unterstützung der Prozesse flexibel einsetzen und kombinieren zu können. Dadurch lässt sich der Übergang von der Organisationsebene zur Informationssystemebene flexibler gestalten und vereinfachen.

Da es nicht sinnvoll erscheint, die Funktionalität aller Applikationen eines Unternehmens in Services zu kapseln, und die Transformation zu einer serviceorientierten Architektur in der Regel einen längeren Zeitraum in Anspruch nimmt,⁵⁴⁸ wird das Konzept der Applikationen weiterhin parallel zu den fachlichen Services beibehalten. Folglich kann eine Geschäftsfunktion sowohl von einer oder mehreren Applikation(en) als auch von einem oder mehreren fachlichen Services unterstützt werden (vgl. Abbildung 55).

Darüber hinaus können fachliche Services Teil von IT-Services sein, wenn sie zusammen mit anderen Servicekomponenten, wie z.B. Infrastrukturelementen, und entsprechenden Zusatzdiensten angeboten werden.

⁵⁴⁵ Vgl. *Winter/Schelp* (2005), S. 46.

⁵⁴⁶ Vgl. *Schemm et al.* (2006), S. 24.

⁵⁴⁷ Vgl. *Schemm et al.* (2006), S. 29 und *W3C* (2004b).

⁵⁴⁸ Vgl. hierzu z.B. auch *Dan et al.* (2004), S. 84.

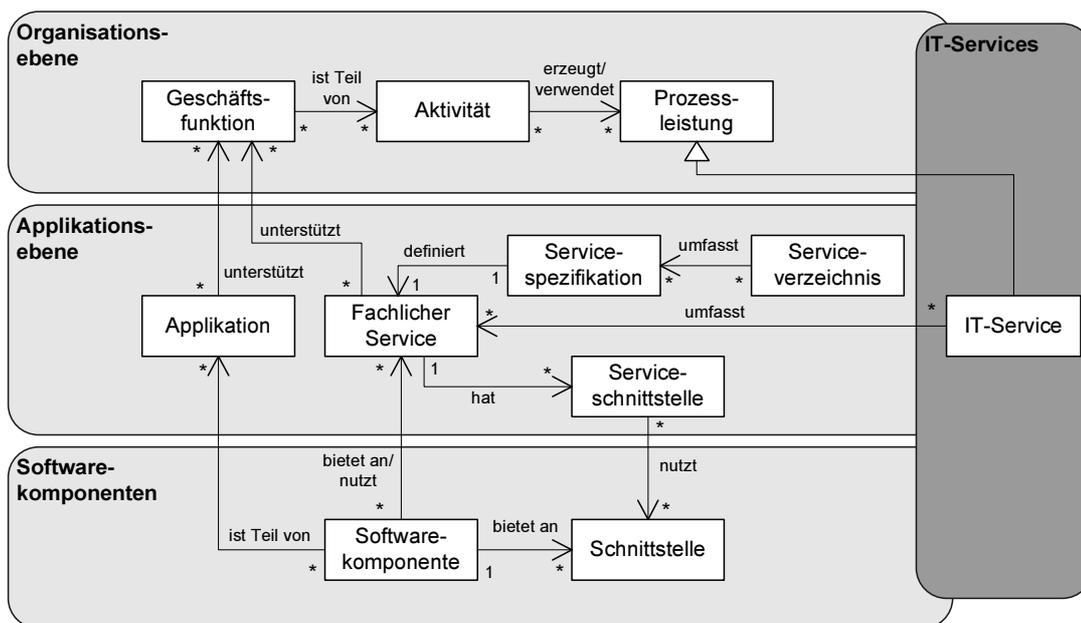


Abbildung 55: Metamodell der SOA-Sicht

Die wesentlichen Metaentitätstypen der SOA-Sicht sind in der folgenden Tabelle 18 genauer spezifiziert.

Fachlicher Service		
Definition	Ein Service im Zusammenhang mit einer serviceorientierten Architektur (Fachlicher Service) stellt ein abstraktes Software-Element bzw. eine Schnittstelle dar, die anderen Softwarekomponenten unter Nutzung von verbreiteten Standards stabile, wieder verwendbare Software-Funktionalität auf einer fachlichen Granularitätsstufe anbietet. ⁵⁴⁹	
Beziehungen zu	Softwarekomponente	Ein fachlicher Service wird von einer oder mehreren Softwarekomponente/n genutzt/angeboten.
	Servicespezifikation	Ein fachlicher Service wird von einer Servicespezifikation beschrieben.
	Serviceschnittstelle	Ein fachlicher Service besitzt eine oder mehrere Serviceschnittstelle/n.
	IT-Service	Ein fachlicher Service ist Teil eines oder mehrerer IT-Services.
	Geschäftsfunktion	Ein fachlicher Service unterstützt eine oder mehrere Geschäftsfunktion/en.
Servicespezifikation		
Definition	Eine Servicespezifikation enthält sämtliche Informationen, die ein Servicenutzer benötigt, ohne die interne Realisierung des Services zu kennen. ⁵⁵⁰	
Beziehungen zu	Serviceverzeichnis	Eine Servicespezifikation ist in einem oder mehreren Serviceverzeichnis/sen enthalten.
	Fachlicher Service	Eine Servicespezifikation definiert genau einen fachlichen Service.
Serviceverzeichnis		
Definition	In einem Serviceverzeichnis werden die verfügbaren fachlichen Services registriert und ihre Servicespezifikation abgelegt. Das ermöglicht dem Servicenutzer, einen geeigneten Service zu identifizieren und aufzurufen. ⁵⁵¹	

⁵⁴⁹ Vgl. Schemm et al. (2006), S. 29 und W3C (2004b).

⁵⁵⁰ Vgl. Stojanovic (2005), S. 40-41 und W3C (2004b).

Beziehungen zu	Servicespezifikation	Ein Serviceverzeichnis enthält eine oder mehrere Servicespezifikation/en.
Serviceschnittstelle		
Definition	Eine Serviceschnittstelle stellt die softwaretechnische Realisierung einer Servicespezifikation dar. Sie nutzt über Komponentenschnittstellen die (Fach-)Funktionalität von Softwarekomponenten. ⁵⁵²	
Beziehungen zu	Fachlicher Service	Eine Serviceschnittstelle wird von genau einem fachlichen Service angeboten.
	Schnittstelle (SW-Komponente)	Eine Serviceschnittstelle nutzt eine oder mehrere Schnittstelle/n einer Softwarekomponente.

Tabelle 18: Metaentitätstypen der SOA-Sicht

4.6 Ebeneninterne Abhängigkeiten zwischen Modellen

Wie bereits zu Beginn dieses Kapitels erwähnt wurde, wird das gesamte Metamodell durch die Darstellung einer grösseren Zahl an Metaentitätstypen sowie der Beziehungstypen rasch unübersichtlich. Aus diesem Grund wurde das Metamodell in problemorientierte Sichten bzw. Modelle unterteilt, welche thematisch zusammengehörende Metaentitätstypen zusammenfassen. Diese problemorientierten Sichten sind nicht disjunkt, und es bestehen zahlreiche Abhängigkeiten zwischen den Modellen auf einer Ebene sowie zwischen den Modellen auf unterschiedlichen Ebenen. Da die Definition der Abhängigkeiten für die Gewährleistung der modellübergreifenden Konsistenz und für die Werkzeugunterstützung im Hinblick auf die Implementierung von Mechanismen, wie beispielsweise Konsistenzprüfungen, Abhängigkeitsanalysen, Simulation etc. von besonderer Bedeutung ist, werden in diesem Abschnitt die Abhängigkeiten nochmals genauer betrachtet und definiert sowie die zentralen Metaentitätstypen identifiziert. Im Hinblick auf die zukünftige Formulierung von Gestaltungsrichtlinien für die Entwicklung der einzelnen Modelle der Unternehmensarchitektur liefern diese zentralen Metaentitätstypen wichtige Hinweise, welche Informationen über das Unternehmen insbesondere in der Anfangsphase eines Unternehmensarchitektur-Projektes erhoben werden sollten.

Auf jeder Ebene gibt es mindestens zwei zentrale Metaentitätstypen, die die unterschiedlichen Modelle einer Ebene zueinander in Beziehung setzen. Auf diese sowie deren Beziehungen zu den Metaentitätstypen der einzelnen Modelle einer Ebene wird im Folgenden genauer eingegangen.

4.6.1 Strategieebene

Die beiden zentralen Metaentitätstypen, die alle Modelle der Strategieebene zueinander in Beziehung setzen und folglich bei der Gestaltung der strategischen Ausrichtung eines Unter-

⁵⁵¹ Vgl. *Schemm et al.* (2006), S. 29.

⁵⁵² Vgl. *W3C* (2004b).

nehmens im Mittelpunkt stehen, sind die **Organisationseinheit**⁵⁵³ sowie die **Marktleistung**. Die folgende Nummerierung der Beziehungen korrespondiert mit der in Abbildung 56 verwendeten Nummerierung.

- (1) Eine Organisationseinheit (Unternehmen oder Geschäftseinheiten) nimmt aus strategischer Sicht eine bestimmte kompetenzbasierte Rolle im Geschäftsnetzwerk wahr.⁵⁵⁴
- (2) Abhängig von dieser kompetenzbasierten Rolle enthält das Leistungsmodell der betrachteten Organisationseinheit im Sinne eines Referenzmodells bzw. einer Typisierung entsprechende Dimensionen und Ausprägungen. Das Leistungsmodell spezifiziert nach HEINRICH, bezogen auf einen bestimmten Stichtag, den markt-, wertschöpfungs- und potenzialbezogenen Zustand einer Organisationseinheit (Unternehmen oder Geschäftseinheit), die selbständig am Markt agiert oder agieren könnte, anhand der Ausprägung bestimmter Dimensionen.⁵⁵⁵ Die Rolle der Organisationseinheit im Geschäftsnetzwerk wird somit durch die Dimensionen und Ausprägungen des Leistungsmodells beschrieben. Darüber hinaus bezieht bzw. erbringt jede am Markt agierende Organisationseinheit Marktleistungen von bzw. für Kooperationspartner im Geschäftsnetzwerk.⁵⁵⁶
- (3) Die Verbindung zum Zielmodell (BSC) wird durch die Metaentitätstypen Strategie und Massnahme hergestellt. Das Zielmodell bzw. die BSC unterstützt das Unternehmen bzw. die Geschäftseinheit dabei, ihre Strategie mit den kurz- bis mittelfristigen Massnahmen in Einklang zu bringen.⁵⁵⁷ Das Unternehmen bzw. jede Geschäftseinheit verfolgt eine bestimmte Strategie, die durch Erfolgsfaktoren und Kennzahlen konkretisiert wird. Da die BSC der Strategieumsetzung dienen soll, liegt der Schwerpunkt auf der Übersetzung der Vision und Strategie in operative Ziele und Massnahmen.⁵⁵⁸ Für die Durchführung bestimmter Massnahmen zur Erreichung der Ziele ist eine bestimmte Organisationseinheit verantwortlich.

⁵⁵³ Auf Strategieebene werden lediglich Unternehmen oder Geschäftseinheiten betrachtet. Diese werden bei der Modellierung der Aufbauorganisation auf Organisationsebene in kleinere Organisationseinheiten eingeteilt.

⁵⁵⁴ Vgl. *Winter* (2003d), S. 96-97.

⁵⁵⁵ Vgl. *Heinrich* (2000), S. 11.

⁵⁵⁶ Vgl. hierzu z.B. *Österle* (2003), S. 32-34.

⁵⁵⁷ Vgl. *Kaplan/Norton* (1996), S. 75.

⁵⁵⁸ Vgl. *Harengel* (2000), S. 81.

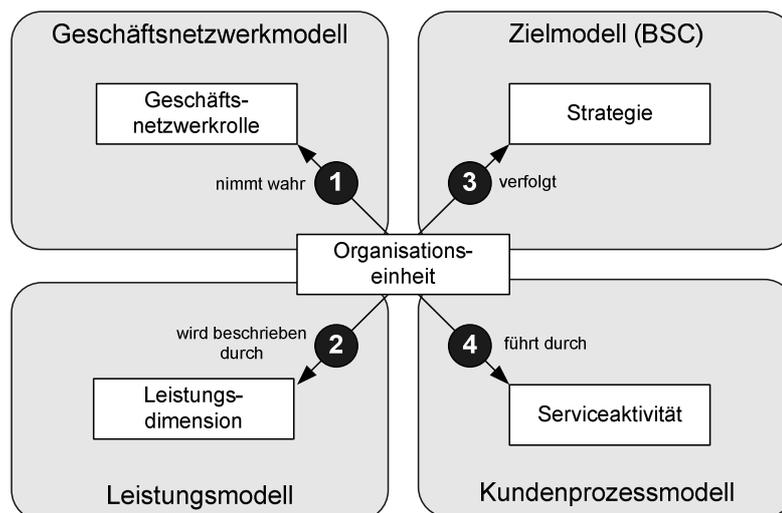


Abbildung 56: Verknüpfung der Modelle auf Strategieebene (OE)

- (4) Die Verknüpfung mit dem Kundenprozessmodell besteht über den Metaentitätstyp Serviceaktivität. Bei der Erstellung des Kundenprozessmodells ist für jede Serviceaktivität zu prüfen, ob sie auf Grundlage des Geschäftsmodells des jeweiligen Unternehmens bzw. der jeweiligen Geschäftseinheit selbst erbracht oder besser von einem Kooperationspartner fremdbezogen werden sollte.⁵⁵⁹ Folglich ist jeder Serviceaktivität des Serviceprozesses eine interne oder externe Organisationseinheit zugeordnet, die diese Serviceaktivität verantwortet und durchführt.

Neben der Organisationseinheit stellt auch die Marktleistung einen zentralen Metaentitätstyp bei der Gestaltung der Strategieebene dar (vgl. Abbildung 57).

- (1) Jede am Markt angebotene Leistung wird von einer oder mehreren Rollen im Geschäftsnetzwerk erbracht bzw. bezogen.
- (2) Die von einem Unternehmen oder einer Geschäftseinheit angebotene Marktleistung wird im Leistungsmodell durch eine oder mehrere Dimensionsausprägungen spezifiziert.

⁵⁵⁹ Vgl. Heinrich (2002b), S. 87.

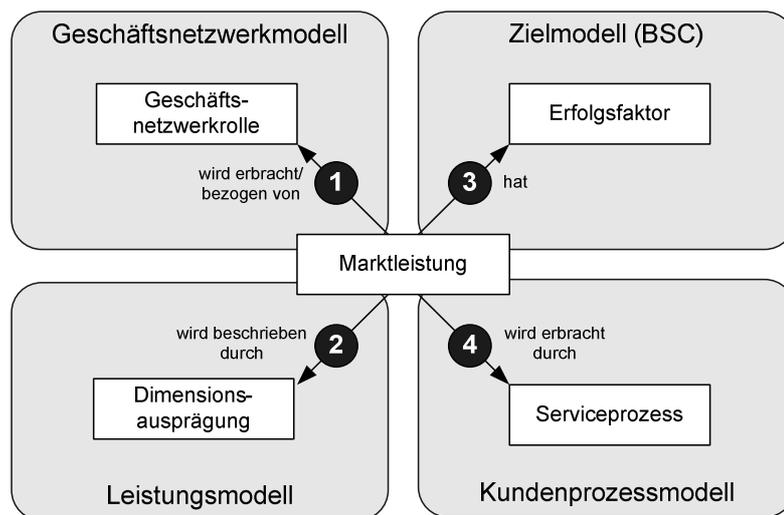


Abbildung 57: Verknüpfung der Modelle auf Strategieebene (Marktleistung)

- (3) Marktleistungen setzen sich aus Prozessleistungen zusammen, die von entsprechenden Geschäftsprozessen auf Organisationsebene erzeugt werden. Analog zur Definition der kritischen Erfolgsfaktoren und Ziele für Geschäftsprozesse lassen sich folglich auf Strategieebene Erfolgsfaktoren und Ziele für Marktleistungen auf Basis externer Anforderungen (Kundenanforderungen) ableiten und in der BSC als strategische Ziele der internen Geschäftsprozessperspektive abbilden (vgl. hierzu Abschnitt 4.1.2.4).
- (4) Die von einem Unternehmen im Geschäftsnetzwerk angebotene Leistung wird durch einen internen Serviceprozess oder durch einen externen Serviceprozess eines Partners erbracht.

Analog zur Strategieebene lassen sich auch auf Organisationsebene zentrale Metaentitätstypen identifizieren.

4.6.2 Organisationsebene

Auf der Organisationsebene sind der **Prozess**, die **Organisationseinheit** sowie die **Führungsgröße** die zentralen Metaentitätstypen, die die unterschiedlichen Modelle bzw. Sichten dieser Ebene miteinander verknüpfen. Die folgenden sechs Beziehungen werden in Abbildung 58 dargestellt.⁵⁶⁰

- (1) Der Metaentitätstyp Prozess ist zum einen ein Element der Prozesslandkarte, in der die wesentlichen Leistungs- und Informationsbeziehungen zwischen den Leistungs-, Unterstützungs- und Führungsprozessen des Unternehmens bzw. der Geschäftseinheit abge-

⁵⁶⁰ Vgl. hierzu auch jeweils das entsprechende Metamodell, mit dem eine Verknüpfung besteht.

- bildet werden. Jeder Prozess verwendet und/oder erzeugt keine, eine oder mehrere Prozessleistung(en).⁵⁶¹
- (2) Zudem knüpft er über den Metaentitätstyp Aktivität an das Modell der Ablauforganisation an. Jeder Prozess besteht aus einer Menge von Aktivitäten, die in der Ablauforganisation in eine bestimmte Reihenfolge gebracht und den entsprechenden Rollenträgern zugeordnet werden.⁵⁶²
 - (3) Eine Verknüpfung zur Aufbauorganisation stellt der Prozess einerseits indirekt über die Beziehungskette Aktivität, Prozessrolle und Rollenträger (vgl. Abbildung 43), andererseits direkt über die Beziehung zur Organisationseinheit her. Jede Aktivität eines Prozesses kann von einer Prozessrolle durchgeführt werden, die wiederum von einem bestimmten Rollenträger der Aufbauorganisation übernommen wird.⁵⁶³ Ausserdem ist jeder Prozess einer verantwortlichen Organisationseinheit zugeordnet.⁵⁶⁴ Dabei kann es sich um eine speziell für das Prozessmanagement eingerichtete Organisationseinheit handeln.
 - (4) Die Verbindung des Prozesses zur Prozessvision wird über die Prozessgrundsätze hergestellt, die beschreiben, wie der Prozess idealerweise gestaltet sein sollte. Jeder Prozess wird durch mehrere Prozessgrundsätze beschrieben.⁵⁶⁵
 - (5) Der Prozess ist sowohl über den Metaentitätstyp kritischer Erfolgsfaktor als auch mit dem Metaentitätstyp Messsystem mit der Prozessführung verknüpft. Jeder Prozess besitzt einen oder mehrere kritische Erfolgsfaktoren, die durch Führungsgrössen operationalisiert werden.⁵⁶⁶ Zudem kann einem Prozess ein Messsystem zugeordnet sein, das die für diesen Prozess definierten Führungsgrössen misst.
 - (6) Die Ergebnisse der Prozessführung werden in Form von Berichten dokumentiert und ausgewertet. Diese Berichte können anschliessend durch eine Inventur für die Erstellung der Informationslandkarte herangezogen werden.⁵⁶⁷

⁵⁶¹ Vgl. hierzu z.B. *Österle* (1995), S. 52.

⁵⁶² Vgl. hierzu z.B. *Scheer* (2001), S. 31.

⁵⁶³ Vgl. hierzu z.B. *Rosemann/zur Mühlen* (1997), S. 79.

⁵⁶⁴ Vgl. hierzu z.B. *Frank* (2002), S. 3028.

⁵⁶⁵ Vgl. hierzu z.B. *Brecht* (2002), S. 168.

⁵⁶⁶ Vgl. hierzu z.B. *Österle* (1995), S. 54.

⁵⁶⁷ Vgl. hierzu z.B. *Strauch* (2002), S. 139.

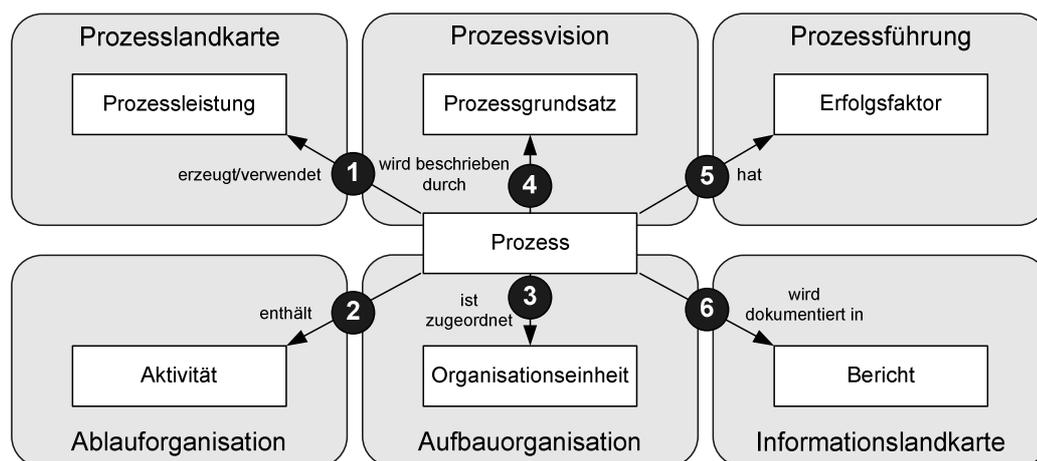


Abbildung 58: Verknüpfung der Modelle auf Organisationsebene (Prozess)

Einen weiteren zentralen Metaentitätstyp auf der Organisationsebene stellt die Organisationseinheit dar (vgl. Abbildung 59).⁵⁶⁸

- (1) Eine Organisationseinheit kann Owner eines oder mehrerer Prozesse(s) sein, die in der Prozesslandkarte abgebildet sind.⁵⁶⁹
- (2) An diese Organisationseinheit geht der Bericht der entsprechenden Prozessführung.⁵⁷⁰
- (3) Als wichtige Ideenquellen für neue Lösungen zur Prozessgestaltung und damit als Grundlage für die Prozessvision können unter anderem Mitarbeiter und Organisationseinheiten herangezogen werden. Ausserdem ist für die Entwicklung und Pflege der Prozessgrundsätze eine bestimmte Organisationseinheit, meist eine speziell für das Prozessmanagement eingerichtete, zuständig.⁵⁷¹
- (4) Wie bereits zuvor festgestellt, wird die Verbindung zwischen der Aufbauorganisation und Ablauforganisation über den Metaentitätstyp Prozessrolle hergestellt, die von einem bestimmten Rollenträger übernommen wird. Dabei kann es sich um eine Organisationseinheit handeln.
- (5) Die Organisationseinheit ist Teil der Aufbauorganisation und kann selbst wiederum mehrere Organisationseinheiten oder mehrere Stellen umfassen.⁵⁷²
- (6) Zuletzt ist die Organisationseinheit auch mit der Informationslandkarte verknüpft, da eine Organisationseinheit bzw. deren Stellen sowohl Konsument als auch Produzent von Informationen sein kann.⁵⁷³

⁵⁶⁸ Vgl. hierzu auch jeweils das entsprechende Metamodell, mit dem eine Verknüpfung besteht.

⁵⁶⁹ Vgl. hierzu z.B. *Frank* (2002), S. 3028.

⁵⁷⁰ Vgl. hierzu z.B. *Brecht* (2002), S. 148.

⁵⁷¹ Vgl. hierzu z.B. *Hess* (1996), S. 183-189.

⁵⁷² Vgl. hierzu z.B. Glossar in *IMG* (1997) und *Rosemann/zur Mühlen* (1997), S. 80.

⁵⁷³ Vgl. hierzu z.B. *Strauch* (2002), S. 181.

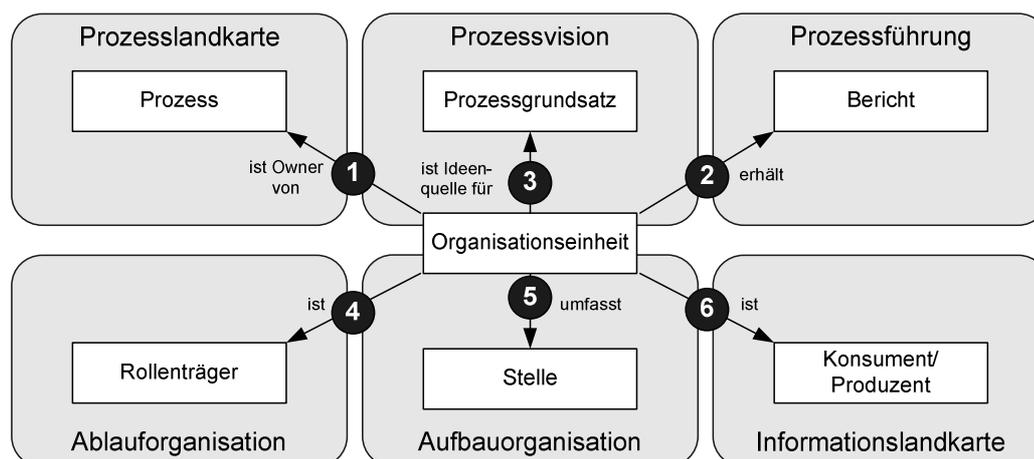


Abbildung 59: Verknüpfung der Modelle auf Organisationsebene (Organisationseinheit)

Neben den beiden zuvor genannten Metaentitätstypen ist auch die Führungsgrösse von zentraler Bedeutung für die Gestaltung der Organisationsebene (vgl. Abbildung 60).

- (1) Jedem Prozess der Prozesslandkarte sind bestimmte Führungsgrössen zugeordnet, die aus der Geschäftsstrategie abgeleitet wurden. Eine Führungsgrösse beschreibt einen quantitativ messbaren, für die Führung eines Prozesses relevanten Sachverhalt. Die einem Prozess zugeordneten Führungsgrössen dienen der Planung und Beurteilung der Prozessqualität im Sinne der kritischen Erfolgsfaktoren, insbesondere der Zeit, Qualität, Kosten und Flexibilität.⁵⁷⁴
- (2) Im Modell der Ablauforganisation können die für einen Prozess abgeleiteten Führungsgrössen konkreten Aktivitäten oder dem Output von Aktivitäten (Prozessleistungen) zugeordnet werden. Dadurch lassen sich sowohl die Effektivität als auch die Effizienz eines Prozesses steuern, planen und kontrollieren.
- (3) Die für einen Prozess definierten Führungsgrössen werden von der verantwortlichen Organisationseinheit kontrolliert. Diese setzt die Führungsgrössen zur Analyse des Prozesses und zur Kontrolle der Prozesszielerreichung ein, um Defizite aufzeigen zu können. Somit besteht auch über die Führungsgrössen eine Verbindung zur Aufbauorganisation.
- (4) Die Prozessvision enthält Ideen für die Neugestaltung eines Prozesses und bildet die Umriss des neuen Prozesses ab.⁵⁷⁵ Als Vorgabe für die Prozessgestaltung fungieren in der Prozessvision die Prozessgrundsätze. Die Gestaltungsmerkmale für die Prozessgrundsätze umfassen unter anderem auch die Erfolgsfaktoren des neu zu gestaltenden Prozesses. Diesen können neue oder bereits definierte Führungsgrössen zugeordnet werden.

⁵⁷⁴ Vgl. Ergebnisse in *IMG* (1997) und *Brecht* (2002), S. 155.

⁵⁷⁵ Vgl. Techniken in *IMG* (1997).

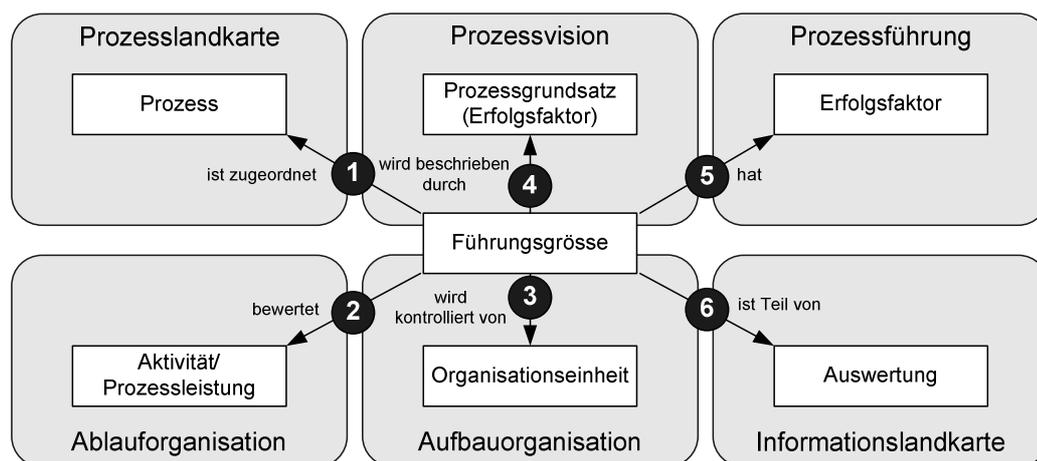


Abbildung 60: Verknüpfung der Modelle auf Organisationsebene (Führungsgrösse)

- (5) Eine Führungsgrösse operationalisiert einen oder mehrere Erfolgsfaktor(en) eines Prozesses und stellt somit das wesentliche Gestaltungselement der Prozessführung dar.
- (6) Die Werte der Führungsgrössen lassen sich aus unterschiedlichen betrieblichen Quellen bestimmen. Je nach Quelle ist zwischen den Ist-Werten und den Plan- oder Soll-Werten zu unterscheiden. Diese Werte werden in regelmässigen Abständen ausgewertet und beispielsweise in Form eines Berichts der verantwortlichen Organisationseinheit (Konsument) zur Verfügung gestellt. Folglich sind die Führungsgrössen auch Teil der Informationslandkarte.

4.6.3 Informationssystemebene

Auf der Informationssystemebene werden, ausgehend von den zuvor spezifizierten Organisationsstrukturen, Informationsbedarfen und Prozessabläufen, Integrationsbereiche innerhalb des Informationssystems festgelegt. Im Normalfall wird ein eng gekoppelter Integrationsbereich durch eine Applikation realisiert, während lose Kopplungen zwischen Integrationsbereichen durch Schnittstellen zwischen Applikationen realisiert werden.⁵⁷⁶ Das generelle Gestaltungsziel der Applikationsebene ist die optimale Integration (bzw. Entkopplung) von Applikationen aus fachlicher Sicht.

Die einzelnen Modelle der Informationssystemebene werden daher unter anderem durch den zentralen Metaentitätstyp **Applikation** miteinander verknüpft. Die folgenden sieben Beziehungen werden in Abbildung 61 dargestellt.⁵⁷⁷

- (1) Die einzelnen geplanten sowie realisierten Applikationen eines Unternehmens oder einer Geschäftseinheit werden in der Applikationsbestandsführung anhand vordefinierter Eigenschaften erfasst und in Applikationsdomänen gruppiert.⁵⁷⁸

⁵⁷⁶ Vgl. Winter (2003a), S. 9.

⁵⁷⁷ Vgl. hierzu auch jeweils das entsprechende Metamodell, mit dem eine Verknüpfung besteht.

- (2) Ferner manipuliert und/oder verwaltet jede Applikation Informationsobjekte, die im Informationsobjektemodell begrifflich harmonisiert und dokumentiert sind.⁵⁷⁹
- (3) Jede Applikation beinhaltet eine oder mehrere Applikationsfunktion(en), die wiederum eine oder mehrere Geschäftsfunktion(en) realisieren.⁵⁸⁰ Somit ist die Verbindung zum Geschäftsfunktionenmodell hergestellt.
- (4) Durch die Beziehungsstruktur zwischen Geschäftsfunktionen, Informationsobjekten und Organisationseinheiten werden bei der Definition der Integrationsarchitektur Integrationsbereiche abgeleitet, die anschliessend bestimmten neu zu entwickelnden Applikationen zugeordnet werden.⁵⁸¹
- (5) Die Verbindung zwischen der Applikation und dem Modell der Komponenten und Plattformen besteht über eine Kompositionsbeziehung mit dem Metaentitätstyp Softwarekomponente sowie mit den von einer Applikation manipulierten bzw. verwalteten Informationsobjekten, die als Datenobjekte auf der Ebene der Softwarekomponenten und Datenstrukturen implementiert sind.⁵⁸²

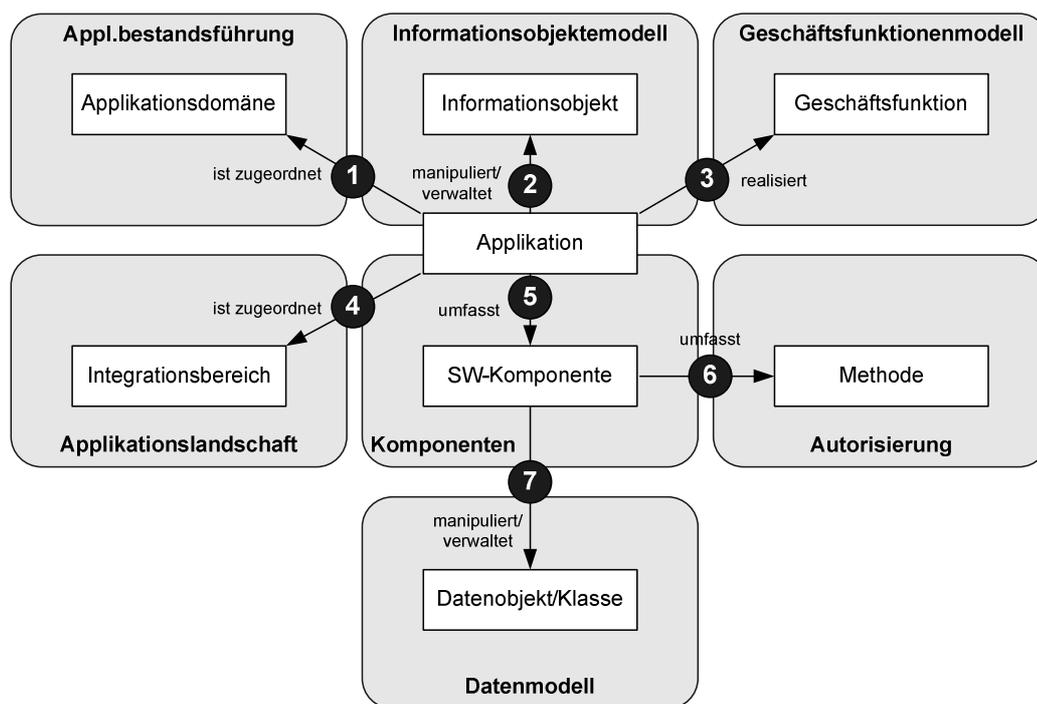


Abbildung 61: Verknüpfung der Modelle auf Informationssystemebene (Applikation)

- (6) Im Autorisierungsmodell werden die Berechtigungen für den Zugriff auf die Methoden und Datenobjekte der existierenden Softwarekomponenten und damit der Applikationen

⁵⁷⁸ Vgl. hierzu z.B. Choinowski et al. (2003), S. 72.

⁵⁷⁹ Vgl. hierzu z.B. Schwinn (2005), S. 138.

⁵⁸⁰ Vgl. hierzu z.B. Schwinn (2005), S. 20.

⁵⁸¹ Vgl. hierzu z.B. Winter (2003a).

⁵⁸² Vgl. hierzu z.B. Schwinn (2005), S. 130.

festgelegt.⁵⁸³ Somit ist auch die Verknüpfung der Applikation mit dem Autorisierungsmodell indirekt über deren Softwarekomponenten hergestellt.

- (7) Ebenso besteht eine indirekte Beziehung zum Datenmodell. Jede Applikation umfasst eine oder mehrere Softwarekomponenten, die auf Datenobjekten operieren und/oder diese verwalten.⁵⁸⁴ Jedes Datenobjekt korrespondiert mit genau einer Klasse des Datenmodells.

Die **Organisationseinheit** stellt auch auf der Informationssystemebene wiederum ein zentrales Verknüpfungselement dar (vgl. Abbildung 62). Hier tritt sie vor allem in der Rolle eines Owners für die unterschiedlichen Elemente der Applikations- sowie der Softwarekomponenten- und Datenstrukturebene auf. Aus diesem Grunde erscheint es sinnvoll, analog zur Definition des Metaentitätstyps Rollenträger (vgl. 4.2.2.4), einen zusätzlichen generalisierten Metaentitätstyp Owner zu definieren. Ein Owner könnte dann sowohl eine Organisationseinheit als auch einzelne Stellen, Personen oder Stellentypen repräsentieren, so dass eine flexiblere und detailliertere Zuordnung von Verantwortlichkeiten möglich ist.

- (1) Eine Organisationseinheit kann für eine oder mehrere Applikation(en), die in der Applikationsbestandsführung erfasst ist/sind, im Sinne eines Owner verantwortlich sein.⁵⁸⁵ Diese Verantwortlichkeit kann sowohl die (Weiter-)Entwicklung und Verwaltung der zugeordneten Applikation(en) als auch die Pflege bzw. Aktualisierung der Dokumentation der Applikation(en) in der Applikationsbestandsführung umfassen.

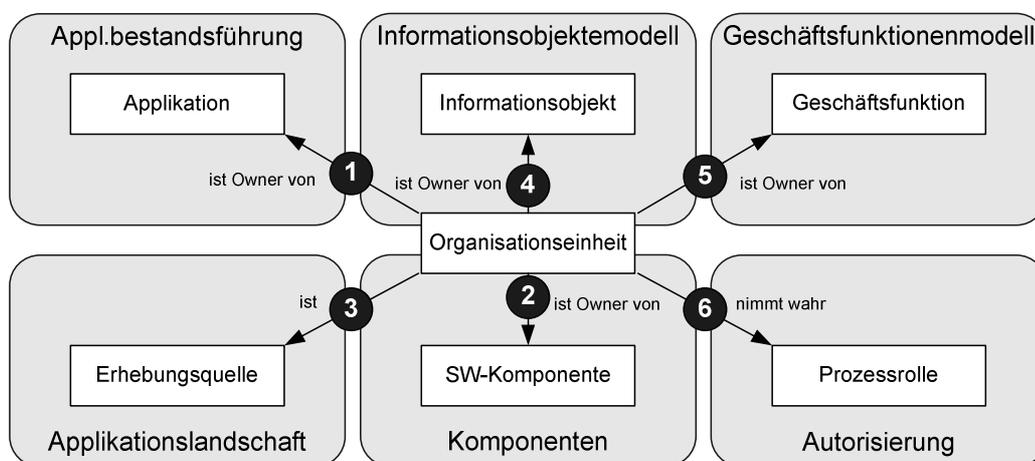


Abbildung 62: Verknüpfung der Modelle auf Informationssystemebene (OE)

- (2) Ebenso besitzt jede Softwarekomponente, jede Plattform, auf der die Softwarekomponenten laufen, sowie jeder Datenspeicher eine bestimmte Organisationseinheit als Owner, die

⁵⁸³ Vgl. hierzu z.B. Ferraiolo et al. (2001), S. 233.

⁵⁸⁴ Vgl. hierzu z.B. Schwinn (2005), S. 142.

⁵⁸⁵ Vgl. hierzu z.B. die Fallstudien in Schwinn (2005).

für die Entwicklung und Verwaltung zuständig ist.⁵⁸⁶ Somit knüpft die Organisationseinheit als Owner unterschiedlicher Elemente an das Modell der Komponenten und Plattformen an.

- (3) Eine Organisationseinheit kann bei der Herleitung der Integrationsbereiche der Integrationsarchitektur eine Erhebungsquelle von Begrifflichkeiten für Informationsobjekte sein.⁵⁸⁷
- (4) Analog zur Verantwortlichkeit für die Pflege und Aktualisierung der Einträge in der Applikationsbestandsführung müssen auch für die Pflege des Informationsobjekte- und
- (5) Geschäftsfunktionsmodells verantwortliche Organisationseinheiten (Owner) festgelegt werden.⁵⁸⁸
- (6) Die Verknüpfung der Organisationseinheit zum Autorisierungsmodell besteht indirekt über den Metaentitätstyp Prozessrolle. Damit die (organisatorischen) Rollenträger (Organisationseinheiten, Stellen, Personen) die informationstechnisch unterstützten Aktivitäten innerhalb eines Prozesses durchführen können, müssen sie über entsprechende Systemberechtigungen verfügen. Diese werden über Benutzerkonten den Rollenträgern zugeordnet und im Autorisierungsmodell dokumentiert.⁵⁸⁹

4.7 Ebenenübergreifende Abhängigkeiten

Nachdem in den vorherigen Abschnitten auf die ebeneninternen Zusammenhänge eingegangen wurde, sollen nun die Zusammenhänge zwischen Metaentitätstypen auf unterschiedlichen Ebenen näher betrachtet werden. Durch die Darstellung der Zusammenhänge lässt sich erkennen, welche Inhalte einer Ebene bei der Modellierung der darunter liegenden Ebene verwendet werden sollen bzw. müssen, wenn von einem Top-Down-Vorgehen ausgegangen wird. Die Identifikation und Spezifikation dieser Abhängigkeiten bildet somit eine wichtige Grundlage für die Beantwortung von Fragestellungen des IT-Business-Alignment.

Wie sich zuvor herausgestellt hat, ist die Organisationseinheit auf allen drei Ebenen ein zentraler Metaentitätstyp. Deshalb erscheint es sinnvoll, die Sicht der Aufbauorganisation nicht nur der Organisationsebene zuzuordnen, sondern orthogonal zu den anderen Ebenen, also ebenenübergreifend zu definieren. Auf Strategieebene wird dann lediglich auf Unternehmen oder Geschäftseinheiten in der Aufbauorganisation verwiesen, auf der Organisations- und Systemebene auf Organisationseinheiten sowie einzelne Stellen oder Personen.

⁵⁸⁶ Vgl. hierzu z.B. *Siegenthaler/Schwinn* (2005), S. 222.

⁵⁸⁷ Vgl. hierzu z.B. *Choinowski et al.* (2003), S. 78.

⁵⁸⁸ Vgl. hierzu z.B. *Choinowski et al.* (2003), S. 78.

⁵⁸⁹ Vgl. hierzu z.B. *Wortmann* (2005), S. 123.

Die Elemente der Aufbauorganisation treten dabei oftmals als Owner unterschiedlicher Elemente der Unternehmensarchitektur auf unterschiedlichen Ebenen auf, so beispielsweise als Owner von Serviceaktivitäten auf Strategieebene, Prozessen auf Organisationsebene, Applikationen bzw. Applikationsdomänen auf Applikationsebene sowie Softwarekomponenten, Plattformen und Datenspeichern auf der Ebene der Softwarekomponenten und Datenstrukturen.

4.7.1 Zusammenhänge zwischen Strategie- und Organisationsebene

Abbildung 63 zeigt vereinfacht die wesentlichen Zusammenhänge zwischen Metaentitätstypen auf der Strategie- und Organisationsebene. Aus Gründen der Übersichtlichkeit sind die meisten ebeneninternen Beziehungen nicht dargestellt. Die folgende Nummerierung der Beziehungen korrespondiert mit der in Abbildung 63 verwendeten Nummerierung.

- (1) Eine Organisationseinheit (Unternehmen oder Geschäftseinheit) nimmt aus strategischer Sicht eine bestimmte Rolle im Geschäftsnetzwerk wahr. Bei der Modellierung der Prozesslandkarte auf Prozessebene müssen anschliessend die Geschäftsprozesse bestimmten Unternehmen bzw. Geschäftseinheiten zugeordnet werden, um einen reibungslosen Ablauf gewährleisten zu können. Dabei kann es sich um das eigene Unternehmen bzw. interne Geschäftseinheiten oder andere Unternehmen bzw. externe Geschäftseinheiten (z.B. im Sinne eines Outsourcings von Geschäftsprozessen⁵⁹⁰) handeln. Heutige Unternehmen bzw. Organisationen besitzen in der Regel eine prozessorientierte Organisationsform, so dass die Organisationseinheiten mit den Geschäftsprozessen korrespondieren bzw. jede Organisationseinheit für einen oder mehrere Geschäftsprozesse verantwortlich ist.⁵⁹¹
- (2) Darüber hinaus umfasst jede Organisationseinheit (Unternehmen oder Geschäftseinheit) auf Strategieebene eine oder mehrere Organisationseinheiten auf Organisationsebene, die wiederum Organisationseinheiten umfassen können oder letztendlich aus einzelnen Stellen und Personen bestehen.⁵⁹²
- (3) Das Zielsystem (Balanced Score Card) liefert Vorgaben in Form von Erfolgsfaktoren und Kennzahlen, die bei der Gestaltung der Prozesse sowie der Prozessführung berücksichtigt werden müssen und zur Einhaltung der Prozessziele beitragen sollen. Die im Zielsystem definierten kritischen Erfolgsfaktoren repräsentieren die für den Erfolg bzw. die für die Erreichung der Unternehmensziele entscheidenden Merkmale der Geschäftsstrategie.⁵⁹³ Analog kann ein kritischer Erfolgsfaktor eines Prozesses als eines der wenigen, erfolgs-

⁵⁹⁰ Zum Outsourcing von Geschäftsprozessen (Business Process Outsourcing) siehe z.B. *Braun/Winter* (2005a).

⁵⁹¹ Vgl. hierzu z.B. *Winter* (2003a).

⁵⁹² Vgl. hierzu z.B. *Frank* (2002), S. 3028.

⁵⁹³ Vgl. hierzu z.B. *Rockart* (1979), S. 85.

entscheidenden Merkmale eines Prozesses definiert werden.⁵⁹⁴ Um eine Ausrichtung der Prozesse an der Geschäftsstrategie und den Unternehmenszielen zu erreichen, erscheint es sinnvoll, dass ein kritischer Erfolgsfaktor eines Prozesses aus einem oder mehreren kritischen Erfolgsfaktor(en) des Zielsystems der Strategie abgeleitet und durch entsprechende Führungsgrößen operationalisiert wird. Die Erfolgsfaktoren und Führungsgrößen der Prozessführung korrespondieren somit mit den kritischen Erfolgsfaktoren und Kennzahlen des Zielsystems.

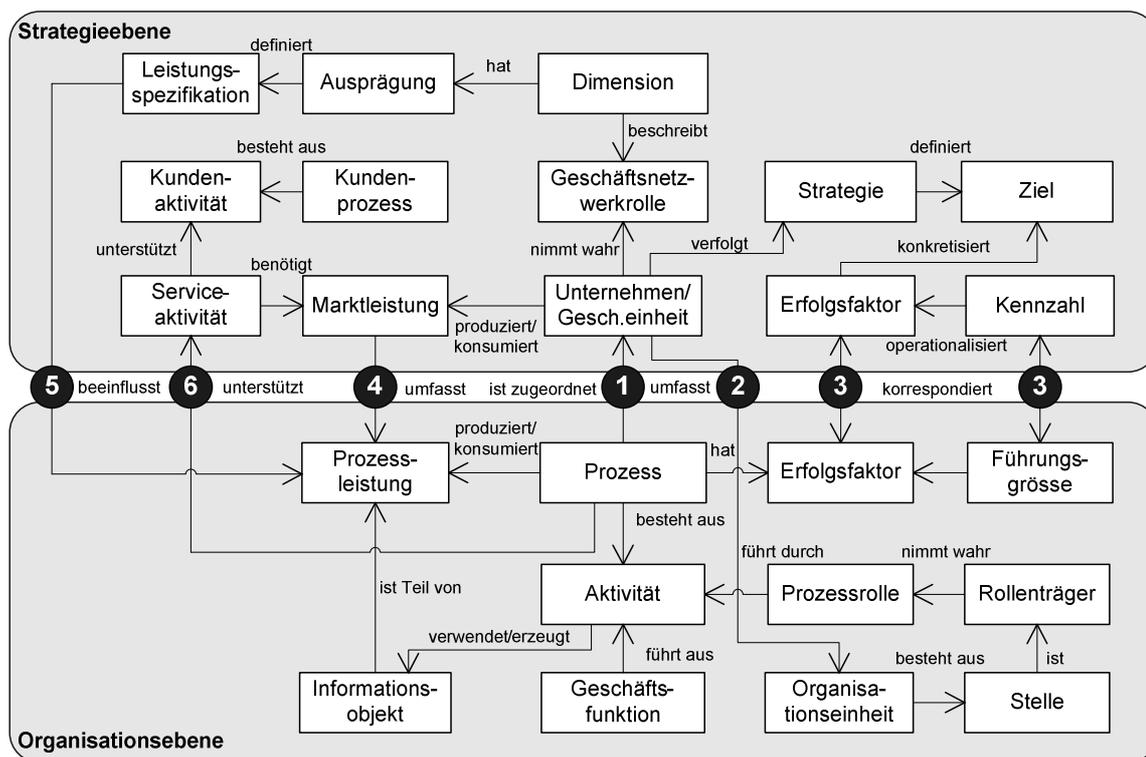


Abbildung 63: Zusammenhänge zwischen Strategie- und Organisationsebene

- (4) Das Prozessverständnis dieser Arbeit ist dadurch charakterisiert, dass ein Prozess seine Leistungen nicht an organisatorische Einheiten, sondern ausschliesslich an andere Prozesse innerhalb oder ausserhalb des eigenen Unternehmens bzw. der eigenen Geschäftseinheit übergibt (Kunden-Lieferanten-Verhältnis zwischen Prozessen).⁵⁹⁵ Prozessleistungen, die an Prozesse von Marktpartnern gehen, sind Teil der Marktleistungen einer Unternehmung.⁵⁹⁶ Marktpartner sind entweder Kooperationspartner im Geschäftsnetzwerk (andere Unternehmen oder Geschäftseinheiten) oder Endkunden. Folglich kann eine Marktleistung eine oder mehrere Prozessleistung(en) der Organisationsebene umfassen.
- (5) Die Ausprägungen der Dimensionen des Leistungsmodells liefern die Leistungsspezifikation. Diese steht in engem Zusammenhang mit der von einem Prozess erzeugten Prozess-

⁵⁹⁴ Vgl. hierzu z.B. *Leidecker/Bruno* (1984), S. 24.

⁵⁹⁵ Vgl. hierzu z.B. *Österle* (1995) und *Brecht* (2002).

⁵⁹⁶ Vgl. *Nieschlag et al.* (1994), S. 234-252.

leistung, welche in der Ablaufplanung auf Prozessebene abgebildet wird, da es beispielsweise für die Prozessmodellierung massgeblich ist, welche Leistungen das Unternehmen anbieten will bzw. kann (z.B. ob das Kerngeschäft einer Bank das Kredit- oder Anlagegeschäft ist).⁵⁹⁷ Die Leistungsspezifikation hat somit unmittelbaren Einfluss auf die Identifizierung der Prozesse, zumindest der Leistungsprozesse der zugrunde liegenden Unternehmung sowie der von ihnen erzeugten Prozessleistungen. In der Regel werden die Leistungsprozesse anhand der Leistungen, die ein Unternehmen anbietet, definiert.⁵⁹⁸

- (6) Das Kundenprozessmodell legt fest, welche Serviceaktivitäten durchgeführt werden müssen, um die einzelnen Kundenaktivitäten eines Kundenprozesses zu unterstützen. Da die Serviceaktivität zum Ausdruck bringt, welche Aktivitäten das Unternehmen bzw. die Geschäftseinheit im Kundenprozess unterstützen will/kann, werden Vorgaben für die zu modellierenden Prozesse bzw. deren Aktivitäten zur Leistungserbringung aufgestellt. Darüber hinaus muss für jede Serviceaktivität festgelegt werden, welche Prozesse diese unterstützen sollen. FRANK definiert beispielsweise in seiner Arbeit eine ähnliche Beziehung zwischen Aktivitäten der Wertschöpfungskette und Prozesstypen auf Organisationsebene.⁵⁹⁹

Eine weitere Abhängigkeit, die aus Gründen der Übersichtlichkeit nicht in Abbildung 63 dargestellt ist, besteht zwischen der Strategie eines Unternehmens bzw. einer Geschäftseinheit und den auf Organisationsebene festgelegten Prozessgrundsätzen. Diese werden unter anderem aus der Strategie abgeleitet.⁶⁰⁰ Somit hat die Definition der Strategie und Ziele unmittelbaren Einfluss auf die Gestaltung der Geschäftsprozesse.

4.7.2 Zusammenhänge zwischen Organisations- und Informationssystemebene

Abbildung 64 zeigt die wesentlichen Zusammenhänge zwischen Metaentitätstypen der Organisations- und Applikationsebene. Die folgende Nummerierung der Beziehungen korrespondiert wiederum mit der in Abbildung 64 verwendeten Nummerierung.

- (1) Jede Geschäftsfunktion und jedes Informationsobjekt auf Organisationsebene besitzt eine oder mehrere Begrifflichkeiten, die aus unterschiedlichen Erhebungsquellen stammen können. Dabei können homonyme und synonyme Begriffe unterschieden werden. Diese werden in einem Homogenisierungsschritt harmonisiert und somit auf einen Standardbegriff festgelegt. Die bezüglich ihrer Begriffe harmonisierten Geschäftsfunktionen und In-

⁵⁹⁷ Vgl. *Choinowski et al.* (2003), S. 36.

⁵⁹⁸ Vgl. *Jonkers et al.* (2004), S. 268.

⁵⁹⁹ Vgl. *Frank* (2002), S. 3028.

⁶⁰⁰ Vgl. hierzu z.B. *Techniken in IMG* (1997).

formationsobjekte dienen als Grundlage für die Ableitung neuer Applikationen bei der nachfolgenden Gestaltung der Integrationsarchitektur.⁶⁰¹

- (2) Als Erhebungsquelle der Begrifflichkeiten von Geschäftsfunktionen bzw. Informationsobjekten kommen die Beschreibungen von Organisationseinheiten sowie Geschäftsprozessen in Betracht.⁶⁰²
- (3) Die unterschiedlichen Informations- und Leistungsflüsse zwischen den harmonisierten Informationsobjekten, Geschäftsfunktionen sowie Geschäftsprozessen bzw. Organisationseinheiten⁶⁰³ definieren eine Beziehungsstruktur. Anhand dieser Beziehungsstruktur können Integrationsbereiche identifiziert werden. Diesen Integrationsbereichen wird anschliessend eine bereits existierende oder neu zu entwickelnde Applikation zugeordnet.
- (4) Eine Applikationsfunktion ist Bestandteil einer Applikation und realisiert eine oder mehrere Geschäftsfunktionen auf Organisationsebene, die bestimmte Aktivitäten von Prozessen ausführen. Darüber hinaus benötigen Applikationsfunktionen als Eingabe Informationsobjekte und/oder erzeugen als Ausgabe neue oder veränderte Informationsobjekte. Im Gegensatz zur Definition von Gutzwiller⁶⁰⁴ geht SCHWINN in seiner Arbeit nicht davon aus, dass jede Funktion von genau einer Applikationsfunktion realisiert wird, sondern dass gleiche Funktionen auch von mehreren Applikationsfunktionen realisiert werden können.⁶⁰⁵ Diese Annahme wird auch in dieser Arbeit übernommen, da dadurch die häufig in der Realität existierenden funktionalen Redundanzen berücksichtigt werden können. Durch die Dokumentation dieser Redundanzen wird gleichzeitig eine Grundlage für zukünftige Projekte der Applikationsintegration geschaffen.
- (5) Jede Applikation ist einer bestimmten Applikationsdomäne (Gruppierung mehrerer Applikationen) zugeordnet, die der Unterstützung eines Geschäftsprozesses dient. Eine Applikationsdomäne fasst mehrere Applikationen, die für die Durchführung eines Prozesses notwendig sind, logisch zusammen.⁶⁰⁶ Applikationsdomänen dienen vor allem der Komplexitätsreduktion. Jeder Applikationsdomäne ist zudem ein Owner (Verantwortlicher) auf Organisationsebene zugeordnet. Dabei kann es sich um eine Organisationseinheit, eine Stelle oder eine einzelne Person handeln.

⁶⁰¹ Vgl. hierzu z.B. *Winter* (2003a).

⁶⁰² Vgl. *Choinowski et al.* (2003), S. 78.

⁶⁰³ Bei einer geschäftsprozessorientierten Organisationsform können die Dimensionen Geschäftsprozesse, Organisationseinheiten und Leistungen zu einer Dimension zusammengefasst werden. Vgl. *Winter* (2003a), S. 5.

⁶⁰⁴ Vgl. *Gutzwiller* (1994), S. 64.

⁶⁰⁵ Vgl. *Schwinn* (2005), S. 19.

⁶⁰⁶ Vgl. hierzu z.B. *Schwinn* (2005), S. 74., *Lankes et al.* (2005), S. 1445 und *Jung* (2004), S. 313. Synonym werden oftmals die Begriffe „logische Einheit“ oder „Building Block“ verwendet.

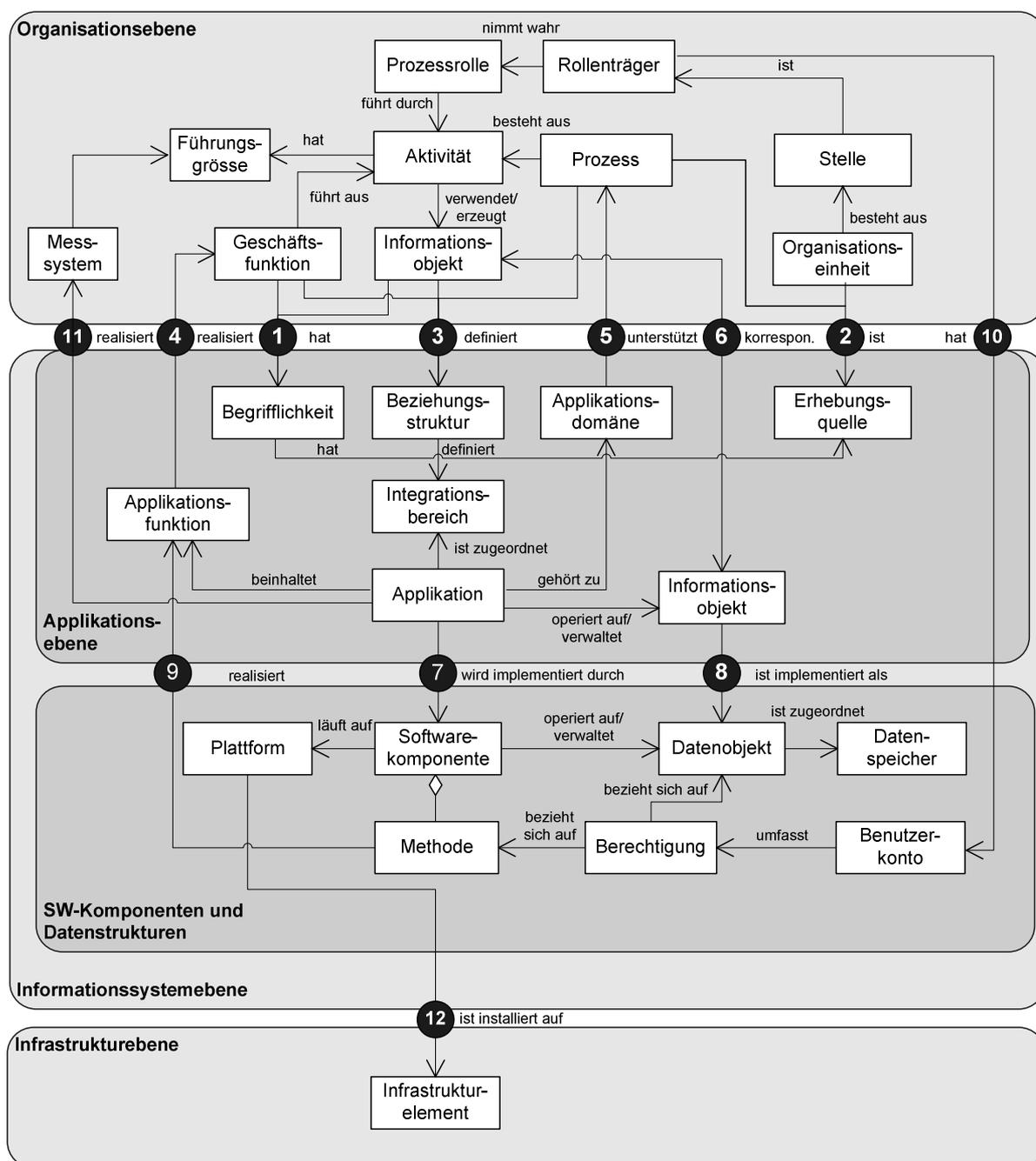


Abbildung 64: Zusammenhänge zwischen Organisations- und Informationssystemebene

(6) Analog zur Unterscheidung zwischen einer Geschäftsfunktion auf Organisationsebene und einer Applikationsfunktion auf Applikationsebene könnte auch zwischen einem Geschäftsobjekt auf Organisationsebene und einem Informationsobjekt auf Applikationsebene unterschieden werden. Da Informationsobjekte allerdings über Funktionen ausgetauscht werden und in der Regel mit den Geschäftsobjekten auf Organisationsebene korrespondieren, wird diese Differenzierung nicht vorgenommen. Stattdessen wird der Metaentitätstyp „Informationsobjekt“ sowohl auf Organisations- als auch auf Applikationsebene verwendet. Auf Organisationsebene werden dann lediglich die von Prozessen er-

zeugten und verwendeten Informationsobjekte erfasst. Auf Applikationsebene können zusätzlich Informationsobjekte, die beispielsweise in den Beschreibungen von Organisationseinheiten oder Stellen vorkommen, erfasst werden. Zudem besteht eine Beziehung zwischen Informationsobjekt und Prozessleistung, da ein oder mehrere Informationsobjekt(e) Teil einer Prozessleistung sein können, sofern es sich dabei um eine Informationsdienstleistung handelt.⁶⁰⁷

- (7) Die Verbindung zur Softwarekomponenten- und Datenstrukturebene besteht über die Metaentitätstypen „Softwarekomponente“ und „Datenobjekt“. Jede Applikation wird durch Softwarekomponenten implementiert, die auf Datenobjekten operieren und diese verwalten. Die in einer Softwarekomponente enthaltene Funktionalität wird in Form von Methoden bereitgestellt.⁶⁰⁸ Im Konzept der Softwarekomponenten ist zudem berücksichtigt, dass die Entwicklung von Applikationen zunehmend durch vorgefertigte, vertikale und horizontale (Standard-)Komponenten bewerkstelligt wird.⁶⁰⁹ Die Applikationsentwicklung wird in Zukunft nach Ansicht einiger Autoren zum Grossteil aus der Zusammenstellung von sofort einsatzfertigen Komponenten sowie deren Anpassung und Erweiterung bestehen.⁶¹⁰ Neue Funktionalitäten werden demzufolge vor allem durch vorgefertigte, sofort einsatzbereite Komponenten von unterschiedlichen Lieferanten zur Verfügung gestellt werden. Im Hinblick auf die ideale Implementierung von Applikationen und die dafür optimale Wiederverwendung von Softwarekomponenten spielt folglich die Dokumentation der Applikationen und Softwarekomponenten sowie von deren Lieferanten und Herstellern eine wichtige Rolle innerhalb der gesamten Unternehmensarchitektur.
- (8) Analog zu der zuvor beschriebenen Beziehung zwischen Applikation und Softwarekomponente ist jedes Informationsobjekt auf der Softwarekomponenten- und Datenstrukturebene als Datenobjekt implementiert, welches einem bestimmten Datenspeicher zugeordnet ist.⁶¹¹
- (9) Ausserdem wird entsprechend der Unterscheidung zwischen Informationsobjekt auf Applikationsebene und Datenobjekt auf Softwarekomponenten- und Datenstrukturebene eine Applikationsfunktion als Methode realisiert.⁶¹²
- (10) Damit die organisatorischen (menschlichen) Rollenträger die informationstechnisch unterstützten Aktivitäten eines Geschäftsprozesses durchführen können, müssen sie über entsprechende Systemberechtigungen verfügen.⁶¹³ Diese Berechtigungen werden auf der

⁶⁰⁷ Vgl. hierzu z.B. Vgl. *Scheer* (2001), S. 93 und Vgl. *Scheer* (2001), S. 67.

⁶⁰⁸ Vgl. *Schwinn* (2005), S. 22 und *Vogler* (2003), S. 31.

⁶⁰⁹ Vgl. *Greenfield/Short* (2003), S. 18.

⁶¹⁰ Vgl. z.B. *Cheesman/Daniels* (2000).

⁶¹¹ Vgl. hierzu z.B. *Schelp/Schwinn* (2005), S. 1335.

⁶¹² Vgl. hierzu z.B. *Schelp/Schwinn* (2005), S. 1335.

⁶¹³ Vgl. *Wortmann* (2005), S. 123.

Softwarekomponenten- und Datenstrukturebene über Benutzerkonten den Rollenträgern auf Organisationsebene zugeordnet. Dadurch erhalten diese Zugriff auf die Methoden und Datenobjekte der vorhandenen Softwarekomponenten.

(11) Jedem informationstechnisch unterstützten Prozess auf Organisationsebene kann ein entsprechendes Messsystem zugeordnet werden, das dessen Führungsgrößen misst und daraus in regelmässigen Abständen einen elektronischen Bericht erstellt.⁶¹⁴ Realisiert wird dieses Messsystem auf Applikationsebene durch eine oder mehrere Applikation(en).

(12) Ein Anknüpfungspunkt zur Infrastruktur- und Technologieebene besteht über den Metaentitätstyp „Infrastrukturelement“. Jede Plattform sowie jede Softwarekomponente ist auf einem bestimmten Infrastrukturelement (z.B. Server) installiert.

Aus Gründen der Übersichtlichkeit ist der Metaentitätstyp „Organisationseinheit“ in Abbildung 64 nicht auf der Applikations- sowie auf der Softwarekomponenten- und Datenstrukturebene eingezeichnet. Organisationseinheiten sowie deren Stellen und Personen können auf diesen beiden Ebenen Verantwortlichkeiten (Ownership) für Applikationsdomänen, einzelne Applikationen oder Softwarekomponenten, Informationsobjekte sowie Datenspeicher und Plattformen zugeordnet werden.

4.8 Detaillierungsgrad und Schnittstellen zu domänenspezifischen Architekturmodellen und Modellierungssprachen

Im Verständnis der vorliegenden Arbeit umfasst die Unternehmensarchitektur alle Schlüsselemente des Unternehmens in aggregierter Form. Für die Durchführung von Impact-Analysen und die Beantwortung von Fragestellungen des IT-Business-Alignment ist eine umfassende Abbildung aller Schlüsselemente in aggregierter Form sowie ihrer Beziehungen nützlicher als eine detaillierte Spezifizierung weniger Elemente einer bestimmten Anwendungsdomäne.

Gerade bei der Abbildung der Gesamtzusammenhänge würde die Verwendung einer detaillierten Modellierungssprache, wie beispielsweise der UML, zahlreiche „kreative“ Hilfskonstruktionen erfordern, um die einzelnen auf unterschiedlichen Ebenen definierten Modelle miteinander integrieren zu können. Zudem würden die auf diese Weise erstellten Modelle einen höheren Detaillierungsgrad aufweisen und wären für Mitarbeiter aus anderen Fachabteilungen nur schwer verständlich. Die detaillierte Abbildung bestimmter Aspekte sollte deshalb nicht im Rahmen der Unternehmensarchitektur, sondern weiterhin in domänenspezifischen Architekturmodellen und mit entsprechenden Modellierungssprachen (und Werkzeugen) erfolgen. Folglich sind die wesentlichen Schnittstellen zu diesen Detailmodellen zu identifizieren. Abbildung 65 zeigt exemplarisch für jede Architekturebene

⁶¹⁴ Vgl. hierzu z.B. *Brecht* (2002), S. 162.

einige ausgewählte Elemente, die Schnittstellen zu domänenspezifischen Detailmodellen und Werkzeugen bieten. Anhaltspunkte für mögliche Schnittstellen bieten in der Regel Metaentitätstypen, die eine rekursive Beziehung (Dekomposition oder Spezialisierung) besitzen.

- (1) Auf Strategieebene bietet beispielsweise der Metaentitätstyp „Marktleistung“ eine mögliche Schnittstelle zu Detailmodellen der strategischen Unternehmensplanung. Die Leistungen/Produkte eines Unternehmens bzw. einer Geschäftseinheit sind oftmals in Form von Produkthierarchiemodellen (Produktbäumen) in speziellen Produktsystemen⁶¹⁵ detailliert erfasst. Der Ansatz der vorliegenden Arbeit bildet lediglich ab, welche Leistungen (bzw. Leistungspakete) das Unternehmen bzw. die Geschäftseinheit anbietet, sowie den Austausch der Leistungen zwischen unterschiedlichen Partnern im Geschäftsnetzwerk. Die detaillierte Dokumentation der Einzelkomponenten und Eigenschaften eines Produktes bzw. einer Leistung sowie deren Analyse sollte dagegen in den operativen Produktsystemen erfolgen.
- (2) Eine weitere Schnittstelle auf Strategieebene zu entsprechenden Detailmodellen besteht über den Metaentitätstyp „Ziel“. Jedes Ziel und jede Kennzahl des Zielmodells sollte entsprechend dem Konzept der BSC Teil einer Ursache-Wirkungs-Kette oder eines Zielmodells sein. Diese Ursache-Wirkungs-Ketten und Zielmodelle sind in der Regel in speziellen BSC⁶¹⁶- oder Performance-Management-Werkzeugen⁶¹⁷ detailliert erfasst.
- (3) Auf Organisationsebene sind beispielsweise die Prozesse und deren Aktivitäten in detaillierten, meist hierarchisch geschachtelten Modellen (z.B. UML-Aktivitätsdiagramm, BPMN⁶¹⁸ oder EPK) der Prozessablaufplanung erfasst. Dafür kommen entsprechende Werkzeuge des Prozess- und Workflowmanagements zum Einsatz.⁶¹⁹ Das in dieser Arbeit definierte Metamodell der Ablaufplanung ermöglicht im Prinzip auch eine beliebig detaillierte Abbildung der Prozessabläufe, da es auf dem Metamodell der EPK basiert. Es erscheint allerdings sinnvoll, diese lediglich mit dem geringsten Detaillierungsgrad (entsprechend dem Detaillierungsgrad der in dem Prozessmanagement-Werkzeug erfassten Modelle) abzubilden und mit dem/den entsprechenden Detailmodell(en) im Prozessmanagement-Werkzeug in Beziehung zu setzen.
- (4) Die auf Informationssystemebene identifizierten Applikationen und Softwarekomponenten können mit Hilfe von speziellen Modellierungssprachen und Werkzeugen des

⁶¹⁵ Zu Produktmodellen und -systemen im Versicherungsbereich vgl. z.B. *Coldewey* (2000) und *Schönsleben/Leuzinger* (1996).

⁶¹⁶ Für eine Beschreibung und Bewertung von BSC-Werkzeugen vgl. z.B. *Marr/Neely* (2001).

⁶¹⁷ Vgl. z.B. *Dinter/Bucher* (2006).

⁶¹⁸ Vgl. *BPMI* (2002).

⁶¹⁹ Für einen Überblick über Prozessmanagement-Werkzeuge vgl. z.B. *Blechar/Simur* (2006).

Softwaredesigns verfeinert werden.⁶²⁰ Im Unterschied zur Modellierung von Organisationsstrukturen und Prozessabläufen hat sich die UML im Bereich des Softwaredesigns als Industriestandard durchgesetzt.⁶²¹ Im Verständnis der UML 2.0 kapselt eine Komponente Zustand und Verhalten der in ihr enthaltenen Elemente und bietet nach aussen Funktionen über definierte Schnittstellen an.⁶²² Damit entspricht der Komponentenbegriff der UML 2.0 weitgehend dem allgemeinen Verständnis sowie dem Verständnis dieser Arbeit. Jede in der Unternehmensarchitektur erfasste (Software-)Komponente kann somit durch eine korrespondierende Komponente in einem UML-Komponentendiagramm repräsentiert und dort detaillierter spezifiziert werden.

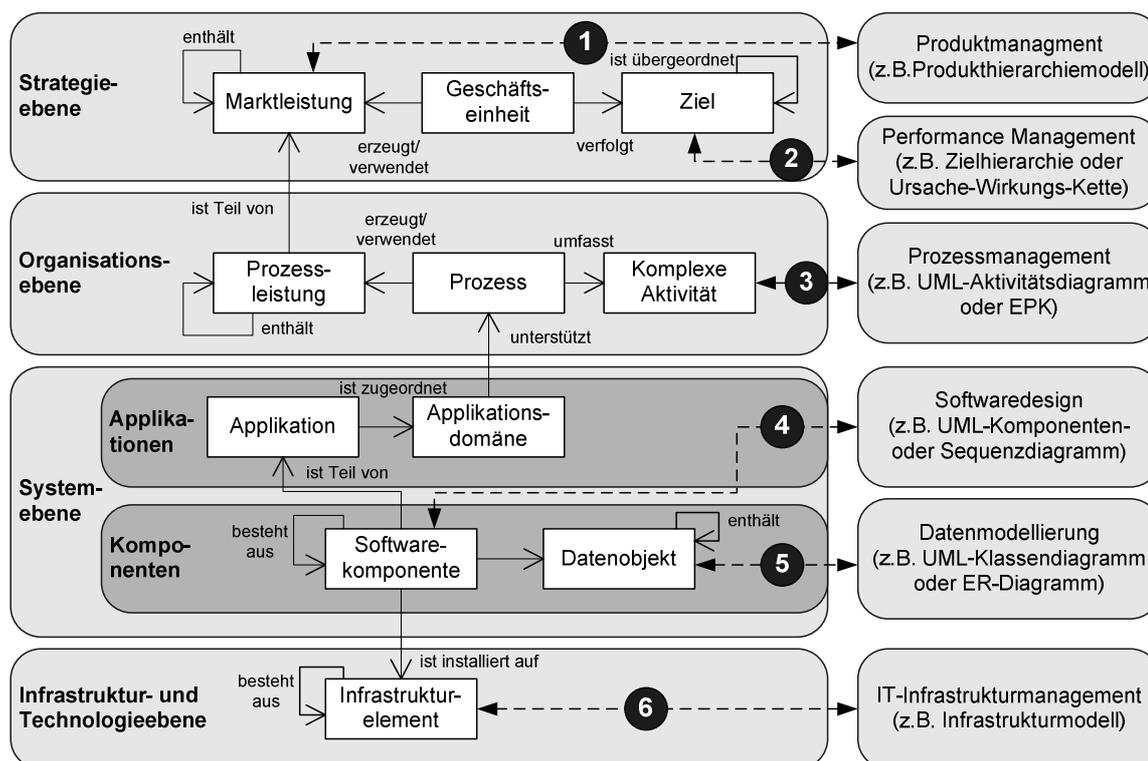


Abbildung 65: Schnittstellen zu domänenspezifischen Architekturen

- (5) Im Bereich der Datenmodellierung und des Datenbankentwurfs hat sich die UML ebenfalls als Standard etabliert. Für die Modellierung der Datenstruktur eines fachlichen Anwendungsbereichs werden in der Regel UML-Klassendiagramme verwendet. Jedes Datenobjekt der Softwarekomponenten- und Datenstrukturebene kann durch eine korrespondierende Klasse in einem UML-Klassendiagramm repräsentiert und in einem Datenmodellierungs-Werkzeug spezialisiert werden (vgl. hierzu auch Abschnitt 4.3.2.5).
- (6) Für die Abbildung der Elemente der Infrastrukturebene definiert der Ansatz der vorliegenden Arbeit ein vereinfachtes Infrastrukturmodell (vgl. Abbildung 52). Dadurch

⁶²⁰ Für einen Überblick über objektorientierte Softwaredesign-Werkzeuge vgl. z.B. Blechar (2005).

⁶²¹ Vgl. Jonkers et al. (2004), S. 281.

⁶²² Vgl. OMG (2005), S. 139.

kann für jede Softwarekomponente, Plattform sowie Entwicklungs- und Betriebskomponente erfasst werden, auf welchem Infrastrukturelement diese installiert ist. Die Infrastrukturelemente lassen sich mit entsprechenden Elementen in einem detaillierten IT-Infrastrukturmodell in Beziehung setzen. Für die Modellierung der IT-Infrastruktur kommen in der Regel IT-Portfolio- oder Systemmanagement-Werkzeuge zum Einsatz.⁶²³

Eine weitere mögliche Schnittstelle, die nicht in Abbildung 65 dargestellt wird, besteht zu einer oder mehreren Autorisierungskomponente(n), die zur Verwaltung der Zugriffsberechtigungen verwendet wird/werden (vgl. hierzu Abschnitt 4.3.2.6).

4.9 Zusammenfassung

Zusammenfassend lässt sich festhalten, dass der Ansatz dieser Arbeit sich von bereits existierenden Unternehmensarchitektur-Ansätzen vor allem durch sein klar definiertes Metamodell, die verwendeten Konzepte und insbesondere deren Zusammenhänge unterscheidet. Im Gegensatz zu vielen anderen Unternehmensarchitektur-Ansätzen fordert er explizit, dass keines der definierten Schlüsselemente in vollem Detaillierungsgrad abgebildet werden sollte, um sich dadurch insbesondere auf die Zusammenhänge zwischen diesen Elementen konzentrieren zu können. Der entwickelte Ansatz adressiert somit vor allem die im Grundlagenteil definierten Kriterien der Konsistenz, Formalisierung und Ganzheitlichkeit (vgl. Abschnitt 2.6.1).

Obwohl das ebenenübergreifende Metamodell im Prinzip auch die detaillierte Modellierung der meisten (Teil-)Bereiche (z.B. Prozessmodellierung oder Datenmodellierung) zulässt, erleichtert es die Erstellung integrierter Modelle mit einem für die Unternehmensarchitektur angemessenen Detaillierungsgrad. Es bildet somit einen Bezugsrahmen, der Elemente und Konzepte aus domänenspezifischen Ansätzen miteinander in aggregierter Form integriert und Schnittstellen zu entsprechenden Detailmodellen bietet. Die Detailmodelle können somit zunächst nebenläufig gestaltet und weiterentwickelt werden. Die entwickelte Unternehmensarchitektur liefert Vorgaben, aus denen entsprechende Anpassungen resultieren, die die Konsistenz des Gesamtsystems bewahren.

⁶²³ Vgl. hierzu z.B. *Adams* (2005).

5 Konstruktion eines Prototyps mit Hilfe eines Metamodellierungswerkzeuges

Nachdem im vorhergehenden Kapitel ein Ansatz für die Modellierung von Unternehmensarchitekturen mit einem ebenenübergreifenden Metamodell definiert wurde, gilt es nun, diesen softwaretechnisch zu unterstützen. Im nächsten Abschnitt wird zunächst ein geeignetes (Meta-)Modellierungswerkzeug ausgewählt. Anschliessend folgt die Beschreibung der Abbildung der einzelnen (Teil-)Metamodelle sowie ihrer Abhängigkeiten mit Hilfe des ausgewählten Werkzeuges. Der entwickelte Software-Prototyp wird daraufhin im Hinblick auf dessen wesentliche Komponenten und Funktionalitäten vorgestellt.

Die folgende Tabelle 19 fasst nochmals die im Grundlagenteil abgeleiteten Anforderungen an die Werkzeugunterstützung zusammen. Diese wurden bei der Wahl des Werkzeuges sowie bei der Konstruktion des Prototyps berücksichtigt.

Kriterium	Beschreibung
Modellierung und Mechanismen	
Anpassbarkeit/ Erweiterbarkeit	Eine Anpassung oder Erweiterung des zugrunde liegenden Metamodells sowie der verwendeten Darstellungsformen sollte weitgehend ohne Programmieraufwand möglich sein.
Integration	Die Integration oder Verbindung von Metamodellen aus unterschiedlichen Anwendungsbereichen sollte möglichst ohne Programmieraufwand erfolgen können.
Konsistenz	Es müssen entsprechende Mechanismen für die Überprüfung der Konsistenz zwischen den erstellten Modellen und deren Abhängigkeiten implementiert werden können.
Migration	Bei der Änderung bzw. Erweiterung des zugrunde liegenden Metamodells muss eine einfache Migration von Modellen, die noch auf einer alten Version des Metamodells basieren, möglich sein.
Wiederverwendung	Es müssen Mechanismen für die Klassifikation und Kategorisierung sowie Suchmuster für die Wiederverwendung von Modellen bereitgestellt werden.
Import/Export	Eine Datenbank (Repository) mit Import-/Export-Funktionalitäten für Metamodelle, Modelle und Methodenfragmente ist ein zentraler Bestandteil eines Metamodellierungswerkzeuges.
Analyse	Das Werkzeug muss Funktionen für die Generierung unterschiedlicher Sichten auf Modelle, Objekte und Attribute zur Verfügung stellen, um damit auch generelle Mechanismen für die Modellanalyse bereitstellen zu können.
Automatisierung	Das Werkzeug sollte Funktionalitäten für die Automatisierung bestimmter Modellierungsaktivitäten bereitstellen, beispielsweise bei der Modellgenerierung basierend auf Referenzmodellen oder der Sichtengenerierung basierend auf importierten Modellinformationen.
Organisation	
Planung	Die Integration mit Projektmanagement-Werkzeugen ermöglicht die Einhaltung und Planung der von der verwendeten Methode definierten Vorgehensweisen und Ergebnisse.
Versionierung	Unterschiedliche, vom jeweiligen Anwendungsbereich abhängige Versionierungskonzepte (beispielsweise zeitliche Versionierung, modellbasierte Versionierung oder Check-In/Check-Out-Mechanismen), sind ein wichtiger Bestandteil.
Rechtekonzept	Anwendungskonzepte werden durch Rollenmodelle und Verantwortlichkeiten unterstützt. Diese müssen auch in dem Rechtekonzept bzw. Autorisierungskonzept des Werkzeuges implementiert sein.

Benutzerfreundlichkeit	Eine intuitiv bedienbare Benutzeroberfläche erleichtert die Einarbeitung von Anwendern. Unterschiedliche Sichten, abhängig von den Kenntnissen der Anwender, sollten ohne Programmieraufwand flexibel generiert werden können.
Referenzmodellierung	Wenn die Methode Best Practices beinhaltet, sollte das Werkzeug das Vorgehen zur Anwendung und Anpassung dieser Best Practices unterstützen.
Technologie	
Verteilung	Es sollte eine verteilte Modellierung und Ergebnisdokumentation möglich sein. Die Ergebnisdokumentation könnte beispielsweise in Form von HTML-Seiten, auf die zentral im Intranet zugegriffen werden kann, erfolgen.
Schnittstellen	Das Werkzeug muss Schnittstellen zu den operativen Systemen anbieten, um die aggregierten Informationen der Unternehmensarchitektur mit den Detailinformationen der operativen Systeme verknüpfen zu können.
Architektur	Das Werkzeug sollte für unterschiedliche Architekturen, wie Stand-alone, Client-Server und Application Hosting, konfigurierbar sein. Eine offene, serviceorientierte Architektur ermöglicht die Integration zusätzlicher Modellierungswerkzeuge, Standard-Interfaces sowie wieder verwendbare Komponenten.
Plattformunabhängigkeit	Das Werkzeug sollte plattformunabhängig einsetzbar sein.

Tabelle 19: Wesentliche Anforderungen an die Werkzeugunterstützung⁶²⁴

5.1 Auswahl eines geeigneten Werkzeuges für die Abbildung des Ansatzes

Der Markt für Werkzeuge zur Modellierung der Unternehmensarchitektur befindet sich zurzeit noch in einer Entwicklungsphase (vgl. Abschnitt 2.4), wodurch die Auswahl geeigneter Werkzeuge erschwert wird. Die Auswahl der betrachteten Werkzeuge richtet sich nach dem so genannten „Magic Quadrant for Enterprise Architecture Tools“ von Gartner.⁶²⁵

Es wurden Anbieter ausgewählt, die unterschiedliche Positionen („Nischenanbieter“, „Marktführer“ und „Visionär“) in der Matrix einnehmen. Popkin Software und IDS Scheer werden von Gartner als Marktführer, Casewise als Visionär und Adaptive als Nischenanbieter eingestuft. Zusätzlich wurden Alfabet und Adonis ausgewählt, da diese vor allem im deutschsprachigen Raum weit verbreitet sind.

Die Analyse wurde im Mai 2005 abgeschlossen. Sie umfasst die in Tabelle 20 aufgeführten Werkzeuge. Die Untersuchung basiert auf Anbieterhomepages sowie Produktbroschüren, Whitepapers, Präsentationen oder Evaluierungsversionen der Lösungen, die im Internet frei zugänglich sind oder auf Anfrage zur Verfügung gestellt wurden. Darüber hinaus wurden von Gartner veröffentlichte Berichte für die Auswahl und Bewertung der Werkzeuge herangezogen. Eine detaillierte Darstellung der ausgewählten Werkzeuge anhand eines qualitativen Bezugsrahmens findet sich im Anhang dieser Arbeit (vgl. Anhang B).

⁶²⁴ Vgl. Abschnitt 2.6.2.

⁶²⁵ Vgl. James (2005a).

Kommerziell verfügbare Werkzeuge (2005)				
Anbieter	Produkt	Domäne ⁶²⁶	Marktposition ⁶²⁷	Frameworks
Adaptive	Enterprise Architecture Manager	Enterprise Architecture, IT-Management	Nischenanbieter	Zachman, FEAF
Alfabet	Strategic IT Management	IT-Management	Nischenanbieter	SITM Framework
BOC	ADONIS	Geschäftsprozessmanagement	(nicht eingeordnet)	ADONIS-Methode
Casewise	Casewise Corporate Modeler	Enterprise Architecture	Visionär	Zachman, FEAF, TEAF, Do-DAF
IDS Scheer	ARIS Design Plattform	Geschäftsprozessmanagement	Marktführer	ARIS Framework
Popkin	Enterprise Architect	Enterprise Architecture	Marktführer	Zachman, TOGAF, DoDAF

Tabelle 20: Übersicht über kommerziell verfügbare Werkzeuge

Die betrachteten Werkzeuge haben gemeinsam, dass sie alle einen oder mehrere Bezugsrahmen zur Strukturierung der Artefakte eines Unternehmens unterstützen. Die am häufigsten unterstützten Bezugsrahmen sind das Zachman Framework, FEAF und TOGAF (vgl. Abschnitt 3.3). In den meisten Fällen werden die Bezugsrahmen als Grafik mit Verweisen zu den entsprechenden Modellen umgesetzt. Einige Werkzeuge unterstützen auch die integrierte Verwendung mehrerer Bezugsrahmen, die Anpassung eines mitgelieferten Bezugsrahmens oder die Implementierung eines neuen, unternehmensspezifischen Bezugsrahmens. Werkzeuge, die primär aus dem Bereich der Repositorys stammen, wie beispielsweise der Enterprise Architecture Manager von Adaptive, bieten in der Regel umfangreichere Importfunktionen. Dagegen bieten Werkzeuge, die eher aus dem Bereich der Modellierung stammen, bessere Visualisierungsfunktionen. Die Modelle, Objekte, Eigenschaften, Beziehungen und weitere Entitäten können in allen Werkzeugen zumindest bis zu einem gewissen Grad individuell angepasst werden. Komplexere Anpassungen müssen aber in der Regel von den Anbietern durchgeführt werden. Die Artefakte werden üblicherweise in Repositorys abgelegt, denen eine gängige relationale Datenbank zugrunde liegt, welche auf einem zentralen Server in einer Multi-User-Umgebung läuft.

Da sich die betrachteten Werkzeuge hinsichtlich der Modellierungsoberfläche, der unterstützten Bezugsrahmen/Methoden, des Repositorys und der Systemarchitektur (vgl. Anhang B) sowie hinsichtlich der Erfüllung der wichtigsten im Grundlagenteil abgeleiteten Kriterien (vgl. Tabelle 21) nicht wesentlich voneinander unterscheiden, ist in der vorliegenden Arbeit das entscheidende Kriterium für die Wahl eines bestimmten Werkzeuges die Erweiterbarkeit/Anpassbarkeit. Zudem ist dieses Kriterium wichtig für die Abbildung des Metamodells mit einem möglichst geringen Programmieraufwand.

⁶²⁶ Dieses Kriterium gibt an, für welche Anwendungsdomäne das Werkzeug ursprünglich entwickelt wurde. Vgl. hierzu auch Abbildung 13.

⁶²⁷ Vgl. James (2005a).

Überblick über die Ergebnisse der Bewertung										
Lösung/Kriterium	Analyse und Simulation	Anpassbarkeit/Erweiterbarkeit	Architektur	Benutzerfreundlichkeit	Import/Export (Repository)	Integration	Konsistenz	Rechtekonzept	Versionierung	Verteilung
Adaptive Enterprise Architecture Manager	X	X	X	X ⁶²⁸	X	X	X	X	X	X
Alfabet SITM 2.0	X	- ⁶²⁹	X	X	X	X	X	X	X	X
BOC ADONIS 3.81	X	X	X	X	X	X	X	X	X	X
Casewise Corporate Modeler 10	X	X	X	X	X	X	X	X	X	X
IDS Scheer ARIS Design Platform 6.23	X	- ⁶³⁰	X	X	X	X	X	X	X	X
Popkin System Architect 10.1	X	X	X	X	X	X	- ⁶³¹	X	-	X

Tabelle 21: Erfüllung der wichtigsten Kriterien⁶³²

Die Entscheidung fiel daher zugunsten von ADONIS aus. Das Werkzeug deckt bereits ohne anwendungsspezifische Anpassungen alle Hauptkriterien ab (vgl. Tabelle 21). Durch sein methodenunabhängiges Metamodellierungskonzept ermöglicht ADONIS eine flexible und einfache Umsetzung sowie Anpassung der Metamodelle des in dieser Arbeit präsentierten Ansatzes mit wenig Programmieraufwand. Sogar komplexere Anpassungen bzw. Erweiterungen können nach relativ geringer Einarbeitungszeit vom Anwender selbst durchgeführt werden.

⁶²⁸ Der Adaptive Enterprise Architecture Manager ist ein reines Repository, das keine eigene Modellierungsoberfläche besitzt. Das Werkzeug unterstützt aber die Interaktion mit zahlreichen anderen Modellierungswerkzeugen, wie z.B. Microsoft Visio.

⁶²⁹ Alfabet ermöglicht dem Anwender lediglich die Erweiterung des Metamodells um zusätzliche Attribute. Entitäten und Beziehungstypen müssen durch Serviceleistungen von Alfabet hinzugefügt werden.

⁶³⁰ Komplexe Anpassungen müssen in der Regel durch Serviceleistungen von IDS Scheer oder dessen Partnern vorgenommen werden.

⁶³¹ Modellübergreifende Konsistenzprüfungen müssen vom Anwender selbst programmiert werden.

⁶³² Für eine Erläuterung der einzelnen Kriterien vgl. Tabelle 19 zu Beginn des Abschnitts 5.1.

5.2 Beschreibung der Abbildung in ADONIS

Nachdem im vorhergehenden Kapitel ein Lösungsansatz für das ebenenübergreifende Metamodell präsentiert wurde, wird nachfolgend beschrieben, wie die einzelnen (Teil-)Metamodelle sowie deren Abhängigkeiten mit Hilfe der zuvor ausgewählten Metamodellierungsplattform abgebildet werden können.

5.2.1 Metamodellierungskonzept

Die ADONIS-Plattform basiert auf einem methodenunabhängigen Metamodellierungskonzept, das unterschiedliche Funktionalitäten und Mechanismen für die Anpassung an eine bestimmte Methode bzw. ein bestimmtes Metamodell zur Verfügung stellt.

Die Plattform besitzt eine komponentenbasierte, verteilte und skalierbare Systemarchitektur (vgl. Abbildung 66). Die Speicherung aller Modell- und Metamodellinformationen erfolgt durch so genannte Persistenzdienste. Persistenzdienste bieten eine Schnittstelle zu einer Datenbank, in der die Daten gespeichert werden, und verbergen somit die zugrunde liegenden Speicherformen, wie beispielsweise spezifische Datenbanksysteme oder Dateisysteme. Zudem ermöglichen Speicherdienste die Verteilung und Wiederverwendung von Komponenten der gespeicherten Modelle und Metamodelle.

Das Meta-Metamodell ist der zentrale Bestandteil der Architektur der ADONIS-Plattform, da es die konzeptionelle Grundlage bildet und mit allen anderen Bestandteilen in Verbindung steht.⁶³³ Es stellt die grundlegenden Konzepte für die Generierung von Metamodellen und Mechanismen zur Verfügung, wie beispielsweise Klassen, Relationen, Attribute, Modelltypen oder Skripte.

Die „Metamodel Base“ umfasst alle Informationen über die aktuell von der Modellierungsplattform verwalteten Metamodelle. Änderungen an den zugrunde liegenden Metamodellen werden automatisch an die „Model Base“ delegiert, um die Modelle und deren zugrunde liegenden Metamodelle konsistent zu halten.

Die „Mechanism Base“ umfasst Informationen zu den Funktionalitäten und Mechanismen (z.B. Analyseabfragen), die auf Modelle und Metamodelle angewendet werden können. Diese Funktionalitäten können entweder direkt in der „Mechanism Base“ oder ausserhalb der Metamodellierungsplattform gespeichert werden.

In der „Model Base“ werden alle auf den Metamodellen basierenden Modelle verwaltet. Die „Model Base“ kommuniziert in regelmässigen Abständen mit der „Metamodel Base“, um Informationen über mögliche Änderungen der Metamodelle zu erhalten und diese an die entsprechenden Modelle weiterzuleiten.

⁶³³ Vgl. *Karagiannis/Kühn* (2002), S. 187.

Zugriffsdienste bieten dateibasierte (z.B. XML, XMI) sowie online-basierte (z.B. Java, IDL) Schnittstellen zu den unterschiedlichen Datenbanken. Entsprechend den Zugriffsrechten können die in den Datenbanken gespeicherten Informationen abgefragt und geändert werden.

Auf Grundlage der Zugriffsdienste unterstützen verschiedene Komponenten die Verwendung und Verwaltung der Metamodellierungsplattform, wie beispielsweise Modell- und Metamodell-Editor, Mechanismen-Editor etc.⁶³⁴

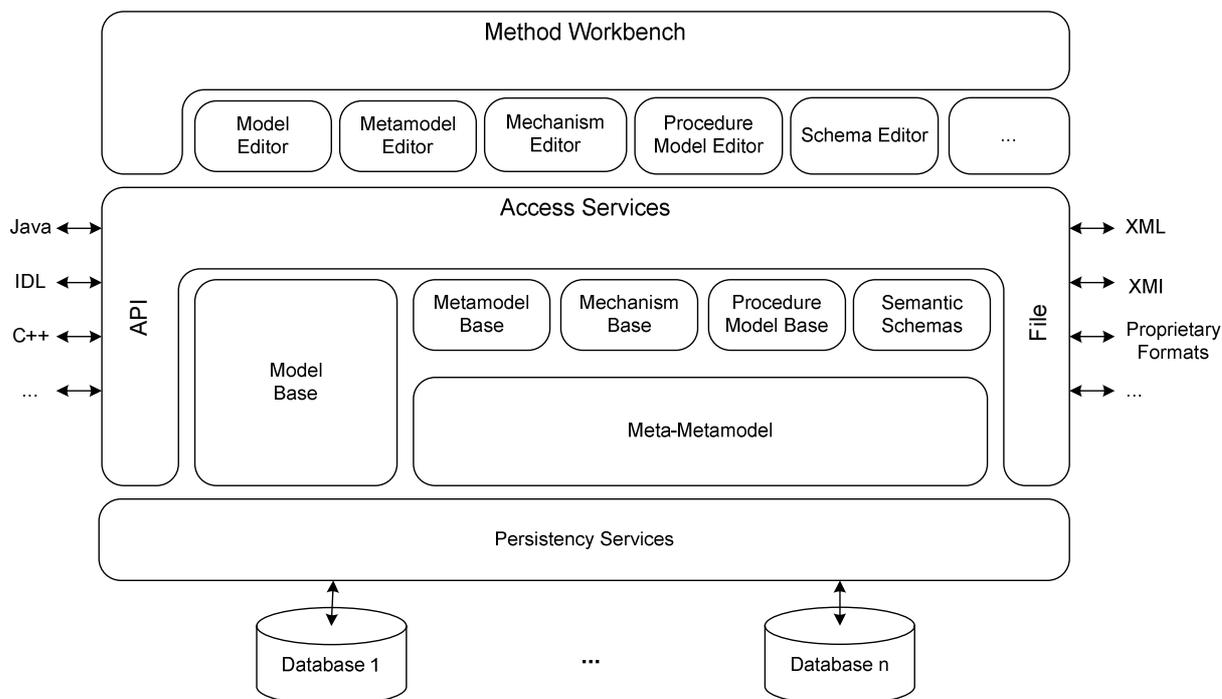


Abbildung 66: Architektur der ADONIS-Plattform⁶³⁵

5.2.2 Basiskonzepte in ADONIS

Um die Abbildung des Metamodells in ADONIS beschreiben zu können, ist es zunächst notwendig, einige grundlegende in ADONIS verwendete Konzepte darzustellen.

Jeder Methode liegt in ADONIS eine Anwendungsbibliothek zugrunde (vgl. Abbildung 67). Diese kann mit Hilfe des ADONIS-Administrations-Toolkits erstellt und angepasst werden. In der Anwendungsbibliothek sind alle Informationen für die individuelle, angepasste Verwendung von ADONIS enthalten. Sie besteht immer aus einer Geschäftsprozessbibliothek (GP-Bibliothek) mit den Informationen für Ablaufmodelle (z.B. Geschäftsprozessmodell) sowie einer Arbeitsumgebungsbibliothek (AU-Bibliothek) mit den Informationen für Aufbauamodelle (z.B. Organigramm). GP- und AU-Bibliotheken enthalten die Definitionen für Modelltypen und Klassen (inklusive Beziehungsklassen).

⁶³⁴ Vgl. Karagiannis/Kühn (2002), S. 187.

⁶³⁵ Vgl. Karagiannis/Kühn (2002), S. 186.

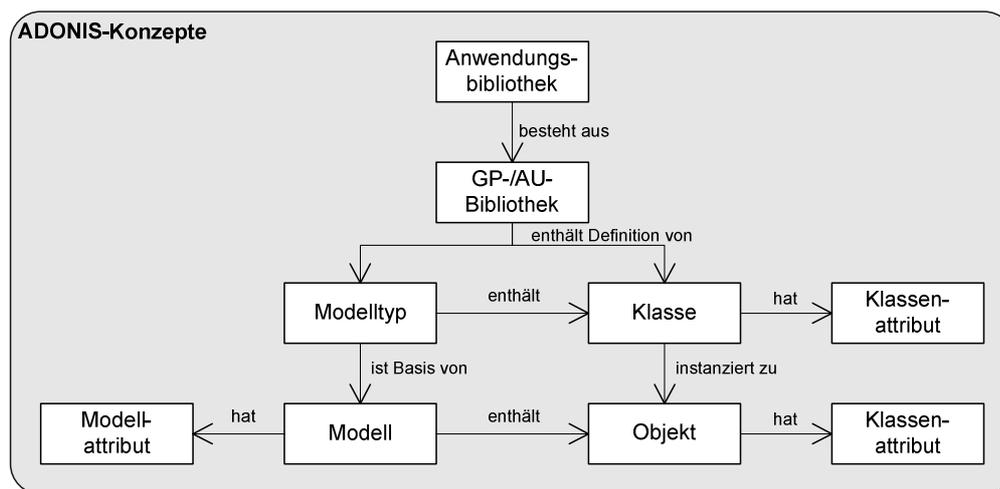


Abbildung 67: Basiskonzepte in ADONIS

Ein Modelltyp ist eine Gruppierung von Klassen. Klassen stellen die Schablone für die durch einen Modellierer angelegten Objekte dar. Klassen besitzen Klassenattribute, welche beispielsweise die grafische Darstellung eines Objekts oder die Anordnung der Objektattribute steuern. Darüber hinaus werden in den Klassen auch die (Objekt-)Attribute definiert. Jedem (Objekt-)Attribut werden bei der Definition ein Attributtyp und ein Standardwert zugewiesen. In der Modellierungskomponente von ADONIS werden auf einem Modelltyp basierende Modelle (z.B. Geschäftsprozessmodelle) angelegt. Modelle besitzen Modellattribute, welche allgemeine Informationen zum Modell (z.B. Erstellungsdatum, Status) enthalten. Ein Modell besteht aus Objekten, welche von Klassen abgeleitet (= instanziiert) sind. Objekte besitzen (Objekt-)Attribute, in welchen die Informationen für die Beschreibung des Modellinhalts hinterlegt werden.

5.2.3 Das ADONIS-Administrations-Toolkit

Die Anpassung der ADONIS-Plattform für die Abbildung der Metamodelle und Analyseauswertungen einer bestimmten Methode erfolgt mit dem Administrations-Toolkit. Jeder Methode liegt, wie bereits zuvor beschrieben, eine Anwendungsbibliothek (AB) zugrunde, die eine methodenspezifische Konfiguration der ADONIS-Plattform definiert. Jedem Benutzer wird eine AB zugewiesen, und umgekehrt können einer AB mehrere Benutzer zugeordnet sein (vgl. Abbildung 68). Jede AB ist sowohl durch die Definition der Methode und deren Metamodelle als auch durch die Definition der Auswertungsmechanismen definiert.

Das Administrations-Toolkit umfasst die folgenden Komponenten:⁶³⁶

- In der Benutzerverwaltung werden Benutzer und Benutzergruppen für die Arbeit mit der ADONIS-Plattform eingerichtet. Jedem Benutzer wird eine Anwendungsbibliothek zugewiesen. Zudem wird jeder Benutzer einer (oder mehreren) Benutzergruppe(n) zugeordnet.

⁶³⁶ Vgl. im Folgenden auch das ADONIS-Benutzerhandbuch.

- In der Bibliothekenverwaltung können Anwendungsbibliotheken importiert, exportiert, umbenannt oder gelöscht werden. Durch die Konfiguration der Bibliotheken können Bibliotheks- und Klassenattribute bearbeitet und überprüft werden.
- Benutzer können später in der Modellierungskomponente des erstellten Werkzeuges Modelle des in dieser Arbeit beschriebenen Ansatzes erstellen und bearbeiten (z.B. Geschäftsnetzwerkmodelle, Leistungsmodelle, Prozesslandkarten etc). Jedes dieser Modelle ist entweder einer GP- oder einer AU-Bibliothek zugeordnet.
- In der Modellverwaltung werden Modellgruppen angelegt, bearbeitet und gelöscht. Ferner wird definiert, welche der bestehenden Benutzergruppen Lese- oder Schreibrechte bzw. keinen Zugriff auf diese Modellgruppen haben. Zudem können Modelle importiert, exportiert und gelöscht sowie Anwendungsmodelle und Attributprofile importiert und exportiert werden.
- In der Attributprofilverwaltung werden Attributprofile angelegt, bearbeitet oder gelöscht. Attributprofile werden in der Modellierungskomponente des Geschäftsprozessmanagement-Toolkits referenziert und stellen somit das Repository-Konzept in ADONIS dar.

Die methodenspezifischen Klassen, Beziehungen und Attribute werden in enger Zusammenarbeit zwischen dem Werkzeugentwickler und dem Metamodellierer bzw. Methodenkonstrukteur in der Anwendungsbibliothek unter Verwendung des Administrations-Toolkits angelegt, um eine möglichst vollständige Abbildung der Metamodelle der Methode zu gewährleisten. In der Anfangsphase sollten daher Werkzeugentwickler und Methodenkonstrukteur gemeinsam vor Ort die Anpassung vornehmen. Zusätzliche Erweiterungen (z.B. bestimmte Analysemechanismen) oder Anpassungen können dann später auch (nach Absprache mit dem Methodenkonstrukteur) von dem Werkzeugentwickler selbst vorgenommen werden.

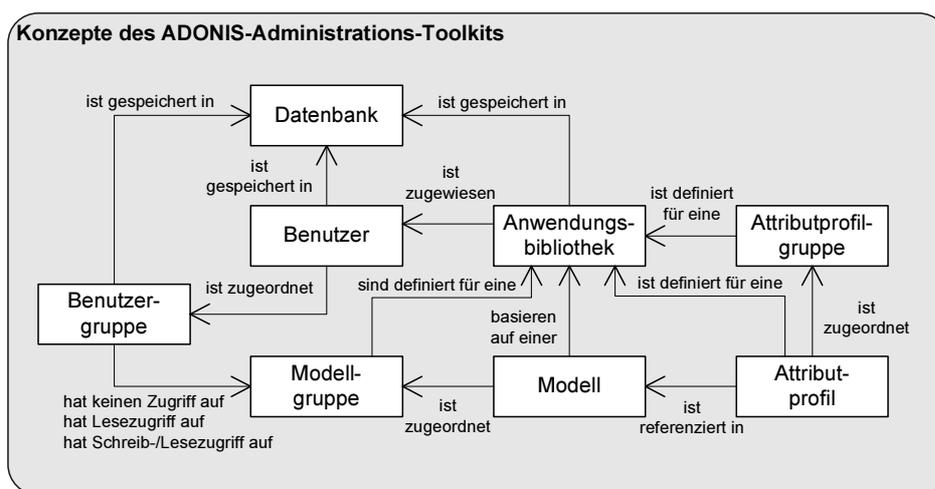


Abbildung 68: Konzepte des ADONIS-Administrations-Toolkits

5.2.4 Abbildung der Metamodelle

Die Abbildung des Metamodells des in Kapitel 4 beschriebenen Ansatzes mit Hilfe der ADONIS-Metamodellierungsplattform erfolgte in mehreren Schritten (vgl. Abbildung 69).

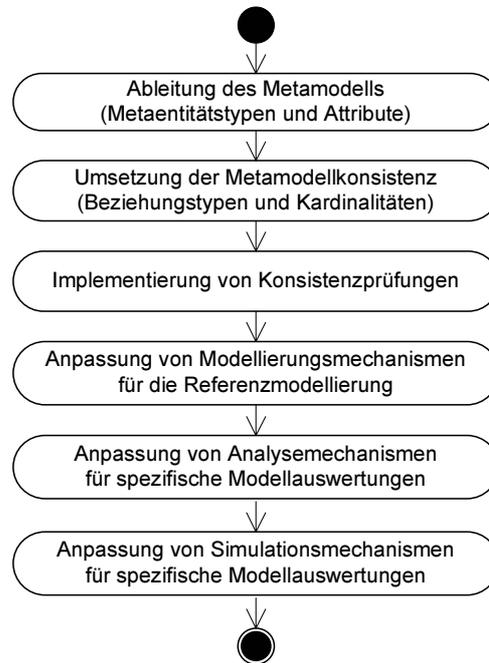


Abbildung 69: Schritte bei der Implementierung des Metamodells in ADONIS

Als Erstes wurde damit begonnen, die einzelnen (Teil-)Metamodelle des Ansatzes der vorliegenden Arbeit von dem Meta-Metamodell der ADONIS-Plattform (vgl. Abbildung 12) abzuleiten und entsprechend anzupassen. Die spezifischen Metaentitätstypen und Beziehungstypen des Ansatzes wurden in der Metamodell Datenbank der Plattform gespeichert. Im nächsten Schritt musste die Konsistenz der einzelnen Metamodelle sichergestellt werden, indem die Kardinalitäten der Beziehungen zwischen Metaentitätstypen sowie zusätzliche Funktionalitäten für die Konsistenzprüfung implementiert wurden.

Da gerade auf Strategieebene einige branchenspezifische Referenzmodelle zur Verfügung stehen, wie beispielsweise für das Leistungsmodell⁶³⁷, war es notwendig, entsprechende Funktionen für die Referenzmodellierung zu implementieren. Diese wurden mit Hilfe von Skripten umgesetzt. Dadurch ist es möglich, die benötigten Komponenten eines Referenzmodells auszuwählen und daraus neue Modelle zu generieren und diese anschliessend an das unternehmensindividuelle Umfeld anzupassen. Um die mit dem Werkzeug erstellten Modelle analysieren zu können, beispielsweise im Hinblick auf ihre Konsistenz und auf die Zusammenhänge zwischen bestimmten Objekten, wurden einige bereits durch die Metamodellierungsplattform vordefinierte Abfragen und Berichte angepasst.

⁶³⁷ Vgl. Heinrich/Winter (2004).

Weitere Analysefunktionen wurden in nachfolgenden Implementierungsschritten und im Rahmen des Testeinsatzes umgesetzt.

Um die in Kapitel 4 beschriebenen Metamodelle in ADONIS abzubilden, musste zuerst eine neue Anwendungsbibliothek angelegt und anschliessend angepasst werden. Diese besteht aus einer GP- und einer AU-Bibliothek. In der GP-Bibliothek wurden alle Metamodelle der Strategie-, Organisations- und Systemebene abgebildet. Die AU-Bibliothek wurde lediglich für die Abbildung des Metamodells der Aufbauorganisation verwendet.

Für jeden Objekttyp des Metamodells (z.B. der Objekttyp „Service Integrator“ im Metamodell des Geschäftsnetzwerks) wurde eine eigene instanzierbare Klasse in ADONIS angelegt. Jede Klasse besitzt Klassenattribute, welche beispielsweise die grafische Darstellung auf der Zeichenfläche, die Anordnung der Attribute in den ADONIS-Notebooks⁶³⁸ und die Verknüpfung mit anderen Modellen steuern. Für Beziehungstypen, die zwei Objekttypen eines Metamodells verbinden, wie z.B. „bezieht Leistung“, kann eine Beziehungsklasse definiert werden. Modellübergreifende Beziehungen bzw. Abhängigkeiten zwischen Modellen oder Objekten können mit Hilfe von Klassenattributen des Typs „Referenz“ abgebildet werden. So steht beispielsweise eine Instanz des Objekttyps „Service Integrator“ immer in Beziehung mit einer Instanz des Objekttyps Organisationseinheit (Unternehmen oder Geschäftseinheit, welche(s) die Rolle des Service Integrator im Geschäftsnetzwerk einnimmt). Durch die Definition einer Referenz kann von dem Objekt des Typs „Service Integrator“ im Geschäftsnetzwerkmodell direkt zu dem entsprechenden Objekt des Typs „Organisationseinheit“ im Modell der Aufbauorganisation navigiert werden.

Nach der Definition der einzelnen Klassen sowie ihrer Attribute, konnte für jedes Metamodell bzw. jeden Ergebnisdokumenttyp ein eigener Modelltyp in ADONIS definiert werden. Ein Modelltyp legt eine Teilmenge aller instanzierbaren Klassen und Beziehungen fest. Jedes Modell, welches mit der Modellierungskomponente erstellt wird, gehört zu einem bestimmten Modelltyp, der nach Anlegen des Modells nicht mehr geändert werden kann.

Bei der Abbildung der einzelnen Metamodellelemente in ADONIS mussten zunächst folgende grundlegenden Fragen beantwortet werden:

- Für welche Metaentitätstypen sollen instanzierbare Klassen angelegt werden?
- Soll ein Metaentitätstyp als eigene Klasse oder als Attribut implementiert werden?
- Welche Attribute soll eine bestimmte Klasse besitzen?
- Welcher Attributtyp soll einem bestimmten Attribut zugewiesen werden?
- Soll eine Beziehung mit Hilfe eines Klassenattributs des Typs „Referenz“ oder als eigene Beziehungsklasse abgebildet werden?

⁶³⁸ Vgl. hierzu Abschnitt 5.4.

- Welche Navigationsrichtungen sollen zwischen Modellen und Objekten möglich sein?
- Welche grafischen Repräsentationen sollen für die Klassen und die Beziehungsklassen verwendet werden?
- Welche Modelltypen sollen definiert werden und welche Klassen sind in diesen enthalten?

Tabelle 22 zeigt beispielhaft einige für den Modelltyp „Geschäftsnetzwerk“ definierten Klassen, Beziehungsklassen, Referenzen, Attribute und Attributtypen (vgl. dazu Abbildung 34).

Modelltyp „Geschäftsnetzwerk“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Service Integrator	Organisationseinheit	Referenz auf Organisationseinheit in Aufbauorganisation
	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Leistungsmodell	Referenz auf Leistungsmodell
...		
Kundenprozess	Kundenprozess	Referenz auf Kundenprozess in Kundenprozessmodell
	Name	Text
	Beschreibung	Text, mehrzeilig
	...	
Endkunde	Name	Text
	Beschreibung	Text, mehrzeilig
	...	
Kundengruppe	Name	Text
	Beschreibung	Text, mehrzeilig
	...	
bezieht Leistung	Art	Enumeration (standardisiert, individualisiert)
	Farbe	kontextabhängig

Tabelle 22: Modelltyp Geschäftsnetzwerk

Im Wesentlichen sind alle in Abschnitt 4 beschriebenen (Teil-)Metamodelle und deren Beziehungen in ADONIS als Modelltypen und Referenzen abgebildet. Eine detaillierte Dokumentation der Abbildung der einzelnen Metamodelle in ADONIS findet sich im Anhang dieser Arbeit (vgl. Anhang A).

5.3 Überblick über den Aufbau und die generellen Funktionalitäten

Der Prototyp besteht aus vier wesentlichen Komponenten.⁶³⁹ Den Kern des Prototyps bildet die Modellierungskomponente. Diese stellt Werkzeuge für die Abbildung der einzelnen Modelle einer Unternehmensarchitektur (z.B. Geschäftsnetzwerkmodell, Prozesslandkarte, Aufbauorganisation) zur Verfügung. Die Modelle können mit Hilfe des grafischen Editors (Modelleditor) entworfen und verändert werden. Zusätzlich besteht auch die Möglichkeit einer tabellarischen Modellierung, die vor allem bei der Eingabe von Attributwerten hilfreich ist.

⁶³⁹ Vgl. im Folgenden auch die Dokumentation der ADONIS-Metamodellierungsplattform.

In der Analysekomponente können Abfragen auf den mit der Modellierungskomponente erstellten Modellen durchgeführt und Beziehungstabellen sowie vordefinierte Pläne erzeugt werden. Die Komponente bietet einerseits standardisierte Abfragen an, welche durch Vervollständigen eines Lückentextes ausgeführt werden können. Darüber hinaus können benutzerdefinierte Abfragen durch Verknüpfung von standardisierten Abfragen erstellt oder vom Anwender selbst definiert werden. Die vordefinierten Abfragen werden vom Administrator im Administrations-Toolkit definiert und dem Anwender bzw. Modellierer zur Verfügung gestellt. Die Darstellung der Ergebnisse der Abfragen kann als Text oder in einem Diagramm erfolgen. Zudem können die Ergebnisse in Text-Dateien exportiert und in anderen Applikationen (z.B. Tabellenkalkulation, Textverarbeitung etc.) weiterverarbeitet werden.

In der Simulationskomponente können Simulationen von Geschäftsprozessen bzw. der Ablauforganisation und der Aufbauorganisation durchgeführt werden. Dafür stehen unterschiedliche Simulationsalgorithmen zur Verfügung, mit deren Hilfe beispielsweise Pfadanalysen von Geschäftsprozessmodellen erstellt sowie Belastungsanalysen von Geschäftsprozessmodellen in Verbindung mit Modellen der Aufbauorganisation durchgeführt werden können.

Die Import/Export-Komponente ermöglicht es, bestehende Modelle und Modellgruppen in ein von der Metamodellierungsplattform abhängiges Dateiformat (ADL-Format) zu exportieren bzw. Dateien in das Werkzeug zu importieren. Der Datei-Export kann zusätzlich zur Datensicherung der erstellten Modelle und Modellgruppen dienen. Darüber hinaus bietet die Import/Export-Komponente die Möglichkeit zur Dokumentation der Modelle. Die Modelle können in eine elektronische Modelldokumentation (z.B. HTML-, XML-Dateien) oder in eine Papierdokumentation (z.B. RTF-Dateien für MS Word) überführt werden. Dadurch können beliebig definierbare Modellinhalte in Dokumente eingebunden und beispielsweise unternehmensweit publiziert werden. Die folgende Tabelle gibt einen Überblick über die wichtigsten Komponenten des Werkzeuges sowie deren Funktionalitäten.

Komponente	Beschreibung	Funktionalitäten
Modelleditor	Grafischer sowie textueller Editor für die Erstellung von Modellen, der unterschiedliche Ansichtsoptionen sowie Navigationsmöglichkeiten zwischen den Modellen bietet	<ul style="list-style-type: none"> • Erstellen/Bearbeiten von Modellen • Grafische/tabellarische Darstellung • Modellnavigation • Modellverwaltung • Prüfung von Modellierungsrichtlinien
Analyse	Durchführung von Abfragen auf den mit der Modellierungskomponente erstellten Modellen sowie Generierung von Beziehungstabellen	<ul style="list-style-type: none"> • Vordefinierte Abfragen • Benutzerdefinierte Abfragen • Beziehungstabellen • Impact-Analysen • Vordefinierte Pläne
Simulation	Durchführung von Simulationsalgorithmen auf Modellen der Ablauf- und Aufbauorganisation	<ul style="list-style-type: none"> • Pfadanalyse • Belastungsanalyse • Berechnung von Kosten und Zeit

Import/Export	Import/Export von Modellen in unterschiedliche Datei-Formate	<ul style="list-style-type: none"> • ADL-Import/Export • Export/Import in HTML und XML • Export/Import in RTF- und Word-Dateien
Dokumentation	Dokumentation der erstellten Modelle und Veröffentlichung im Intranet	<ul style="list-style-type: none"> • MS Word • HTML-Seiten • Intranet

Abbildung 70: Komponenten und Funktionalitäten des Prototyps

5.4 Erstellen von Modellen im Modelleditor

Der Modelleditor enthält alle Funktionen, um Modelle zu erstellen und zu bearbeiten. Für die Eingabe von Daten steht zusätzlich die tabellarische Modelldarstellung zur Verfügung. Jedes Modell wird in einem eigenen Modellfenster dargestellt und enthält Objekte und Konnektoren (Beziehungen).

Beim Anlegen eines neuen Modells erscheint zunächst ein Dialog, welcher die einzelnen Modelltypen entsprechend der Ebenen Geschäftsstrategie, Organisation, Informationssystem und IT-Infrastruktur gruppiert anzeigt (siehe Abbildung 71). Nachdem ein bestimmter neu zu erstellender Modelltyp ausgewählt wurde, wechselt die Ansicht zur Modellierungsoberfläche. Auf der linken Seite der Modellierungsoberfläche werden die für den entsprechenden Modelltyp verfügbaren Klassen und Beziehungsklassen (bzw. deren Notationselemente) angezeigt. Für den Modelltyp „Geschäftsnetzwerk“ werden dort beispielsweise die Klassen „Service Integrator“, „Shared Service Provider“, „bezieht Leistung“ etc. angezeigt. Durch Anklicken einer Klasse können mehrere Objekte dieser Klasse auf dem Arbeitsbereich platziert werden.

Die Attributwerte eines Objektes werden über das so genannte Notebook festgelegt. Durch einen Doppelklick auf ein bestimmtes Objekt wird dessen Notebook angezeigt. Dieses enthält die für die Klasse dieses Objektes definierten Attribute. Dort können nun die für dieses Objekt geltenden Attributwerte entsprechend den definierten Attributtypen eingetragen werden. Häufig vorkommende Attributtypen sind beispielsweise Text, Zahl, Aufzählung und Referenz. Um die Attributwerte der Objekte einer bestimmten Klasse bzw. der Konnektoren eines bestimmten Beziehungstyps von bestehenden Modellen systematisch bearbeiten zu können, kann die tabellarische Modellierung verwendet werden.

Nach der Erstellung eines neuen Modells können dessen Kardinalitäten überprüft werden. Diese sind durch das dem Modell zugrunde liegende Metamodell definiert. Die Kardinalitäten geben an, wie viele Objekte einer Klasse in einem Modell enthalten sind, wie viele Konnektoren einer Beziehung von einem Objekt wegführen und wie viele Konnektoren einer Beziehung zu einem Objekt hinführen. Der Bereich der Kardinalitäten wurde bei der Abbildung des Metamodells des Ansatzes in der Anwendungsbibliothek begrenzt (Kardinalitätsregeln), um beispielsweise Modellierungsrichtlinien überprüfen zu können. Da die Kardinalitätsregeln nicht automatisch während der Modellierung geprüft werden, muss die Prüfung der Kardinali-

täten eines Modells manuell über den entsprechenden Menüeintrag gestartet werden. Sollten dabei die definierten Kardinalitätsregeln verletzt werden, wird eine entsprechende Fehlermeldung angezeigt. Die Prüfung kann daraufhin abgebrochen und die angezeigte Verletzung korrigiert werden, oder es kann mit der Prüfung fortgefahren werden.

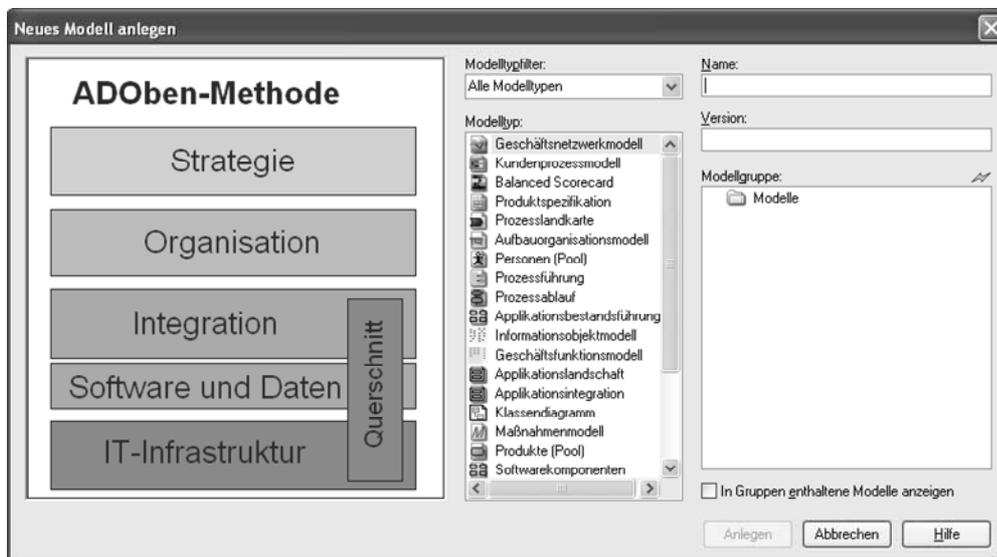


Abbildung 71: Anlage eines neuen Modells

In einer Multi-User-Umgebung können die Benutzer auf die erstellten Modelle und Objekte gemeinsam zugreifen. Diese ermöglicht das kollaborative Arbeiten mehrerer Benutzer an zentral gespeicherten Modellen von unterschiedlichen Standorten aus. Offline-Arbeiten an Modellen können mit dem Repository synchronisiert werden. Ein Zugriffs- und Rechtesystem sowie eine Reihe von Sicherheitsoptionen verhindern den unbefugten Zugriff auf die in der zentralen Datenbank gespeicherten Modelle. So können beispielsweise Benutzergruppen einer Modellgruppe zugeordnet werden und die Zugriffsrechte dieser Benutzergruppen auf die Modellgruppe definiert werden. Die Zugriffsrechte werden dabei nach „kein Zugriff“, „Lesezugriff“ und „Schreib-/Lesezugriff“ eingestuft. Des Weiteren lässt sich der Zugriff auf Komponenten für bestimmte Benutzer einschränken, so dass diesen nicht alle Funktionalitäten und Mechanismen des Werkzeuges zur Verfügung stehen.

5.5 Durchgängigkeit und Abbildung der Gesamtzusammenhänge

Die Abhängigkeiten zwischen den Modellen auf einer Ebene sowie denen auf unterschiedlichen Ebenen werden durch modellübergreifende Referenzen abgebildet. Diese bieten die Möglichkeit, von einem Objekt eines Modells auf andere Modelle (Modellreferenz) oder Objekte in anderen Modellen (modellübergreifende Objektreferenz) zu verweisen. Da die Abbildung der Zusammenhänge zwischen den einzelnen Modellen von besonderer Bedeutung ist, werden nachfolgend zur besseren Veranschaulichung einige Referenzen zwischen Modellen

des in dieser Arbeit vorgestellten Ansatzes zur Modellierung der Unternehmensarchitektur beispielhaft beschrieben.

- Eine Rolle im Geschäftsnetzwerk ist immer mit einer Organisationseinheit (Unternehmen bzw. Geschäftseinheit) assoziiert (vgl. Abbildung 63). Durch Anklicken einer Rolle im Geschäftsnetzwerkmodell kann der Modellierer direkt zur entsprechenden Organisationseinheit im Modell der Aufbauorganisation navigieren (siehe Abbildung 72).
- Eine weitere Referenz besteht beispielsweise jeweils zwischen einem bestimmten Kundenprozess im Geschäftsnetzwerkmodell und dessen detaillierter Darstellung mit seinen Teilaktivitäten und Teilleistungen im entsprechenden Kundenprozessmodell. Durch Anklicken des Kundenprozessobjekts im Geschäftsnetzwerkmodell wird das entsprechende Kundenprozessmodell im Arbeitsbereich geöffnet (siehe Abbildung 72).

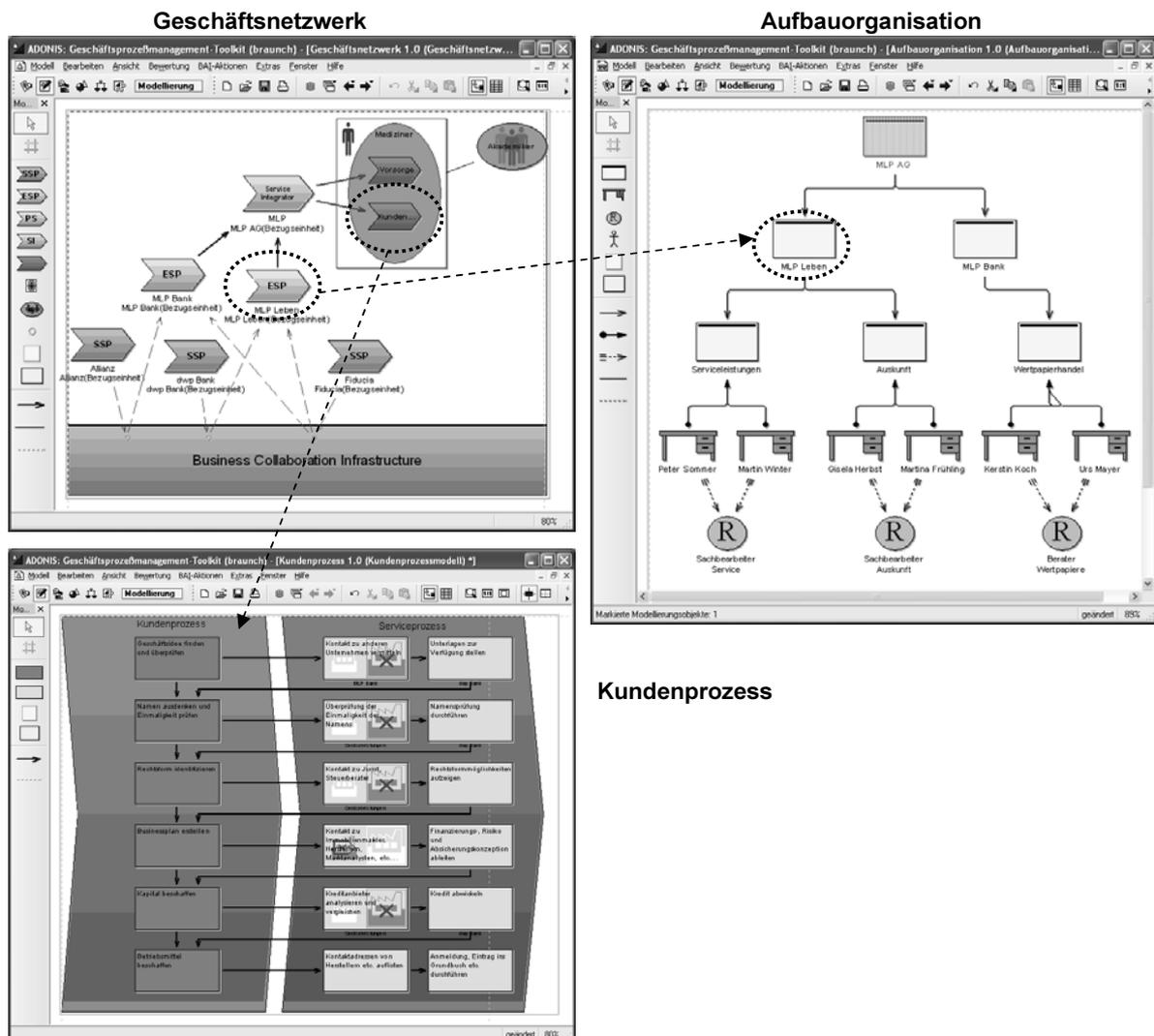
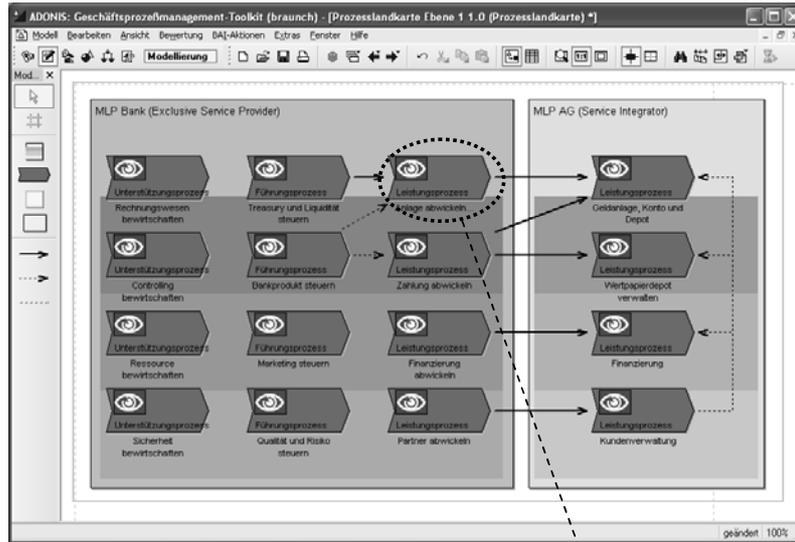
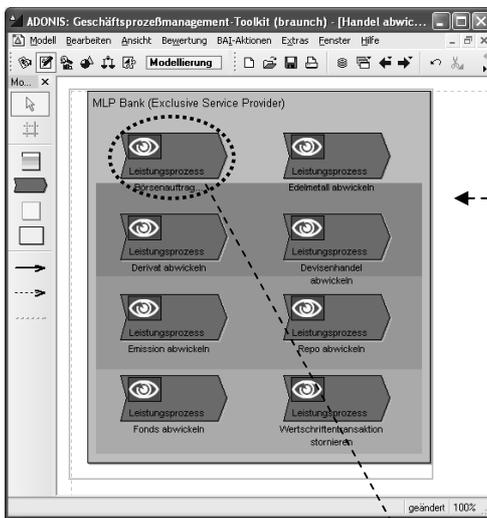


Abbildung 72: Modellübergreifende Referenzen

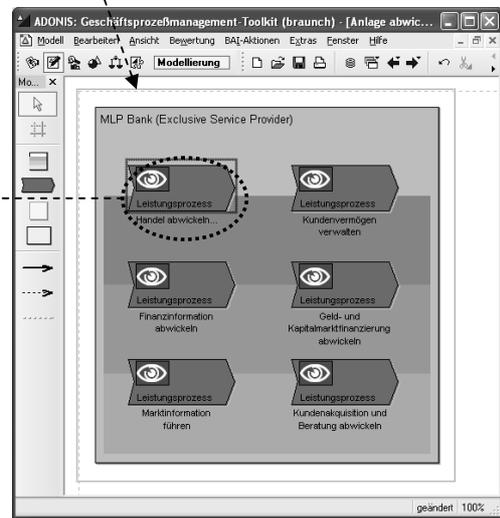
PL Level 1



PL Level 3



PL Level 2



Prozessablauf

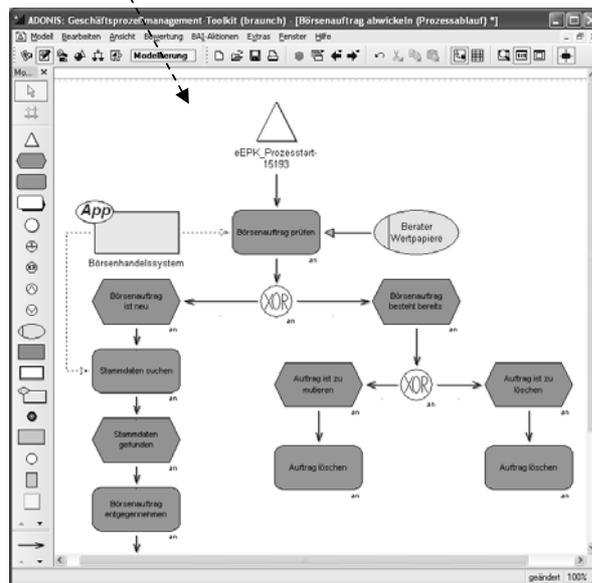


Abbildung 73: Prozesshierarchie

- Die auf Prozessebene erzeugten Prozessleistungen werden in einem Pool hierarchisch strukturiert verwaltet. Sie können bei der Modellierung der Prozesslandkarte sowie des Prozessablaufs verwendet werden. Darüber hinaus können für jede Prozessleistung eine oder mehrere Referenzen auf eine Leistungsspezifikation definiert werden (vgl. Beziehung 5 in Abbildung 63).
- Ein Prozess innerhalb einer Prozesslandkarte kann entweder auf eine weitere Prozesslandkarte oder auf den entsprechenden Prozessablauf verweisen (siehe Abbildung 73). Innerhalb des Prozessablaufs kann von einer Aktivität auf einen detaillierteren Prozessablauf referenziert werden. Dadurch ist eine hierarchische Prozessmodellierung mit unterschiedlichen Detaillierungsgraden möglich.
- Die kritischen Erfolgsfaktoren und Kennzahlen der Prozessführung (vgl. Abbildung 74) korrespondieren mit den auf Strategieebene aus der Balanced Scorecard abgeleiteten kritischen Erfolgsfaktoren und Kennzahlen (vgl. Beziehung 3 in Abbildung 63). Diese Korrespondenz wird ebenfalls durch eine modellübergreifende (Objekt-)Referenz abgebildet. Darüber hinaus können die Kennzahlen auf Prozessebene auch als Referenz auf die Prozessführung innerhalb eines Prozessablaufs dargestellt werden.

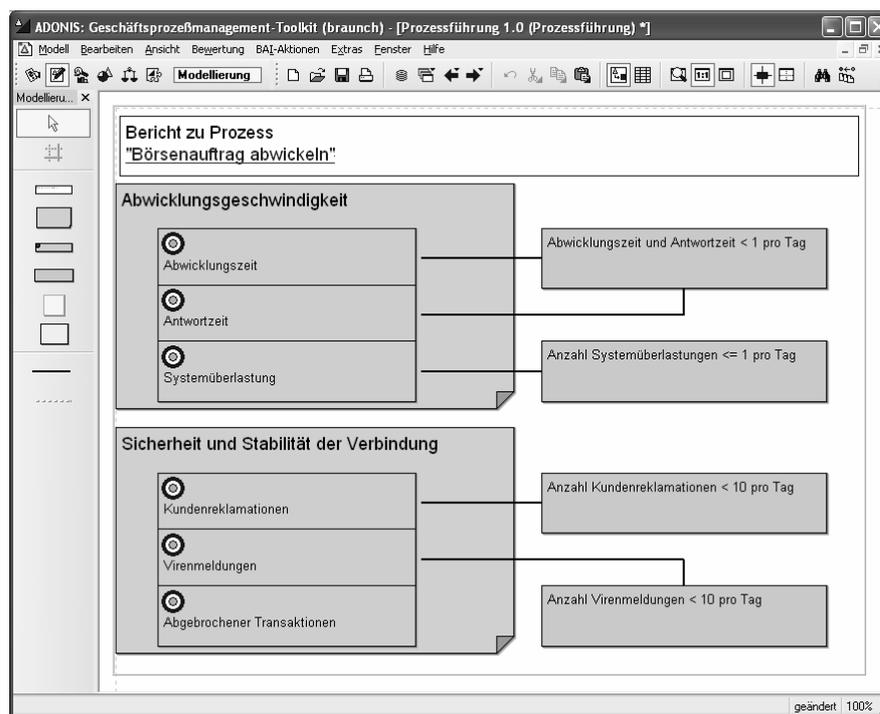


Abbildung 74: Beispiel Prozessführung

- Die Geschäftsfunktionen und Informationsobjekte der auf Prozessebene modellierten Prozessabläufe besitzen jeweils eine Referenz auf die entsprechende Geschäftsfunktion bzw. das entsprechende Informationsobjekt im Geschäftsfunktions- bzw. Informationsobjektmodell. Ausserdem besitzen die im Geschäftsfunktions- bzw. Informationsobjektmodell dokumentierten Geschäftsfunktionen bzw. Informationsobjekte Referenzen auf die

zugrunde liegenden Erhebungsquellen (Prozesse der Prozesslandkarte, Organisationseinheiten der Aufbauorganisation oder Applikationen der Applikationsbestandsführung)

- Die aus den harmonisierten Geschäftsfunktionen, Informationsobjekten und Organisationseinheiten abgeleiteten Applikationen werden in der Applikationsbestandsführung verwaltet. Dort können sie bei der Modellierung der Prozessabläufe referenziert werden, um die informationstechnische Unterstützung der Aktivitäten von Prozessen abzubilden (vgl. Beziehung 4 in Abbildung 64).

Die Arbeit mit mehreren, eventuell aufeinander referenzierenden Modellen wird durch die Modellnavigation des Werkzeuges vereinfacht. Eine praktische Funktion der Modellnavigation ist der Modell-Explorer. Der Explorer ist ein Sichtfenster, in dem die erstellten Modelle und Modellgruppen sowie deren Statusinformation angezeigt werden. Er ermöglicht einen Überblick über alle vorhandenen Modelle in Form einer Baumstruktur und einen direkten Zugriff auf einzelne Modelle zur Bearbeitung. Zudem bietet er eine eigene Symbolleiste, die Zugriff auf die Funktionen und Sichten ermöglicht. Weitere wichtige Funktionen der Modellnavigation, die dem Modellierer die Arbeit erleichtern, sind die Modellumlauffunktion für das Navigieren zwischen mehreren Modellfenstern und die Funktion für das automatische Öffnen von referenzierten Modellen.

Das Werkzeug bietet darüber hinaus die Möglichkeit, die modellübergreifenden Referenzen zu klassifizieren. Dies ist in allen Modellauswahllisten von Bedeutung, in welchen die Option „Inklusive referenzierter Modelle“ verfügbar ist (z.B. „Modell öffnen“, „Abfragen“, „Export“), da für die in diesen Listen selektierten Modelle alle referenzierten Modelle gemäß ihren Einstellungen ermittelt und weiterverarbeitet werden. Dabei wird ein „Referenzbaum“ mit allen erreichbaren bzw. aufgrund der Einstellungen referenzierten Modellen gebildet.

Durch die Einstellungen zu den modellübergreifenden Referenzen kann festgelegt werden, bis in welche Tiefe (Level) die Referenzen verfolgt werden und ob es sich um Haupt- oder Nebenreferenzen handelt. Die Tiefe gibt an, bis zu welchem Level – ausgehend vom selektierten Modell – die Referenzen verfolgt und die referenzierten Modelle berücksichtigt werden. Durch die Klassifikation in Hauptreferenz oder Nebenreferenz kann festgelegt werden, wie nach dem Verfolgen einer Referenz weiter vorgegangen werden soll. Hauptreferenzierte Modelle werden behandelt wie das (selektierte) Ausgangsmodell, d.h. alle in diesen Modellen enthaltenen Referenzen werden verfolgt, wenn die Beschränkung der Tiefe dies zulässt. Bei nebenreferenzierten Modellen werden die in diesen Modellen enthaltenen Referenzen nur dann weiter verfolgt, wenn die enthaltenen Referenzen vom gleichen Attribut eines Objekts der gleichen Klasse ausgehen und die Beschränkung der Tiefe dies zulässt. Im Gegensatz zu Hauptreferenzen wird die Beschränkung der Tiefe bei Nebenreferenzen pro Ebene im Referenzbaum erneut ausgewertet. Bei Hauptreferenzen wird die Tiefe global über alle Ebenen hinweg überprüft. Die nachfolgende Abbildung 75 zeigt beispielhaft einen einfachen Referenzbaum.

Das "Modell A" ist das Ursprungsmodell, von dem aus der Referenzbaum gebildet wird. Die Modelle "Modell B", "Modell C" und "Modell D" werden über eine Hauptreferenz (z.B. Subprozesse des Prozesses, der in „Modell A“ abgebildet ist) von "Modell A" referenziert und bilden gemeinsam den Referenzbaum der Hauptreferenzen (Hauptbaum). Eine Tiefe von "2" dieser Hauptreferenz bedeutet daher, dass im zu bildenden Referenzbaum das "Modell D" unberücksichtigt bleibt und damit nicht angezeigt wird.

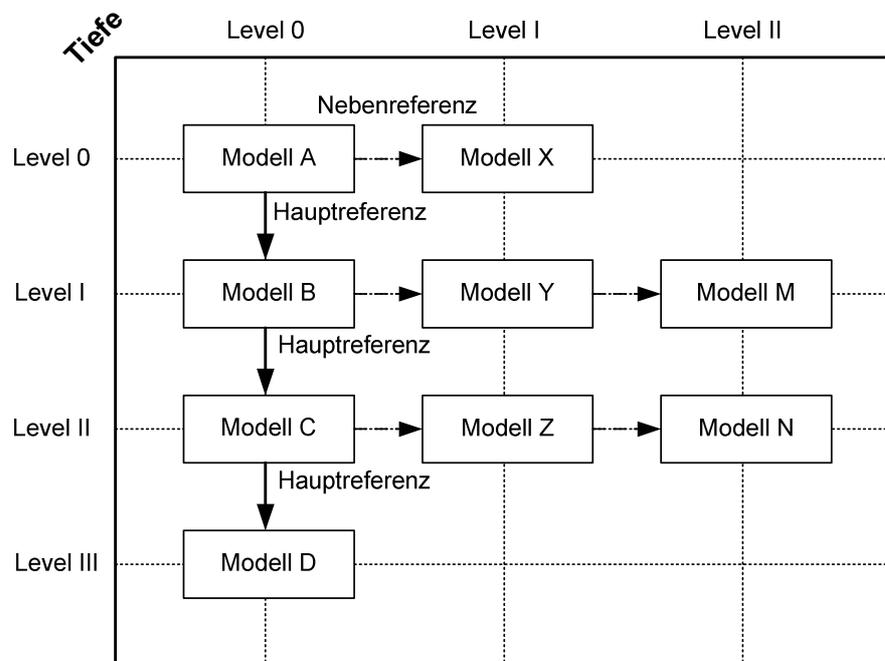


Abbildung 75: Referenzbaum

Die Nebenreferenzen werden pro Level betrachtet, d.h. die Tiefenauswertung erfolgt immer von einem Modell des Hauptbaumes ausgehend, wobei die Zählung wieder bei 0 beginnt. Eine Tiefe von "1" bei den Nebenreferenzen bedeutet bezogen auf Abbildung 75 daher, dass im zu bildenden Referenzbaum die Modelle "Modell M" und „Modell N“ unberücksichtigt bleiben.

5.6 Analyse der erstellten Modelle

Mit der Analysekomponente können die erstellten Modelle sowie deren Beziehungen statisch ausgewertet werden. Die Funktionalität „Abfragen/Reports“ ermöglicht die Auswertung beliebiger Modellinhalte (Objekte bzw. Konnektoren und deren Attribute), die den in der Abfrage definierten Kriterien genügen. Die Ergebnisse dieser Abfragen können grafisch und tabellarisch dargestellt sowie in unterschiedlichen Datei-Formaten gesichert werden.

Zudem können die in der Anwendungsbibliothek vordefinierten Abfragen durchgeführt und so genannte Beziehungstabellen erstellt werden. Die Beziehungstabellen stellen existierende Beziehungen zwischen zwei Klassen beliebiger Modelltypen dar. Die Ergebnisse der vordefinierten Abfragen und die Beziehungstabellen können ebenfalls gesichert werden.

Das Werkzeug bietet zwei Arten von Abfragen. Bei einer standardisierten Abfrage wird die Abfrage mit Hilfe eines Lückentextes durchgeführt, ohne einen AQL-Ausdruck⁶⁴⁰ angeben zu müssen (vgl. Abbildung 76). In den Lückentext müssen die für die Abfrage notwendigen Informationen eingegeben und aus der Liste der Klassen, Attribute oder Vergleichsoperatoren die gewünschten Werte ausgewählt werden. Typische standardisierte Abfragen sind beispielsweise:

- Ermittle alle Objekte, die mit dem Objekt [*Objektname*] der Klasse [*Auswahl*] über die Beziehung [*Auswahl*] verbunden sind.

Mit dieser standardisierten Abfrage können alle Objekte ermittelt werden, die mit dem definierten Objekt in Beziehung stehen. Die Wahl des Objekts „Handel abwickeln“ der Klasse "Prozess" und der Beziehung „bezieht Leistung“ ermittelt z.B. alle Prozesse in den ausgewählten Modellen, die von diesem Prozess eine Leistung beziehen.

- Ermittle alle Objekte der Klasse [*Auswahl*] mit Attribut [*Auswahl*] [*Vergleichsoperator*] [*Eingabe*].

Durch diese Abfrage werden alle Objekte der gewählten Klasse mit einem bestimmten Attributwert ausgegeben. So liefert z.B. die Auswahl der Klasse „Aktivität“ mit dem Attribut „Kosten“, dem Vergleichsoperator „>“ und der Eingabe des Wertes „1000“ alle Objekte der Klasse „Aktivität“, deren Kosten größer als 1000 sind.

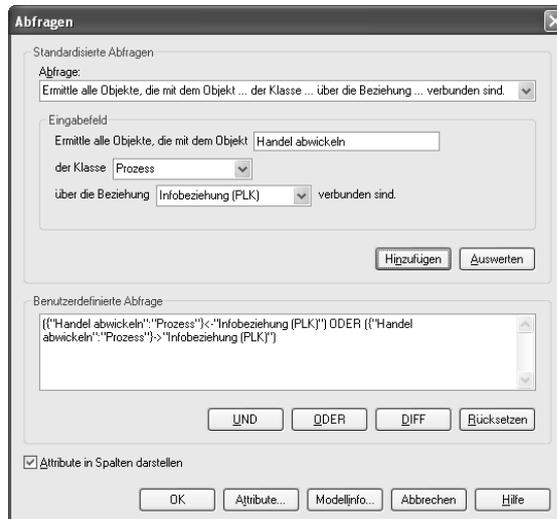


Abbildung 76: Standardisierte und benutzerdefinierte Abfragen

Bei einer benutzerdefinierten Abfrage definiert der Anwender die Abfrage durch Verknüpfen von standardisierten Abfragen oder durch AQL-Ausdrücke, welche vom Anwender direkt

⁶⁴⁰ AQL (ADONIS Query Language) ist die Abfragesprache der ADONIS-Plattform, mit der Abfragen auf allen Modellen durchgeführt werden können. Eine Definition der Syntax dieser Abfragesprache findet sich im Benutzerhandbuch von ADONIS.

eingegeben werden. Das Ergebnis eines AQL-Ausdrucks ist eine Menge von Objekten bzw. Konnektoren, die dem gewünschten Suchkriterium genügt.

Eine weitere Funktionalität der Analysekomponente sind die so genannten Beziehungstabellen. Diese stellen die Existenz von Beziehungen dar, wobei in diesem Fall unter Beziehung sowohl Konnektoren (innerhalb eines Modells) als auch modellübergreifende Referenzen zu verstehen sind. In einer Beziehungstabelle wird angezeigt, welche Von-Objekte (z.B. Person) mit welchen Nach-Objekten (z.B. Rollen) verbunden sind (vgl. Abbildung 77). Aufgrund der Möglichkeit, auch die Beziehungen von modellübergreifenden Referenzen auszuwerten, können die dargestellten Von- und Nach-Objekte in unterschiedlichen Modellen, auf unterschiedlichen Ebenen und auch in Modellen unterschiedlicher Typen enthalten sein.

	Berater Finanzierung	Berater Wertpapiere	Sachbearbeiter Auskunft	Sachbearbeiter Service
Beat Braun	x			
Gisela Herbst			x	
Kerstin Koch		x		
Martin Winter				x
Martina Frühling			x	
Peter Sommer				x
Reto Frey	x			
Urs Mayer		x		

Abbildung 77: Beziehungstabelle

Für die Analyse der Ablauf- und Aufbauorganisation stehen unterschiedliche Simulationsalgorithmen zur Verfügung. Dadurch können beispielsweise Restrukturierungsmaßnahmen vorweggenommen und deren Auswirkungen aus verschiedenen Blickwinkeln betrachtet werden.

Die Pfadanalyse ermöglicht eine Bewertung der erstellten Modelle der Ablauforganisation (Prozessmodelle) ohne Betrachtung der Aufbauorganisation. Für ein Prozessmodell (inklusive eventuell aufgerufener Subprozesse) liefert die Pfadanalyse prozess- und pfadbezogene Ergebnisse. Die prozessbezogenen Ergebnisse beschreiben durchschnittliche Zeiten und Kosten des Prozessmodells. Die durchlaufenen Pfade können anhand ihrer Auftrittswahrscheinlichkeiten und den von ihnen definierten Zeit- und Kostenkriterien bewertet werden. Zusätzlich werden die einzelnen Pfade im Hinblick auf die Zeit- und Kostenkriterien grafisch und textuell dargestellt. Ausgangsbasis für die Pfadanalyse ist ein Prozessmodell mit allen zugehörigen Subprozessen. Dieser Prozess sowie die von ihm aufgerufenen Subprozesse müssen korrekt modelliert sein, d.h. sie müssen den Modellierungsrichtlinien (z.B. Kardinalitäten) entsprechen. Des Weiteren müssen für die einzelnen Aktivitäten des Prozesses die durchschnittlichen Zeiten und Kosten ermittelt und im Prozessmodell dokumentiert sein (vgl. Abbildung

78). Bei der Durchführung der Pfadanalyse werden anschliessend die Pfade des Prozessmodells durchlaufen und die dabei entstehenden Zeiten und Kosten ermittelt.

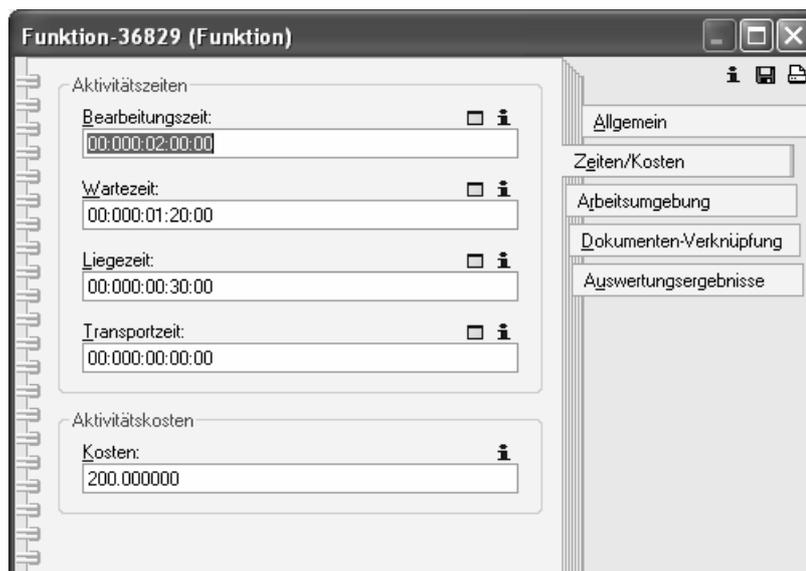


Abbildung 78: Pfadanalyse und Simulation

Die Belastungsanalyse simuliert Prozesse unter Berücksichtigung der zugehörigen Aufbauorganisation. Dabei werden die Aktivitäten der simulierten Prozesse von den ihnen zugeordneten Rollenträgern ausgeführt. Gegebenenfalls werden für die Bearbeitung Ressourcen benutzt. Dadurch können prozess- und periodenbezogene Aussagen bezüglich prozess- und stellenbezogener Belastungen gemacht werden. Auf Basis der Ergebnisse der Belastungsanalyse kann beispielsweise eine Personalbedarfsplanung durchgeführt werden. Ausgangsbasis für die Belastungsanalyse ist ein Anwendungsmodell, bestehend aus genau einem Aufbauorganisationsmodell und mindestens einem Prozessmodell. Zusätzlich zur korrekten Modellierung des Ablaufes, analog zur Pfadanalyse, ist bei der Belastungsanalyse die Definition der Bearbeiter (Rollen bzw. Rollenträger) der einzelnen Aktivitäten erforderlich. Wenn in den Prozessmodellen Ressourcen (z.B. Applikationen, die die Durchführung einer Aktivität unterstützen) modelliert wurden, ist auch eine Ressourcenzuordnung erforderlich.

Neben den zuvor genannten Analyse- und Simulations-Mechanismen lassen sich auch sogenannte Impact-Analysen durchführen. Die Impact-Analyse ist zur Erkennung der Auswirkungen von Veränderungen an der Unternehmensarchitektur und damit für ein proaktives Management von zentraler Bedeutung. Grundlage dieser Analyseform sind die im Metamodell definierten Beziehungen zwischen unterschiedlichen Modellen bzw. Sichten, die sich gegebenenfalls auf unterschiedlichen Ebenen befinden. Die Ausführung einer Impact-Analyse erfordert das Durchlaufen unterschiedlicher Sichten und Ebenen der Unternehmensarchitektur und die Auswertung der entsprechenden Beziehungen, um festzustellen, ob die Änderung eines Objektes sich über diese Beziehungen auf andere Objekte auswirkt. Dadurch können Auswirkungen von Änderungen an Objekten bzw.

Elementen der Informationssystemebene (z.B. Ausfall einer Plattform oder Softwarekomponente) auf die Organisations- und Strategieebene vorhergesagt werden (Bottom-Up). Umgekehrt lassen sich aber auch die Auswirkungen von Veränderungen an der Geschäftsstrategie (z.B. Änderung des Kundenprozesses oder Einführung einer neuen Leistung) auf die Organisations- und Informationssystemebene abschätzen (Top-Down). Ein Beispiel für das Ergebnis einer Impact-Analyse findet sich in Abschnitt 6.1.6.

5.7 Export/Import und Dokumentation

Da die Unternehmensarchitektur vor allem als Kommunikationsbasis für Mitarbeitende aus unterschiedlichen Unternehmensbereichen dient, stellt der Export bzw. Import sowie die Dokumentation und Veröffentlichung der erstellten Modelle eine wichtige Funktionalität des Werkzeuges dar.

Die Import/Export-Komponente ermöglicht es, bestehende Modelle in ein von der Metamodellierungsplattform abhängiges Dateiformat (ADL-Format) zu exportieren bzw. Dateien in das Werkzeug zu importieren. Der Datei-Export kann zusätzlich zur Datensicherung dienen.

Darüber hinaus bietet die Import/Export-Komponente die Möglichkeit zur Dokumentation der Modelle. Die Modelle können in unterschiedliche Dateiformate (z.B. HTML-, XML-Dateien, RTF-Dateien für MS Word) überführt werden. Dadurch können beliebig definierbare Modellinhalte in Dokumente eingebunden und beispielsweise unternehmensweit im Intranet publiziert werden (vgl. Abbildung 79).

Das Erstellen von Dokumentationsdateien wird durch die Auswahl des entsprechenden Menüpunktes im Menü "Dokumentation" (z.B. "HTML-Generierung") gestartet. Nach Anklicken des Menüpunktes öffnet sich das Modellauswahlfenster, in welchem alle in der ADONIS-Datenbank gespeicherten Modelle aufgelistet werden. Darin können die zu dokumentierenden Modelle ausgewählt werden. Durch die Aktivierung der Option "Inklusive referenzierter Modelle" werden alle selektierten Modelle und die aus diesen Modellen referenzierten Modelle in die Dokumentation überführt. Durch Anklicken des Buttons "Referenzen" können die Einstellungen hierzu angepasst werden.

Bei der HTML-Generierung werden für die Referenzen zwischen den Modellen und Objekten entsprechende Links in den HTML-Seiten erzeugt, so dass eine einfache Navigation zwischen den einzelnen Dokumenten möglich ist.

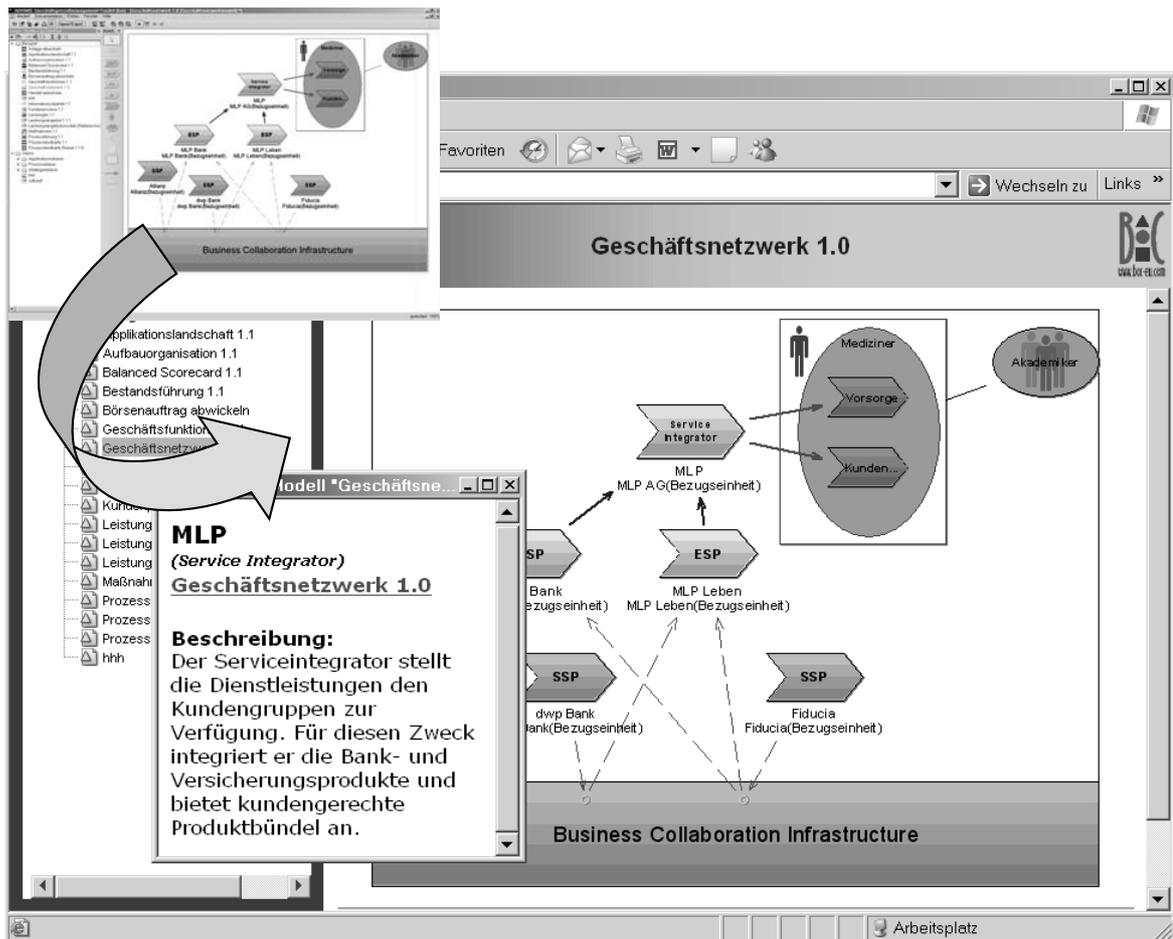


Abbildung 79: Dokumentation und Veröffentlichung im Intranet

6 Multiperspektivische Evaluation

Nachdem im vorherigen Kapitel die prototypische Realisierung des in Kapitel 4 präsentierten Ansatzes zur Modellierung der Unternehmensarchitektur beschrieben wurde, gilt es nun, den Ansatz im Hinblick auf seine Zweckmässigkeit, Vollständigkeit und Konsistenz zu überprüfen.

Da jede bisher in der Wirtschaftsinformatik verwendete Evaluierungsmethode bestimmte Stärken und Limitationen aufweist sowie auf unterschiedlichen Prämissen und Vorstellungen beruht, sollten Artefakte der Wirtschaftsinformatik-Forschung (z.B. Referenzmodelle, Metamodelle und Modelle) aus unterschiedlichen Perspektiven evaluiert werden.⁶⁴¹ FETTKE/LOOS schlagen hierfür einen Bezugsrahmen vor, der nach empirischen, deskriptiven, eigendisziplinären und fremddisziplinären Perspektiven unterscheidet (vgl. Abbildung 80).

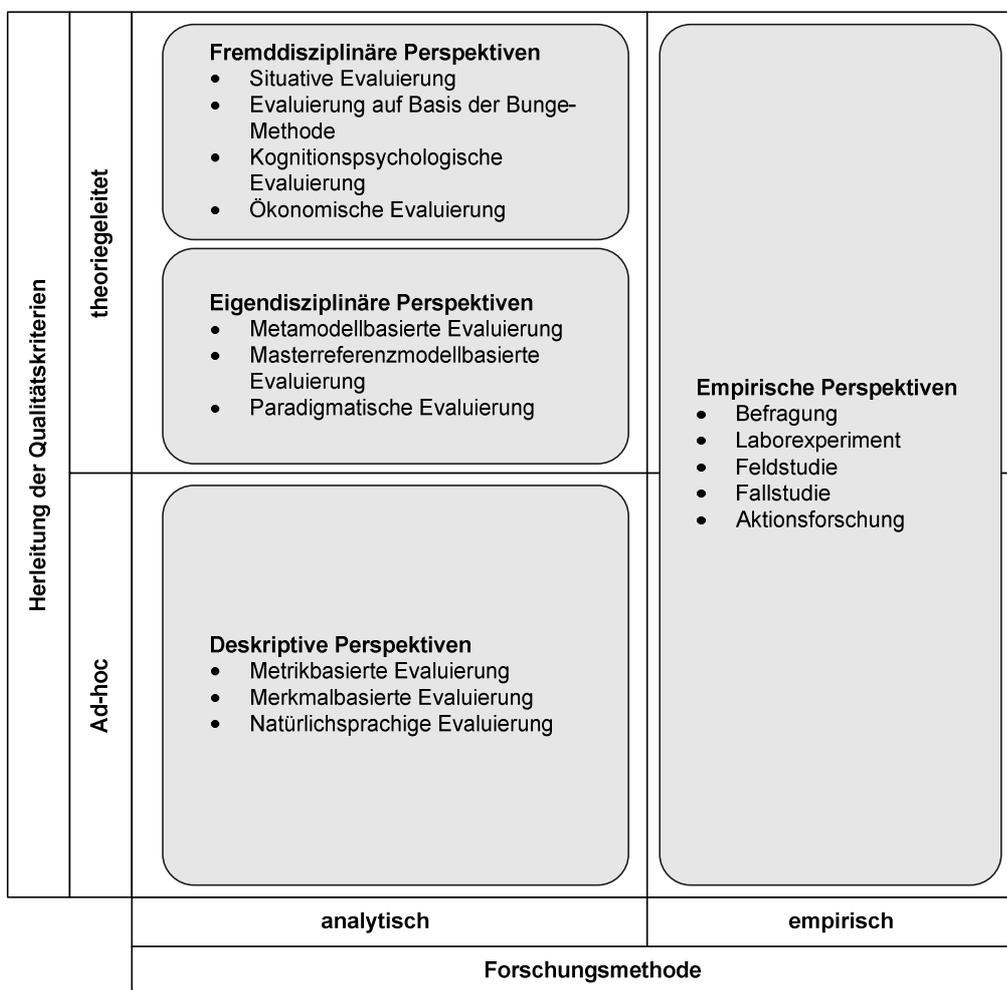


Abbildung 80: Multiperspektivische Evaluation⁶⁴²

⁶⁴¹ Vgl. Fettke/Loos (2004a), S. 1.

⁶⁴² Vgl. Fettke/Loos (2003), S. 2946.

Dieser Bezugsrahmen bezieht sich zwar auf die Evaluierung von Referenzmodellen, kann aber auch für die Evaluierung anderer Artefakte, wie beispielsweise Methoden und Metamodelle, herangezogen werden. Zudem kann das in dieser Arbeit konstruierte Metamodell als eine Art Referenz-Metamodell (Empfehlung) betrachtet werden, da es wesentliche Metaentitätstypen für die Abbildung von Unternehmensarchitekturen beinhaltet und für den Einsatz in einem unternehmensindividuellen Umfeld gegebenenfalls modifiziert werden muss.⁶⁴³

Im Folgenden wird der Versuch unternommen, zumindest ansatzweise eine multiperspektivische Evaluierung vorzunehmen. Die Evaluierung des in dieser Arbeit vorgestellten Ansatzes sowie dessen Metamodells erfolgt primär anhand einer Fallstudie (empirische Perspektive) sowie anhand von vordefinierten Kriterien (deskriptive Perspektive). Darüber hinaus wird versucht, die Metaentitätstypen und deren Beziehungen mit den Metamodellelementen der in Kapitel 3 betrachteten Ansätze zu vergleichen, um mögliche Lücken oder Redundanzen zu identifizieren (eigendisziplinäre Perspektive). Auf die Evaluierung aus Sicht einer fremddisziplinären Perspektive wird in dieser Arbeit verzichtet.

Die Abbildung des Metamodells in einem Metamodellierungswerkzeug hat bereits gezeigt, dass dessen Elemente und Beziehungen ausreichend formal spezifiziert und konsistent sind. Der daraus entstandene Software-Prototyp wurde anschliessend anhand einer Fallstudie getestet. Die Zielstellung einer Fallstudie ist im Allgemeinen, eine konkrete Modellierungssituation in einer Organisation bzw. einem Unternehmen zu einem bestimmten Zeitpunkt zu erfassen und zu dokumentieren.⁶⁴⁴ Dabei wird eine bestimmte Problemstellung der (Referenz-)Modellierung praktisch gelöst. Der Inhalt der Fallstudie ist die umfassende Dokumentation der Problemstellung, des Lösungswegs sowie der resultierenden Ergebnisse. Die Fallstudie dieser Arbeit soll zudem zeigen, dass der Ansatz dieser Arbeit in der Praxis gegebene Sachverhalte widerspiegelt, existierende Probleme adressiert und somit Praxisrelevanz besitzt.

Die Ergebnisse von Fallstudien sind zwar nicht objektiv nachvollziehbar und nur bedingt generalisierbar, da der betrachtete Anwendungsfall meist nicht repräsentativ ist und die Ergebnisse der subjektiven Interpretation des einzelnen Forschers unterliegen.⁶⁴⁵ Trotz dieser Nachteile ermöglicht die Evaluierung mittels Fallstudien aber die Gewinnung einer Vielzahl verschiedener subjektiver Eindrücke und Informationen. So ist es beispielsweise möglich,

⁶⁴³ Die Anwendung eines Referenzmodells, einer Methode oder eines Metamodells erfolgt im Anschluss an den Konstruktionsprozess im jeweiligen unternehmensspezifischen Kontext. Sie umfasst die Modifikationen aufgrund unternehmensspezifischer Anforderungen, die durch manuelle Modifikationen durch den Anwender vorgenommen werden. Vgl. *Schütte* (1998), S. 318 und *Wortmann* (2005), S. 100. Vgl. hierzu auch Abschnitt 2.2.2.

⁶⁴⁴ Vgl. *Fettke/Loos* (2004a), S. 18.

⁶⁴⁵ Vgl. *Fettke/Loos* (2004a), S. 19.

bestimmte Modellierungskonzepte im Hinblick auf ihre Konsistenz zu untersuchen bzw. ihre Realisierbarkeit und Nützlichkeit zu demonstrieren.⁶⁴⁶

Die Dokumentation der Fallstudie orientiert sich in dieser Arbeit an der Methode Promet Business Engineering Case Studies (BECS). Die wesentlichen Bestandteile einer Fallstudie sind demnach:⁶⁴⁷

- Unternehmen: allgemeine Eckdaten des betrachteten Unternehmens,
- Ausgangssituation und Problemstellung: Beschreibung der bisherigen Lösung/des bisherigen Vorgehens der generellen Problemstellung sowie der daraus resultierende Leidensdruck,
- Projekt und neue Lösung: Beschreibung des Projekts (Vorgehen, Ziele usw.) sowie der wesentlichen Ergebnisse,
- Einordnung in den Kontext und Fazit: Beschreibung, in welchem Kontext die Fallstudie in Bezug auf den in dieser Arbeit präsentierten Ansatz steht.

Die Informationen zu der Fallstudie basieren primär auf Interviews mit Architektur-Verantwortlichen des betrachteten Unternehmens (vgl. Anhang C) sowie zur Verfügung gestellten Dokumenten.

Neben der Fallstudie wird eine metamodellbasierte Evaluierung vorgenommen. Dabei werden die Metaentitätstypen und Beziehungen mit einem so genannten Master-Metamodell verglichen. Eine derartige Analyse erlaubt die Überprüfung der Vollständigkeit und Redundanzfreiheit der verwendeten Konzepte sowie die Identifikation möglicher Spezialisierungen bzw. Generalisierungen.⁶⁴⁸

Ergänzend zu den beiden zuvor genannten Evaluierungsmethoden erfolgt eine natürlich-sprachliche und merkmalsbasierte Evaluierung des in dieser Arbeit präsentierten Metamodells. Bei einer natürlich-sprachlichen Evaluierung werden die Charakteristika, Stärken und Schwächen eines Modells ermittelt und ausschliesslich verbal beschrieben.⁶⁴⁹ Die Evaluierung kann mehr oder weniger strukturiert erfolgen. Diese Art der Evaluierung ist ebenfalls stark subjektiv geprägt, es können dadurch aber spezielle qualitative Aspekte leichter zum Ausdruck gebracht werden. Eine merkmalsbasierte Evaluierung definiert zudem eine Menge von Merkmalen, anhand derer Modelle charakterisiert werden können. Die verwendeten Evaluierungsmerkmale werden nicht theoretisch abgeleitet, sondern ad hoc eingeführt.⁶⁵⁰ Da bisher keine

⁶⁴⁶ Für die Anwendung von Fallstudien zur Evaluation von Methoden und entsprechenden Werkzeugen vgl. z.B. auch *Kitchenham et al.* (1995).

⁶⁴⁷ Vgl. *Senger/Österle* (2004).

⁶⁴⁸ Vgl. hierzu *Fettke/Loos* (2004a), S. 10-11.

⁶⁴⁹ Vgl. *Fettke/Loos* (2004a), S. 7.

⁶⁵⁰ Vgl. *Fettke/Loos* (2004a), S. 8.

allgemeingültigen Merkmale vorhanden sind, wird auf die in Abschnitt 2.6 genannten Kriterien zurückgegriffen. Da diese auf Basis der genannten Ziele und treibenden Kräfte der Unternehmensarchitektur hergeleitet wurden, sollte deren Bedeutung weitgehend nachvollziehbar sein.

6.1 Fallstudie bei der Winterthur Group

Die Winterthur verfügt über langjährige Erfahrungen bei der Entwicklung und der Pflege von Architekturen in unterschiedlichen Bereichen, wie beispielsweise Informationssystem-, Prozess- und Sicherheitsarchitekturen. Diese Fallstudie untersucht das von der Winterthur entwickelte Metamodell für die Abbildung der Unternehmensarchitektur. Dabei wird zunächst das Unternehmen anhand einiger Eckdaten charakterisiert. Anschliessend wird darauf eingegangen, welche Ausgangssituation bzw. Problemstellung die Winterthur dazu veranlasst hat, ein Metamodell zu entwickeln, und welchen Nutzen und Zweck sie sich dadurch erwartet. Daraufhin wird das Metamodell der WGR überblicksartig beschrieben und mit dem Metamodell des in dieser Arbeit präsentierten Ansatzes verglichen. Dabei werden die Metaentitätstypen der beiden Metamodelle einander gegenübergestellt und zugeordnet, um Übereinstimmungen sowie mögliche Redundanzen oder Lücken zu identifizieren. Ausserdem wird anhand eines Pilot-Projektes gezeigt, wie die Informationen über die Unternehmensarchitektur der Winterthur mit Hilfe des in dieser Arbeit entwickelten Prototyps erfasst und analysiert werden können. Zuletzt folgt eine kurze Zusammenfassung der Ergebnisse dieser Fallstudie.

6.1.1 Allgemeine Informationen zur Winterthur Group

Die Winterthur Group (WGR) ist eine Schweizer Versicherungsgesellschaft mit Hauptsitz in Winterthur.⁶⁵¹ Das Angebot der international tätigen Gruppe umfasst eine breite Palette von Personen-, Sach- und Haftpflichtversicherungslösungen sowie Lebensversicherungs- und Pensionskassenlösungen für Privat- und Unternehmenskunden. Rund 19000 Mitarbeiterinnen und Mitarbeiter arbeiten weltweit bei der Winterthur Group, davon sind über 70% ausserhalb der Schweiz beschäftigt.

Die Winterthur Group gliedert sich in die zwei Segmente Life & Pensions und Non-Life. Das Segment Life & Pensions bietet Versicherungsprodukte für Firmenkunden sowie Lebensversicherungs- und Rentenprodukte für Privatkunden an. Das Segment Non-Life deckt die Versicherungsbedürfnisse von Privaten sowie kleineren und mittleren Unternehmen ab. Es bietet eine umfangreiche Palette von Versicherungsprodukten wie Motorfahrzeug-, Feuer-, Sach- und allgemeine Haftpflichtversicherung sowie Unfall- und Krankenversicherungslösungen an. In den beiden Segmenten Life & Pensions und Non-Life setzt die Winterthur Group je nach Markt Agenten, Broker, Banken oder Direktvertriebskanäle ein.

⁶⁵¹ Vgl. im Folgenden *Winterthur* (2005).

Die Winterthur Group ist in die vier regionalen Einheiten Schweiz, Deutschland, Market Group 1 und Market Group 2 aufgeteilt. Die Market Group 1 umfasst Grossbritannien, Spanien, Belgien, Tschechische Republik, Polen, Ungarn und die Slowakische Republik. Die Market Group 2 umfasst die USA, Japan, Hong Kong, Niederlande, Kanada, Taiwan und Indonesien.

Seit 1997 ist die Winterthur Group eine hundertprozentige Tochtergesellschaft der Credit Suisse Group. Ende 2004 hat sich die Credit Suisse Group dafür entschieden, die Winterthur Group auf den Börsengang vorzubereiten.

Die folgenden Ausführungen beziehen sich im Wesentlichen auf den IT-Bereich der Winterthur Schweiz. Dieser erbringt jedoch auch Dienstleistungen an die ausländischen Markteinheiten.

WINTERTHUR GROUP (WGR)	
Gründung	1875 Gründung der «Schweizerischen Unfallversicherungs Aktiengesellschaft» in Winterthur. Im gleichen Jahr Expansion nach Deutschland, Elsass-Lothringen, Belgien, Niederlande, Österreich-Ungarn und Dänemark.
Firmensitz	Winterthur, Schweiz
Branche	Versicherungen
Geschäftsfelder	Personen-, Sach- und Haftpflichtversicherungen, Lebensversicherungen und Pensionsvorsorge
Firmenstruktur	Seit 1997 ist das Unternehmen ein Teil der Credit Suisse Group Regional gegliedert in die Gebiete Europa, Nordamerika und Asien
Homepage	http://www.winterthur.com
Geschäftsvolumen	27.3 Mrd. CHF
Bruttoprämien	21.4 Mrd. CHF
Reingewinn	699 Mio. CHF
Mitarbeiter	19020
Kunden	Privat- und Firmenkunden

Tabelle 23: Fakten zur Winterthur Group⁶⁵²

6.1.2 Ausgangssituation und Problembeschreibung

Die IT-Organisation und Applikationslandschaft der WGR sind durch die langjährige Selbstständigkeit der einzelnen Unternehmensbereiche sehr dezentral geprägt und weisen heute daher folgende Eigenschaften auf:⁶⁵³

- Die Applikationslandschaft der WGR umfasst eine Vielzahl heterogener Applikationen, die in der Vergangenheit grösstenteils selbst entwickelt wurden. Die heutigen Architek-

⁶⁵² Vgl. Winterthur (2005).

⁶⁵³ Vgl. hierzu auch Wortmann (2005), S. 65.

turprinzipien betonen dagegen vor allem den Einkauf von Applikationen. Der Entwicklungsprozess ist entsprechend darauf ausgerichtet und sieht einen Subprozess „Solution Evaluation“ (SOP) vor. Gerade in den Bereichen Finanzen, Human Resources und Asset Management kommt in Zukunft Standardsoftware zum Einsatz.

- Die Applikationslandschaft weist eine hohe Komplexität auf. Eine Vielzahl unterschiedlicher Applikationen und verteilter, mehrschichtiger Softwaresysteme erschweren die Identifikation von Konfigurationsproblemen und von Abhängigkeiten zwischen einzelnen Softwarekomponenten.

Im Rahmen einer Restrukturierung im Jahre 2003 wurde die stark dezentrale Organisation der IT durch eine zentrale Entscheidungsfindung und -kontrolle abgelöst. Zentrale Einheiten wie „Architektur“ oder „Engineering“ können nun Standards IT-übergreifend durchsetzen. Ausserdem wurden die IT-Prozesse mit der Einführung der IT Infrastructure Library (ITIL) klar definiert und standardisiert. Im Bereich der Softwareentwicklung wird mit der Definition des „Solution Evaluation Process“ (SOP) die Entwicklung und Überführung von Software in die Produktion standardisiert. Das Projekt wurde im März 2006 erfolgreich abgeschlossen und alle Betroffenen werden seitdem mit grossem Aufwand geschult. Wesentliche Treiber dieser Entwicklungen waren vor allem regulatorische Anforderungen, wie beispielsweise der Sarbanes-Oxley Act.

Innerhalb der WGR werden jeden Tag im Rahmen der Durchführung verschiedenster IT-Prozesse⁶⁵⁴ durch den IT-Bereich sowie teilweise durch die Fachbereiche Ergebnisse produziert und bearbeitet. Verteilt über das ganze Unternehmen geschieht dies oftmals unkoordiniert, das heisst einzelne Organisationseinheiten und Projekte pflegen oft individuelle Verständnisse, wie ihre Ergebnisse aufgebaut sind. Bestenfalls sind solche Ad-hoc-Verständnisse dokumentiert (Glossar, Richtlinien, Templates etc.), oft befinden sie sich aber lediglich in den Köpfen der Mitarbeiter. Dies kann lokal und zeitlich begrenzt gut funktionieren. Es liegt aber auf der Hand, dass ein solcher Umgang mit dem im Unternehmen verfügbaren Know-how höhere Kosten verursacht. Es können folgende Probleme im Unternehmen identifiziert werden:

- Eine grosse Vielfalt von Ausprägungen ähnlicher Ergebnisse wird gepflegt.
- Diskussionen um dieselben Themen wiederholen sich laufend.
- Zusammenhänge zwischen Ergebnissen werden nicht erkannt.
- Inkonsistenzen und Redundanzen treten auf.

⁶⁵⁴ Mittlerweile ist der „Solution Evaluation Process“ (SOP) für den IT-Bereich verbindlich. Für die Fachbereiche ist der Projekt Management Prozess, der Teil des SOP ist, verbindlich.

Um diesen offensichtlichen Mängeln in der IT entgegenzuwirken, ist es nach Meinung des IT-Bereichs der WGR daher notwendig, alle Ergebnisse

- als Output von Prozessen zu definieren (Welche Ergebnisse werden produziert?),
- mit einer gemeinsamen Struktur als Basis abzubilden (Wie sehen die Ergebnisse aus?),
- in einem Repository, das nach dieser vorgegebenen Struktur konfiguriert worden ist, zu speichern (Wo werden die Ergebnisse abgelegt?).

Die derart gesammelten Informationen beschreiben in ihrer Gesamtheit, wie das Unternehmen aufgebaut ist und funktioniert. Sie beschreiben somit dessen Architektur. Die Struktur, welche den gesammelten Informationen über das Unternehmen zugrunde liegt, soll durch das WGR Enterprise Architecture Metamodel definiert (nachfolgend kurz Metamodell genannt) werden.

In Zukunft soll folglich für das Erfassen und Verwalten von Informationen über das Unternehmen folgendes gelten:

- Das Metamodell dient als einzige Referenz, um die Strukturen für das Erfassen der Informationen festzulegen.
- Wenn das entsprechende Repository nicht verfügbar ist, kann das Verwalten der Informationen auch mit Dokumenten erfolgen. Dies sollte langfristig gesehen eine Übergangslösung sein, denn auf diese Weise gesammelte Informationen sind nicht in der gleichen Masse im Unternehmen integriert und verfügbar wie Informationen in einem Repository. Wichtig ist, dass für solche Dokumente Vorlagen (Templates) verwendet werden, die vom Metamodell abgeleitet worden sind. So wird wenigstens die gemeinsame Syntax und Semantik gewährleistet.
- Die Strukturen des Repository sowie der Dokumentenvorlagen (Templates) müssen mit dem Metamodell übereinstimmen. Die Verantwortlichen für das Werkzeug und die Vorlagen haben also dafür zu sorgen, dass sie entweder einen vorhandenen Ausschnitt des Metamodells für die entsprechende Konfiguration verwenden oder ihre Bedürfnisse mit den Metamodell-Verantwortlichen absprechen, damit das Metamodell entsprechend erweitert werden kann.

6.1.3 Ein zentral gepflegtes Metamodell als Lösungsansatz

Durch die Verwendung eines gemeinsamen Metamodells sowie eines entsprechenden Repositories soll in Zukunft die Beantwortung von Fragen der folgenden Art möglich sein:

- Welche Applikationen unterstützen welche Geschäftsfunktionen/Prozesse?
- Welche Rollendefinitionen gibt es?
- Welche Datenobjekte werden wo verwendet?

- Welche Prozesse funktionieren nicht mehr durchgängig, wenn eine Komponente/Applikation ausfällt?
- Welche Softwarekomponenten sind davon betroffen, wenn ein Infrastrukturelement (z.B. Server) ersetzt werden muss.

Zudem können Datenbestände aus verschiedenen nach dem gleichen Metamodell strukturierten Repositorys einfacher zusammengeführt werden. Um eine professionelle IT zu betreiben, ist es essenziell, solche Fragen zuverlässig und effizient beantworten zu können. Nur so können die Informationen kontrolliert bewirtschaftet werden.

Aus der zuvor dargelegten Positionierung des Metamodells und des entsprechenden Repository lässt sich deren Zweck ableiten. Das Metamodell soll die Strukturen der Fachbereiche sowie des IT-Bereichs übergreifend darstellen. Das Ziel ist die Erstellung einer zentralen, standardisierten und zusammenhängenden Sichtweise über das Unternehmen als Ganzes – zumindest über jenen Teil der Fachbereiche, der mit der IT in direktem Zusammenhang steht, sowie die IT selbst. Folglich werden die Elemente, welche die IT-Unterstützung der Fachbereiche betreffen, systematisch erfasst, vereinheitlicht und zu einer konsistenten Einheit zusammengeführt.

Das Metamodell beschreibt demnach die aus Sicht der Fachbereiche und der IT relevanten

- Elemente, Beziehungen, Attribute,
- deren Semantik,
- Qualitätsmerkmale / Mindestanforderungen.

Des Weiteren sollen die IT-Prozesse zielgerichtet unterstützt werden. Das Metamodell repräsentiert die statische Sicht auf die Elemente der Fachbereiche und der IT. Es dient als Basis, um Repositorys zu strukturieren. Die derart strukturierten Repositorys werden durch Prozesse verwaltet und genutzt. Über die gesamte WGR-IT-Prozesslandschaft sorgt ein zentrales Metamodell somit für eine einheitliche, konsistente Pflege der IT-relevanten Daten.

Das Metamodell ist somit eine Basis für

- Qualitätssicherung und Konsistenz zwischen verschiedensten Initiativen,
- integriertes Denken und Handeln (in den verschiedensten Kontexten)
- und verteilte Unternehmens-Repositorys.

Das Metamodell dient als Kommunikationsmittel. Mit dem Metamodell als Bauplan können Kontexte aufgezeigt werden (Wie hängen einzelne Metamodell-Elemente zusammen?), Landkarten oder Repositorys positioniert werden (Prozesslandkarte, Subdomänen-Beschreibungen, Service-Repository) sowie Checklisten, Templates und Strukturen für Repositorys abgeleitet werden.

Das Metamodell ist folglich ein integraler Bestandteil der Unternehmensarchitektur und dient dazu, eine gemeinsame Terminologie herbeizuführen und eine Basis für die Strukturierung von Repositories zu bilden.

6.1.4 Metamodell der Winterthur Group

Das Metamodell der WGR soll die Schlüsselemente der Fachbereiche und des IT-Bereichs sowie deren Beziehungen übergreifend darstellen.⁶⁵⁵ Die folgende Abbildung 81 gibt eine Übersicht über die wesentlichen Elemente des Metamodells sowie deren Zusammenhänge.

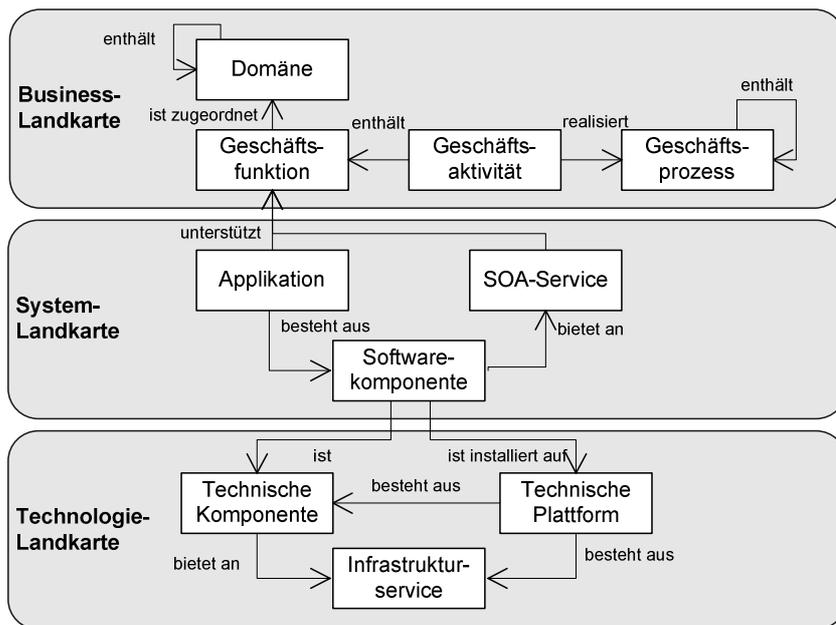


Abbildung 81: Vereinfachtes Metamodell der Winterthur Group

Ein Geschäftsprozess kann in detailliertere Geschäftsprozesse aufgeteilt werden und wird durch Geschäftsaktivitäten realisiert. Eine Geschäftsaktivität repräsentiert eine abgeschlossene Bearbeitungseinheit (logische Transaktion) und kann sowohl ausführend (operativ) als auch führend (dispositiv) sein. Eine Geschäftsaktivität existiert nur im Kontext eines Geschäftsprozesses. Sie enthält Geschäftsfunktionen als elementare Verarbeitungseinheiten. Eine Geschäftsfunktion ist einer (Sub-)Domäne zugeordnet. Sie kann Funktionalitäten für mehrere Geschäftsaktivitäten erfüllen (z.B. Partner suchen) und kann IT-unterstützt sein (durch Softwarekomponenten und Applikationen). Eine Applikation besteht aus mehreren Softwarekomponenten und stellt selbst eine Softwarekomponente dar. Eine Softwarekomponente bietet Services an (zur Unterstützung von Geschäftsfunktionen). Sie kann auf mehreren technischen Plattformen installiert sein und ist als technische Komponente ein Bestandteil einer technischen Plattform. Eine technische Komponente ist eine speziell für die Wiederverwendung entwickelte Softwarekomponente, die technische (von der reinen Geschäftsfunktionalität los-

⁶⁵⁵ Vgl. im Folgenden *Bernet* (2005).

gelöste) Dienste implementiert. Eine technische Plattform besteht aus technischen Komponenten und den dazugehörigen Infrastrukturservices. Sie bietet eine technische Infrastruktur (Hardware, Operating System Builds) mit den Services an, die benötigt werden, um die Softwarekomponenten lauffähig zu machen.

Das Metamodell ist in die Bereiche Business-Landkarte, System-Landkarte und Technologie-Landkarte untergliedert. Abbildung 81 gibt einen Überblick über die Inhalte der einzelnen Metamodell-Bereiche sowie bisher definierte thematische Metamodell-Sichten.

- **Übersicht:** Vereinfachte, selektive Sicht auf die Kernelemente des Metamodells im Kontext der Metamodell-Bereiche. Diese Sicht soll den wesentlichen Inhalt und Aufbau des Metamodells auf einen Blick erkennbar machen.

	Thematische Sichten
Prozesse	<ul style="list-style-type: none"> • Geschäftsprozess-Sicht • Aufbauorganisations-Sicht • Prozess-Rollen-Sicht • Servicevertrag-Sicht
Informationssysteme	<ul style="list-style-type: none"> • IT-Servicesicht • Applikationsportfolio-Sicht • Plattform-Sicht
Inhalt dient unter anderem zur Beschreibung von	
Business-Landkarte	<ul style="list-style-type: none"> • Geschäftsprozessen • Geschäftsregeln • Geschäftsfunktionen • Organisationsstrukturen
System-Landkarte	<ul style="list-style-type: none"> • Business-Komponenten • Business-Applikationen • Applikationslandkarten • Datenelementen
Technologie-Landkarte	<ul style="list-style-type: none"> • Technischen Komponenten • Technischen Applikationen • Laufzeitumgebungen des Betriebs • Technischen Applikationsplattformen

Abbildung 82: Sichten und Inhalte des Metamodells der WGR

- **Übersicht – erweitert:** Bereichsübergreifende Sicht der Kernelemente. Diese Sicht enthält zwar nur die Kernelemente des Metamodells, ist aber schon zu komplex, um das Metamodell auf einen Blick zu erfassen. Der Zweck dieser Sicht ist es, auf einem hohen Level eine detailliertere Sicht zu bieten.
- **Thematische Detailsichten:** Sichten, welche einen bestimmten Aspekt darstellen (z.B. Geschäftsprozesse, Services, Aufbauorganisation, Domänen, Servicevertrag)

Da das Metamodell ein zentraler Bestandteil der Unternehmensarchitektur ist und, ebenso wie die später darauf basierenden Modelle, als Kommunikationsmittel dienen soll, wird dessen Dokumentation im Intranet veröffentlicht (vgl. Abbildung 83). Damit ist es für den IT-Bereich und für die unterschiedlichen Fachbereiche an einem zentralen Ort zugänglich.

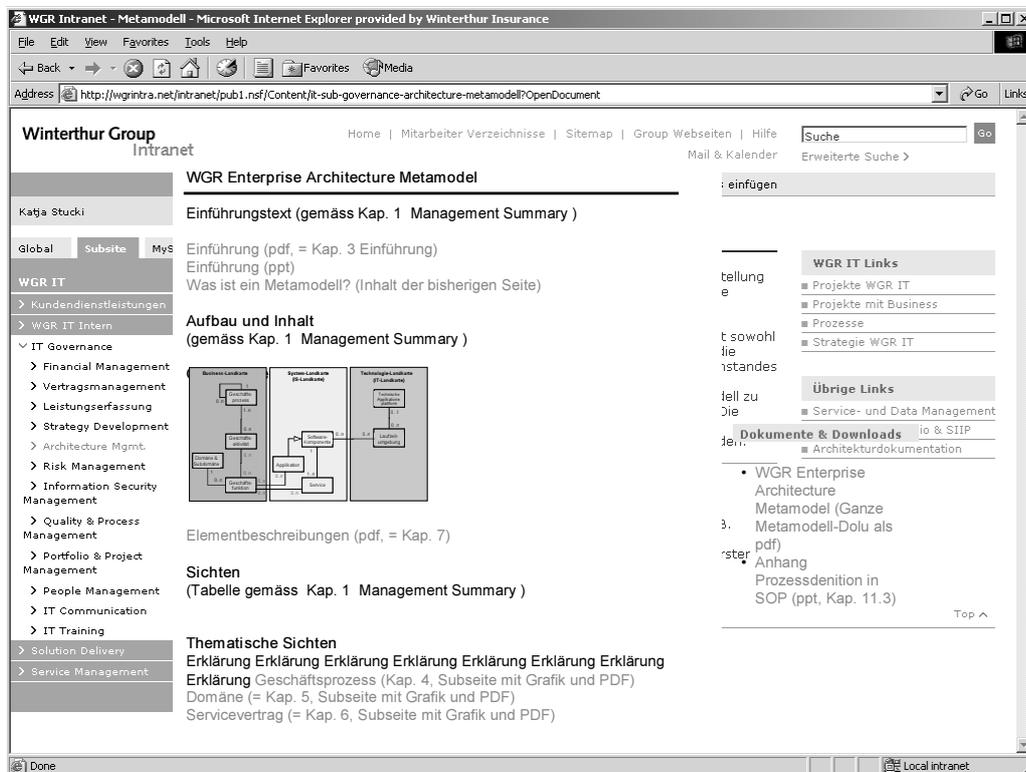


Abbildung 83: Dokumentation des WGR-Metamodells im Intranet

6.1.5 Vergleich der Metamodelle

Die thematischen Sichten sowie die Inhalte der Bereiche Business-, System- und Technologie-Landkarte des WGR-Metamodells überschneiden sich grösstenteils mit den Sichten und Inhalten der Ebenen „Organisationsebene“, „Applikationsebene“ und „Softwarekomponenten- und Datenstrukturebene“ des in dieser Arbeit präsentierten Metamodells (vgl. dazu Abbildung 32). Im Vergleich zum Metamodell des in dieser Arbeit präsentierten Ansatzes definiert das Metamodell der WGR keine Metaentitätstypen zur Abbildung der Geschäftsstrategie, wie beispielsweise Geschäftsnetzwerkrollen, Unternehmensziele und Erfolgsfaktoren, Marktleistungen, Kundengruppen etc. Auf der Organisationsebene bzw. im Bereich der Business-Landkarte bieten beide Metamodelle weitestgehend die gleichen Sichten und Metaentitätstypen. Dem Metamodell der WGR fehlen lediglich Elemente zur Beschreibung der Informationslandkarte sowie der Prozessvision.

Im Folgenden werden die Metaentitätstypen der einzelnen Sichten des WGR-Metamodells detaillierter betrachtet und den Metaentitätstypen des in dieser Arbeit definierten Metamodells

gegenübergestellt, um dadurch Übereinstimmungen sowie Redundanzen oder Lücken identifizieren zu können.

Geschäftsprozess-Sicht

Die folgende Tabelle 24 zeigt einen Vergleich zwischen den wesentlichen Metaentitätstypen der Geschäftsprozess-Sicht des WGR-Metamodells sowie korrespondierenden Metaentitätstypen des in dieser Arbeit vorgestellten Ansatzes (vgl. dazu die Metamodelle in Abschnitt 4.2.2).

Metaentitätstypen Metamodell WGR/ Ansatz dieser Arbeit		Korrespondierende Metaentitätstypen dieser Arbeit														
		Aktivität	Applikationsdomäne	Ereignis	Erfolgsfaktor	Führungsprozess	Geschäftsfunktion	Geschäftsobjekt	Kennzahl	Leistungsprozess	OE des PM	Prozess	Prozessleistung	Prozessrolle	Techn. Rollenträger	Unterstützungsprozess
Metaentitätstypen der Geschäftsprozess-Sicht	(Sub-)Domäne		X													
	Erfolgsfaktor				X											
	Geschäftsaktivität	X														
	Geschäftsereignis			X												
	Geschäftsfunktion						X									
	Geschäftsobjekt							X								
	Geschäftsprozess										X					
	Kerngeschäftsprozess									X						
	KPI								X							
	Owner										X					
	Prozessleistung												X			
	Prozessrolle													X		
	Sachmittel														X	
	Steuernder GP					X										
	Unterstützender GP															X

Tabelle 24: Vergleich Metaentitätstypen Ablauforganisation

Daraus ist ersichtlich, dass sich jeder Metaentitätstyp der Geschäftsprozess-Sicht des WGR-Metamodells auf einen Metaentitätstyp des Metamodells dieser Arbeit abbilden lässt. Auch die Beziehungstypen zwischen den einzelnen Metaentitätstypen sind weitgehend deckungsgleich oder lassen sich indirekt über mehrere Beziehungstypen herstellen. So existiert beispielsweise im Metamodell dieser Arbeit kein Beziehungstyp „ist zugeordnet“ zwischen den Metaentitätstypen „Geschäftsobjekt“ und „Domäne“. Entsprechend dem Metamodell dieser Arbeit korrespondiert aber jedes Geschäftsobjekt mit einem Informationsobjekt, welches von einer Applikation verwaltet wird, welche wiederum einer bestimmten Domäne zugeordnet ist (vgl. Abbildung 64). Somit ist letztlich auch das Geschäftsobjekt einer Domäne zugeordnet.

Aufbauorganisations-Sicht

Tabelle 25 zeigt die wesentlichen Metaentitätstypen der Aufbauorganisations-Sicht sowie korrespondierende Metaentitätstypen des Metamodells dieser Arbeit. Auch hier stimmen die Metaentitätstypen der beiden Metamodelle fast vollständig überein (vgl. hierzu auch Abbildung 42). Die Beziehungstypen sind ebenfalls direkt oder indirekt in beiden Metamodellen vorhanden. Die Verbindung zwischen Aufbau- und Ablauforganisation wird in beiden Metamodellen über den Metaentitätstyp „Prozessrolle“ hergestellt.

Metaentitätstypen Metamodell WGR/ Ansatz dieser Arbeit		Organisationsebene											
		Organisationseinheit	Org. Rollenträger	Qualifikation	Person	Primäre OE	Prozessrolle	Rollenträger	Sekundäre OE	Standort	Stelle	Stellentyp	Techn. Rollenträger
Aufbauorganisations-Sicht	Organisationseinheit	X											
	Org. Rollenträger		X										
	Person				X								
	Primäre OE					X							
	Prozessrolle						X						
	Rollenträger							X					
	Sekundäre OE								X				
	Skill			X									
	Standort									X			
	Stelle										X		
	Stellentyp											X	
	Techn. Rollenträger												X

Tabelle 25: Vergleich Metaentitätstypen Aufbauorganisation

IT-Service- und IT-Servicevertrag-Sicht

Korrespondierende Metamodellelemente der IT-Servicesicht sind im Metamodell dieser Arbeit nicht enthalten. Dies ist darauf zurückzuführen, dass die Gestaltung der Informationssystemebene, entsprechend dem in dieser Arbeit vorgeschlagenen Ansatz, sich nach der optimalen Strukturierung von Applikationen richtet.

Die Bewirtschaftung der Applikationslandschaft in einem langfristigen Sinne steht bei Standards, die eine serviceorientierte Perspektive verfolgen, wie beispielsweise ITIL, eher im Hintergrund. Fraglich ist deshalb, welche Ansatzpunkte solche Standards bieten, um auch den langfristigen Betrieb zu unterstützen.⁶⁵⁶ Die Definition der IT-Service- und IT-Servicevertrag-Sicht hat sich zum Zeitpunkt dieser Fallstudie auch bei der WGR noch in einer Konsolidierungsphase befunden. Deshalb wurden diese Sichten nicht in das Metamodell dieser Arbeit

⁶⁵⁶ Vgl. Schelp et al. (2004).

integriert und im Prototyp abgebildet. Da aber in der Praxis in letzter Zeit verstärkt Konzepte zum Einsatz kommen, welche die Serviceorientierung betonen und mit ITIL ein De-facto-Standard zur Verfügung steht, der eine serviceorientierte Perspektive verfolgt, wurde in Abschnitt 4.4 darauf eingegangen, wie sich eine solche Perspektive in das hier präsentierte Metamodell integrieren lässt.

Application Portfolio und Technical Platform View

Die Informationssystemebene als die der Organisationsebene logisch nachfolgende Gestaltungsebene wird in dieser Arbeit in die zwei Unterebenen „Applikationsebene“ sowie „Softwarekomponenten und Datenstrukturen“ unterschieden. Die Metaentitätstypen dieser beiden Ebenen stimmen weitgehend mit den Metaentitätstypen der im WGR-Metamodell definierten Sichten „Application Portfolio“ und „Technical Platform View“ überein (vgl. hierzu die folgende Tabelle 26 und Abschnitt 4.3.2).

Metaentitätstypen Metamodell WGR/ Ansatz dieser Arbeit		Applikationen/Softwarekomponenten						
		Applikation	Betriebskomponente	Entwicklungstechnologie	Infrastrukturelement	Plattform	Schnittstelle	Softwarekomponente
Application Portfolio	Application	X						
	Application Platform					X		
	Business Component							X
	Development Environment			X				
	Runtime Environment		X					
	Software Component							X
	Technical Component							X
	Technical Platform					X		

Tabelle 26: Vergleich Metaentitätstypen Softwarekomponenten

Auf der Ebene der Softwarekomponenten differenziert das WGR-Metamodell zwischen verschiedenen Softwarekomponenten und Plattformen. Der Metaentitätstyp „Softwarekomponente“ wird weiter in die Metaentitätstypen „Geschäftskomponente“ und „technische Komponente“ spezialisiert. Eine Geschäftskomponente wird als eine Softwarekomponente definiert, die geschäftsbezogene SOA-Services (Service Oriented Architecture) anbietet. Eine technische Komponente stellt dagegen eine speziell für die Wiederverwendung entwickelte Softwarekomponente dar, die technische (von der reinen Geschäftsfunktionalität losgelöste) Dienste implementiert. Zusätzlich werden Applikations- und technische Plattformen unterschieden. Eine technische Plattform bietet eine technische Infrastruktur (Hardware, Operating System

Builds) mit den Services an, die benötigt werden, um die Softwarekomponenten einer Applikations-Plattform lauffähig zu machen.

Da es sich bei den zusätzlichen Metaentitätstypen lediglich um Spezialisierungen handelt, können diese durch das Vererbungskonzept in das Metamodell dieser Arbeit einfach integriert und im Prototyp abgebildet werden.

6.1.6 Abbildung mit Hilfe des Prototyps

Im Rahmen eines Pilotprojektes wurde der Prototyp zur Abbildung der Unternehmensstrukturen in der zentralen Architekturabteilung der WGR eingesetzt. Wie der vorhergehende Vergleich der Metamodelle gezeigt hat, existieren im Metamodell des in dieser Arbeit präsentierten Ansatzes bereits Metaentitäts- und Beziehungstypen, die weitestgehend mit den Metaentitäts- und Beziehungstypen des von der WGR erarbeiteten Metamodells korrespondieren. Es waren daher für den Einsatz des Prototyps bei der WGR nur geringe Anpassungen oder Erweiterungen notwendig. Überwiegend handelt es sich dabei um Attribute, die bestimmten Metaentitätstypen hinzugefügt werden mussten. Um bestimmte Fragestellungen beantworten zu können, mussten zudem spezielle Analyseabfragen und Beziehungstabellen definiert werden. Zusätzlich zu den Informationen auf Organisations- und Informationssystemebene, die auch im Metamodell der WGR definiert sind, wurden Informationen über die Geschäftsstrategie entsprechend dem in dieser Arbeit definierten Metamodell erhoben und mit den Modellen auf Organisationsebene in Beziehung gesetzt. In diesem Abschnitt werden exemplarisch einige Modelle vorgestellt, die mit Hilfe des Prototyps abgebildet wurden und Teile der Unternehmensarchitektur der WGR repräsentieren.

Geschäftsnetzwerk

Das Geschäftsnetzwerk bildet das Zusammenwirken von Unternehmen bzw. Geschäftseinheiten zur gemeinsamen Abdeckung eines bestimmten Kundenprozesses ab. Dabei wird jedem Unternehmen bzw. jeder Geschäftseinheit eine bestimmte Rolle zugeordnet und deren Leistungsbeziehungen spezifiziert. Abbildung 84 zeigt exemplarisch einen Ausschnitt des Geschäftsnetzwerks, in dem die WGR die Rolle eines Exclusive Service Provider (ESP) einnimmt.⁶⁵⁷ In diesem Geschäftsnetzwerk produziert die WGR die Leistung „Auto versichern“ für den Service Integrator „AutoScout24“. Dieser unterstützt ganzheitlich den Kundenprozess „Auto kaufen“, indem er Leistungen von unterschiedlichen Service Providern integriert. Die WGR trägt folglich mit ihrer Leistung dazu bei, dass ein Teil dieses Kundenprozesses abgedeckt werden kann. Ausserdem bietet die WGR ihre Leistung nicht nur über die Plattform von AutoScout24 an, sondern auch dem Endkunden direkt über unterschiedliche Vertriebskanäle (z.B. Internet, Telefon etc.).

⁶⁵⁷ Es können weitere Geschäftsnetzwerke identifiziert werden, in denen die WGR eine andere Rolle wahrnimmt oder mit anderen Geschäftspartnern in Beziehung steht und andere Kundenprozesse abgedeckt werden.

AutoScout24 ist ein elektronischer Marktplatz für Neu- und Gebrauchtwagen in der Schweiz. Auf der Internetseite des Marktplatzes sind gegenwärtig rund 93'000 Angebote online verfügbar. Darüber hinaus bietet AutoScout24 Informationen und Links zu Themen rund um das Auto und Unterstützung beim Autokauf an, wie beispielsweise Autobewertung, Finanzierung, Versicherung etc. AutoScout24 hat somit die Rolle eines Service Integrator und unterstützt den Kundenprozess des Autokaufs ganzheitlich.

Geschäftsnetzwerk

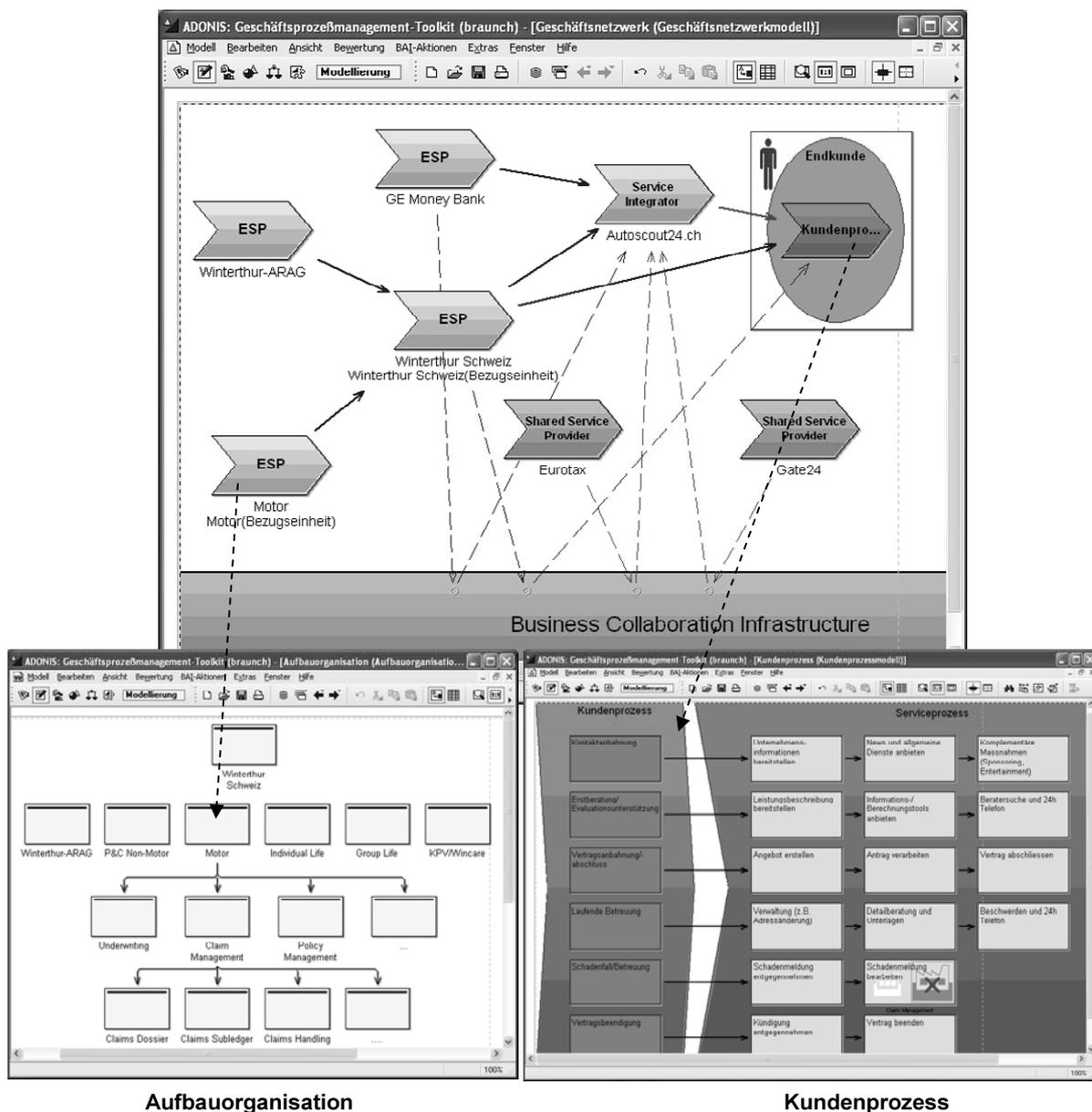


Abbildung 84: Geschäftsnetzwerk

Die WGR bietet auf der Plattform von AutoScout24 ihre KFZ-Versicherung und ihren KFZ-Rechtsschutz an. Andere Kooperationen hat AutoScout24 beispielsweise mit der GE Money Bank für die Finanzierung und das Leasing, mit Eurotax für die Autobewertung sowie mit

Gate 24 für die Suche in Kleinanzeigen. Der KFZ-Rechtsschutz ist eine Leistung der Winterthur-ARAG, einer selbständigen und unabhängigen Tochterfirma der Winterthur. Sie entstand 1998 durch den Zusammenschluss des deutschen Rechtsschutzversicherers ARAG (Schweiz) und der Winterthur-Rechtsschutz. Der Vertrieb der Rechtsschutzversicherungen erfolgt sowohl über die Verkaufsorganisation der Winterthur als auch über sieben regionale Geschäftsstellen der Winterthur-ARAG.

Ausgehend vom Geschäftsnetzwerkmodell können weitere Modelle referenziert werden. Der Teil des Kundenprozesses, der aus Sicht der WGR relevant ist, wird in einem eigenen Kundenprozessmodell detaillierter spezifiziert, in dem die einzelnen Aktivitäten des Kunden sowie die entsprechenden Serviceaktivitäten der WGR festgelegt werden. Des Weiteren verweisen die Rollen im Geschäftsnetzwerk auf Organisationseinheiten in der Aufbauorganisation, sofern diese vom Unternehmen selbst (Winterthur Group) bzw. von Geschäftseinheiten des eigenen Unternehmens (Geschäftseinheit „Motor“) wahrgenommen werden.

Kundenprozess

Abbildung 85 zeigt den Teil des Kundenprozesses „KFZ-Versicherung abschliessen“, der aus Sicht der Winterthur relevant ist, sowie die von der WGR für die Unterstützung der Kundenaktivitäten durchzuführenden Serviceaktivitäten. Das Kundenprozessmodell veranschaulicht einerseits, welche Ziele der Prozess beim Kunden verfolgt und wie der Prozess beim Kunden abläuft. Auf der anderen Seite veranschaulicht es, wie der Prozess im eigenen Unternehmen am Kundenbedarf ausgerichtet werden kann.

Die Kontaktabahnung findet in diesem Fall über den elektronischen Marktplatz von Auto-Scout24 statt. Nachdem der Kunde dort auf das Angebot der WGR gestossen ist, wird er sich zunächst auf der Internetseite der WGR Informationen über das Unternehmen und allgemeine Informationen zum Angebot besorgen. Nach der Kontaktabahnung folgt die Evaluationsunterstützung und Erstberatung. Der Kunde wird in der Regel eine detailliertere Leistungsbeschreibung anfordern, das Berechnungstool verwenden, eventuell einen Berater suchen und per Telefon oder E-Mail kontaktieren. Möchte der Kunde daraufhin einen Vertrag abschliessen, muss ein entsprechendes Angebot erstellt, der Antrag verarbeitet und der Vertrag abgeschlossen werden. Der Kundenprozess und damit die Beziehung zum Kunden ist aber mit dem Vertragsabschluss nicht beendet. Während der Vertragslaufzeit ist eine laufende Betreuung des Kunden durch die Verwaltung seiner Daten, Detailberatungen, das Zur-Verfügung-Stellen spezieller Unterlagen und die Entgegennahme von Beschwerden durchzuführen. Tritt der Schadenfall ein, muss die Schadenmeldung entgegengenommen und bearbeitet werden. Bei der Vertragsbeendigung wird die Kündigung entgegengenommen und der Vertrag beendet. Der Kundenprozess und damit die Beziehung zum Kunden endet somit aus Sicht der WGR, und es sind keine weiteren Serviceaktivitäten mehr notwendig.

Das Kundenprozessmodell enthält zwei wesentliche Arten von Referenzen. Zum einen muss jeder Serviceaktivität eine für deren Durchführung verantwortliche Organisationseinheit zu-

geordnet werden. Dabei kann es sich um interne sowie externe Einheiten handeln. Damit ist im Kundenprozessmodell ersichtlich, welche Serviceaktivitäten selbst (von der Winterthur bzw. einer ihrer Geschäftseinheiten) und welche von Partnern durchgeführt werden. Die Serviceaktivitäten „Schadenfall entgegennehmen“ und „Schadenfall bearbeiten“ werden beispielsweise von der Organisationseinheit „Claims Management“ verantwortet. Zum anderen wird jeder Serviceaktivität ein entsprechender zugrunde liegender Geschäftsprozess der Prozesslandkarte auf Organisationsebene zugeordnet. Der Serviceaktivität „Schadenmeldung bearbeiten“ liegt in diesem Fall der Geschäftsprozess „Schadenbearbeitung“ der Prozesslandkarte „Schaden“ zugrunde.

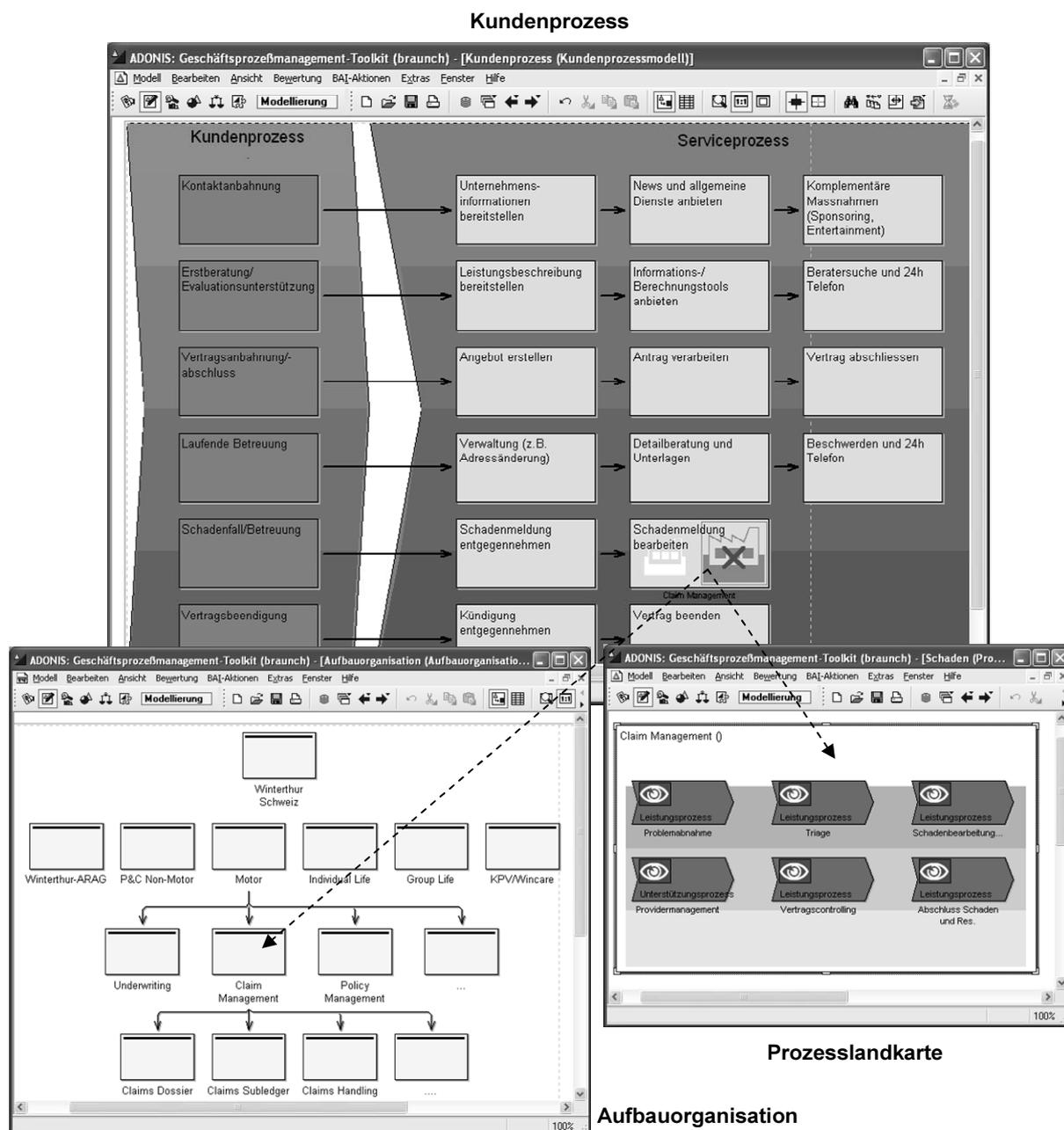


Abbildung 85: Kundenprozessmodell

Prozesslandkarte

Abbildung 86 zeigt die Abbildung der WGR Prozesslandkarte mit Level-1-Geschäftsprozessen in dem entwickelten Prototyp. Die Geschäftsprozesse werden gruppiert nach Steuerungs-, Leistungs- und Unterstützungsprozessen dargestellt.⁶⁵⁸

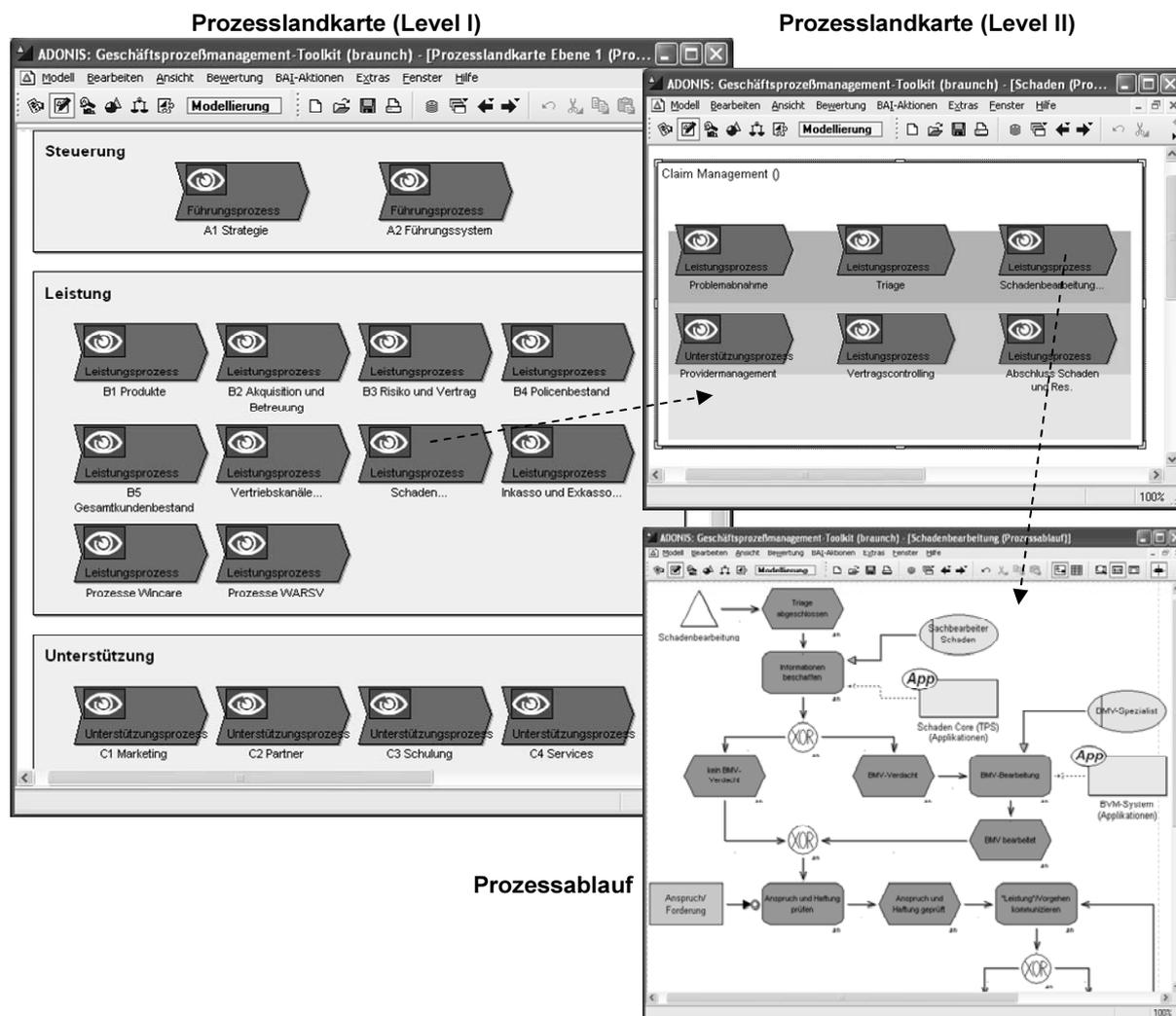


Abbildung 86: Prozesslandkarte der WGR

Zusätzlich sind jedem Geschäftsprozess bestimmte Attribute und Referenzen auf Elemente in anderen Modellen zugeordnet. So können einem Prozess beispielsweise Leistungen im Leistungsmodell zugeordnet werden, die von diesem Prozess erzeugt werden, oder es kann eine Organisationseinheit aus dem Aufbauorganisationsmodell referenziert werden, die für den Prozess verantwortlich ist. Zudem kann ein Prozess auf eine weitere Prozesslandkarte oder

⁶⁵⁸ Seit April 2006 besitzt die Winterthur eine internationalweit gültige „Value Stream Landscape“. Der Treiber dafür war Lean Sigma. Die Landkarte gruppiert die Prozesse neu in "Customer Relationship Processes" und "Business Enablers". Zum Zeitpunkt der Modellierung der Prozesslandkarte mit dem Prototyp war diese Landkarte noch nicht definiert, weshalb noch die Gruppierung nach Steuerungs-, Leistungs- und Unterstützungsprozessen abgebildet ist.

auf einen detaillierten Prozessablauf verweisen. Innerhalb der Prozesslandkarte können zusätzlich Leistungs- und Informationsbeziehungen zwischen den einzelnen Prozessen abgebildet werden.

In dem zuvor geschilderten Beispiel verweist die Serviceaktivität „Schadenmeldung bearbeiten“ auf den Prozess „Schadenbearbeitung“ in der Prozesslandkarte „Schaden“ (Level 2). Durch Anklicken dieses Prozesses wird der detaillierte Prozessablauf dargestellt.

Prozessablauf

Abbildung 87 zeigt die Ablaufplanung der Schadenbearbeitung in vereinfachter Form. Der Zweck dieses Prozesses (festgehalten in der Prozessvision) besteht unter anderem darin, alle für die Bearbeitung notwendigen Informationen inklusive Beweissicherung zu beschaffen, einen möglichen Versicherungsmissbrauch zu erkennen und zu beurteilen, die Entscheidung über die Leistung sowie das Vorgehen an die Schadenbeteiligten zu kommunizieren, Leistungen zu erbringen bzw. ungerechtfertigte Leistungen abzuwehren und den Schadenabschluss mit allen notwendigen Abschlussarbeiten durchzuführen.

Der Prozess beginnt mit dem Abschluss der Schadenfalltriage⁶⁵⁹, welche ein entsprechendes Ereignis auslöst und damit den Prozess der Schadenbearbeitung anstösst. Der für die Schadenbearbeitung verantwortliche Sachbearbeiter muss die benötigten Informationen mit Hilfe der Applikation „Schaden Core“ beschaffen. Dabei handelt es sich um das Schadenverwaltungssystem für den Kernbereich der Schadenerledigung. Liegt der Verdacht eines Versicherungsbetrugs vor, muss der Fall mit dem BMV-Spezialisten der Region besprochen werden. Wird der Verdacht bestätigt, übernimmt der BMV-Spezialist die BMV-Verarbeitung unter Verwendung des BMV-Systems, ein regelbasiertes System zur Bekämpfung von Versicherungsbetrug. Anschliessend folgt die Prüfung des Anspruchs und der Haftung. Diese umfasst sämtliche Aktivitäten der Fallbearbeitung, die zur Vorbereitung des Deckungs-/Haftungsentscheides bzw. des Entscheides, ob eine Leistungspflicht besteht, notwendig sind. Die Entscheide sind zu dokumentieren und stellen somit den Output dieser Aktivität dar. Sind der Anspruch und die Haftung geprüft, wird im nächsten Schritt dem Anspruchsteller und/oder Versicherungsnehmer der getroffene Deckungs-/Haftungsentscheid bzw. die „Leistung“ in geeigneter Form kommuniziert (mündlich/schriftlich) und das weitere Vorgehen vereinbart. Hat dieser den Entscheid akzeptiert, werden die eingehenden Rechnungen durch den Sachbearbeiter auf ihre Vereinbarkeit mit den getroffenen Vereinbarungen geprüft. Entsprechen die Rechnungen den Vereinbarungen, kann die Leistung an die/den Begünstigten entrichtet werden. Besteht keine Regresschance, kann der Schadenfall mit der Durchführung aller notwendigen administrativen Abschlussarbeiten beendet werden.

⁶⁵⁹ Bei der Schadenfalltriage werden die eingehenden Schadenmeldungen nach Hauptbranchen sowie teilweise auch nach Schadenbranchen vorsortiert.

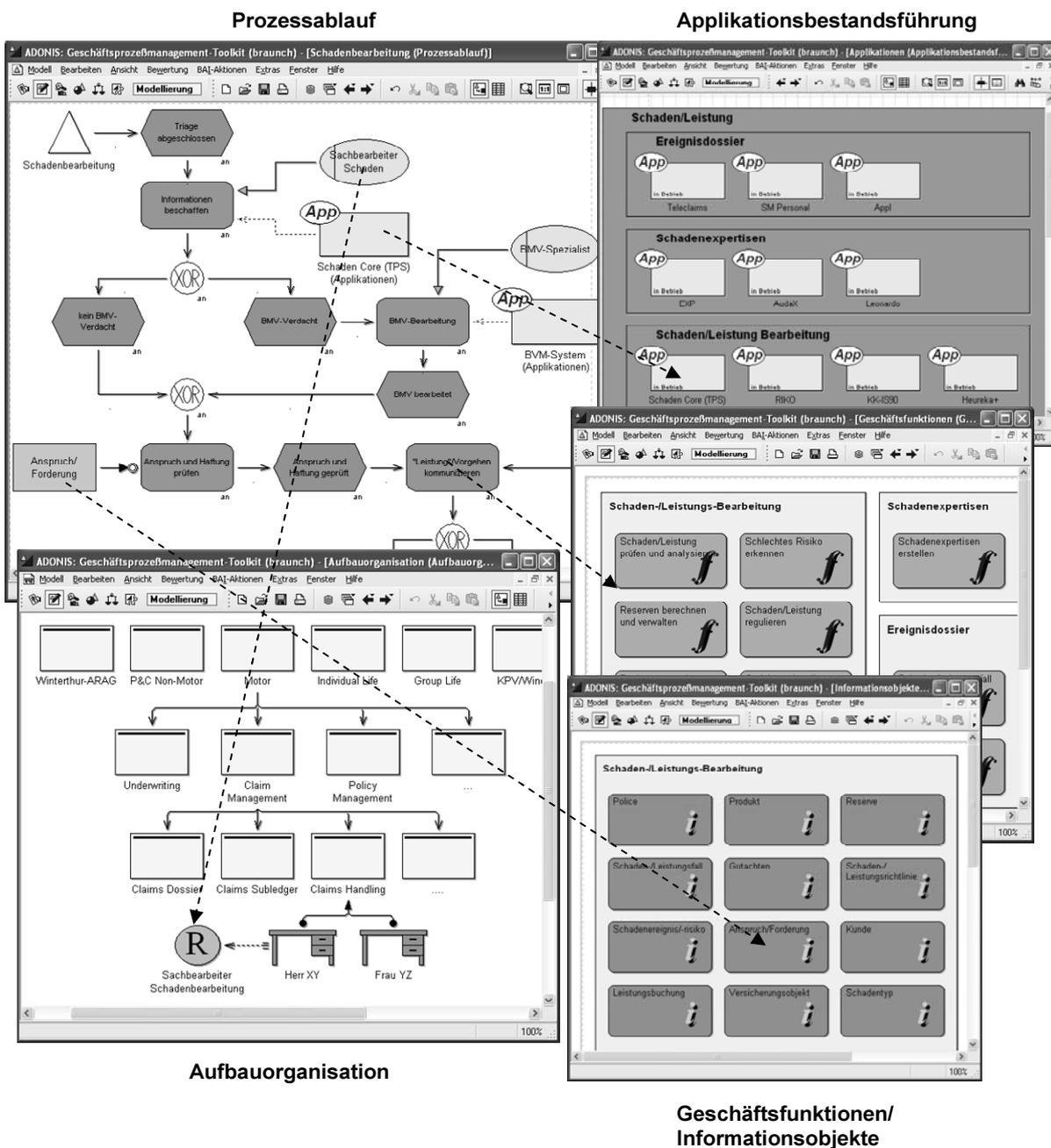


Abbildung 87: Ablaufplanung Schadenbearbeitung

Das Modell der Ablaufplanung der Schadenbearbeitung enthält entsprechend dem zugrunde liegenden Metamodell Referenzen auf Objekte in anderen Modellen. Jeder Aktivität der Ablaufplanung ist über die Prozessrolle ein für die Durchführung verantwortlicher Rollenträger der Aufbauorganisation zugewiesen. So ist beispielsweise bei der Schadenbearbeitung die Prozessrolle „BMV-Spezialist“ für die Durchführung der Aktivität „BMV-Verdacht“ zuständig und in der Aufbauorganisation der Organisationseinheit „Claims Management“ zugeordnet. Die informationstechnische Unterstützung der Durchführung einer Aktivität wird durch eine Referenz auf die entsprechende Applikation in der Applikationsbestandsführung dargestellt. Das Schadenverwaltungssystem „Schaden Core“ wird für die meisten Aktivitäten der

Schadenbearbeitung eingesetzt. In der Applikationsbestandsführung ist sie dementsprechend der Applikationsdomäne „Schaden/Leistung bearbeiten“ zugeordnet. Weitere Referenzen bestehen zum Informationsobjekte- und Geschäftsfunktionenmodell. Die Aktivität „Anspruch und Haftung prüfen“ produziert beispielsweise ein Informationsobjekt, das die Entscheidung dokumentiert. Dieses ist im Informationsobjektmodell abgebildet und wird dort referenziert. Des Weiteren wird eine informationstechnisch unterstützte Aktivität von mehreren Geschäftsfunktionen ausgeführt, die in Form von Applikationsfunktionen realisiert sind, welche wiederum Bestandteil der entsprechenden Applikation sind (vgl. Abbildung 46). Jeder Aktivität der Ablaufplanung können daher mehrere Geschäftsfunktionen zugewiesen werden, die im Geschäftsfunktionenmodell abgebildet sind. So wird die Aktivität „Informationen beschaffen“ unter anderem durch die Geschäftsfunktion „Suche Schadenbericht“ unterstützt, welche durch die Applikationsfunktionen „Einfache Suche“, „Erweiterte Suche“ und „Anzeige der Suchergebnisse“ der Applikation „Schaden Core“ realisiert wird.

Aufbauorganisation

Die folgende Abbildung 88 zeigt einen Ausschnitt der Aufbauorganisation der Winterthur, der die für die Schadenbearbeitung relevanten Organisationseinheiten enthält. Das Modell der Aufbauorganisation wird von allen Ebenen der Unternehmensarchitektur referenziert.

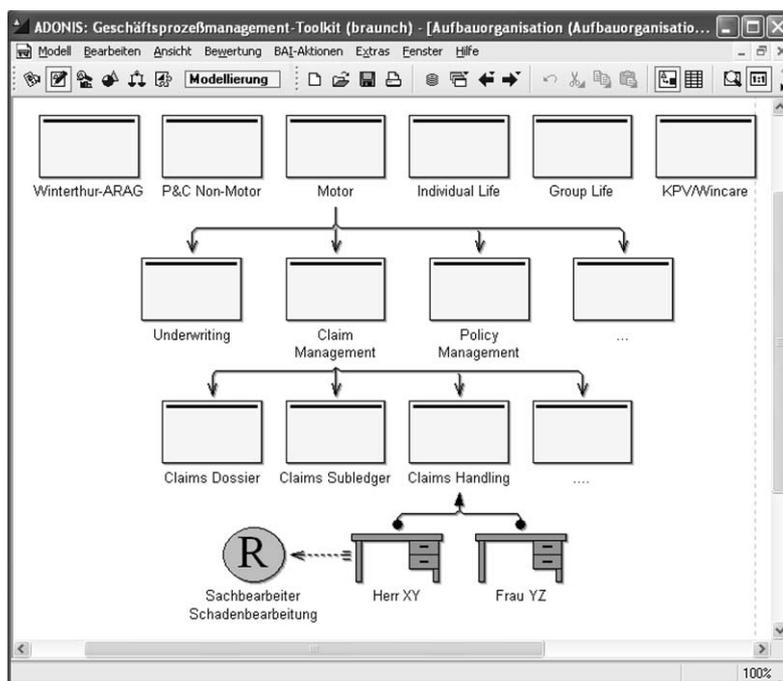


Abbildung 88: Aufbauorganisation

Die Modelle der Strategieebene verweisen lediglich auf das gesamte Unternehmen bzw. auf Geschäftseinheiten (Winterthur Group, Geschäftseinheit „Motor“). Modelle der Organisations- und Informationssystemebene verweisen dagegen auf einzelne Organisationseinheiten, Stellen und Personen. Der Geschäftseinheit „Motor“ ist beispielsweise die Rolle „Exclusive Service Provider“ im Geschäftsnetzwerk zugeordnet. Die Prozessrolle „Sachbearbeiter Scha-

denbearbeitung“ steht in Beziehung mit einzelnen Aktivitäten der Ablaufplanung „Schadenbearbeitung“. Ferner ist auf Applikationsebene die Applikationsdomäne „Schaden/Leistung bearbeiten“ der Organisationseinheit „Claims Management“ zugeordnet.

Geschäftsfunktions- und Informationsobjektmodell

Die Erhebung der Informationsobjekte und Geschäftsfunktionen erfolgte anhand der Dokumentation der Prozessabläufe und der Organisationsstruktur. Abbildung 89 zeigt einen Ausschnitt der für den Prozessablauf der Schadenbearbeitung ermittelten Informationsobjekte und Geschäftsfunktionen.

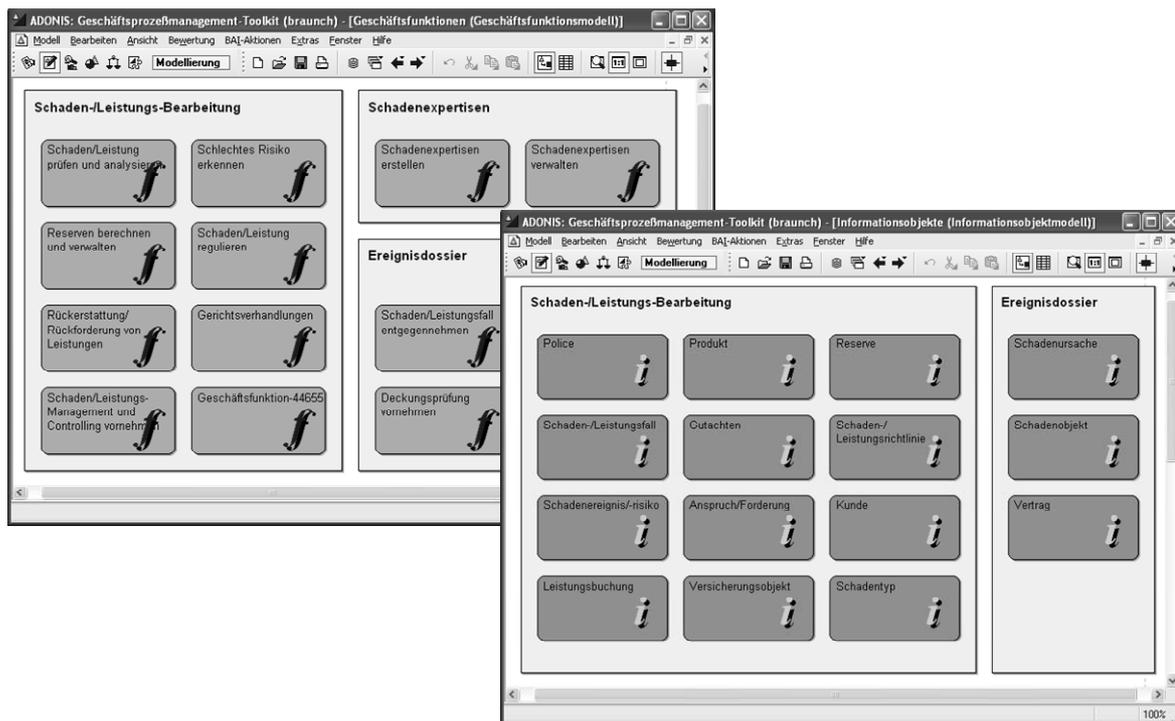


Abbildung 89: Geschäftsfunktions- und Informationsobjektmodell

Analog zu den Applikationen sind auch die Geschäftsfunktionen und Informationsobjekte bestimmten Domänen und Subdomänen zugeordnet. Für jede Geschäftsfunktion und jedes Informationsobjekt sind die Erhebungsquelle sowie Homonyme und Synonyme dokumentiert. Zudem können für jedes Informationsobjekt die entsprechenden Datenobjekte und analog für jede Geschäftsfunktion die entsprechenden Applikationsfunktionen erfasst werden.

Applikationen

Die in Abbildung 90 dargestellten Applikationen und Domänen zeigen einen Teil des Gesamtbestandes aktuell bei der WGR existierender und geplanter Applikationen. Diese sind entsprechenden Domänen und Subdomänen zugeordnet, die bestimmte Geschäftsprozesse unterstützen. Es wird zwischen IST-Applikationen, die im SOLL-Zustand unverändert bleiben, SOLL-Applikationen, die neu zu entwickeln sind, und zu ersetzenden bzw. entsorgenden Applikationen unterschieden. Daneben ist jede Applikation einer der drei Geschäftsbereiche „Non-Life“, „Life&Pensions“ oder „Corporate Center“ zugeordnet. Jede Applikation besitzt

Referenzen auf die von ihr manipulierten Informationsobjekte, auf die von ihr realisierten Geschäftsfunktionen sowie auf die entsprechenden Softwarekomponenten.

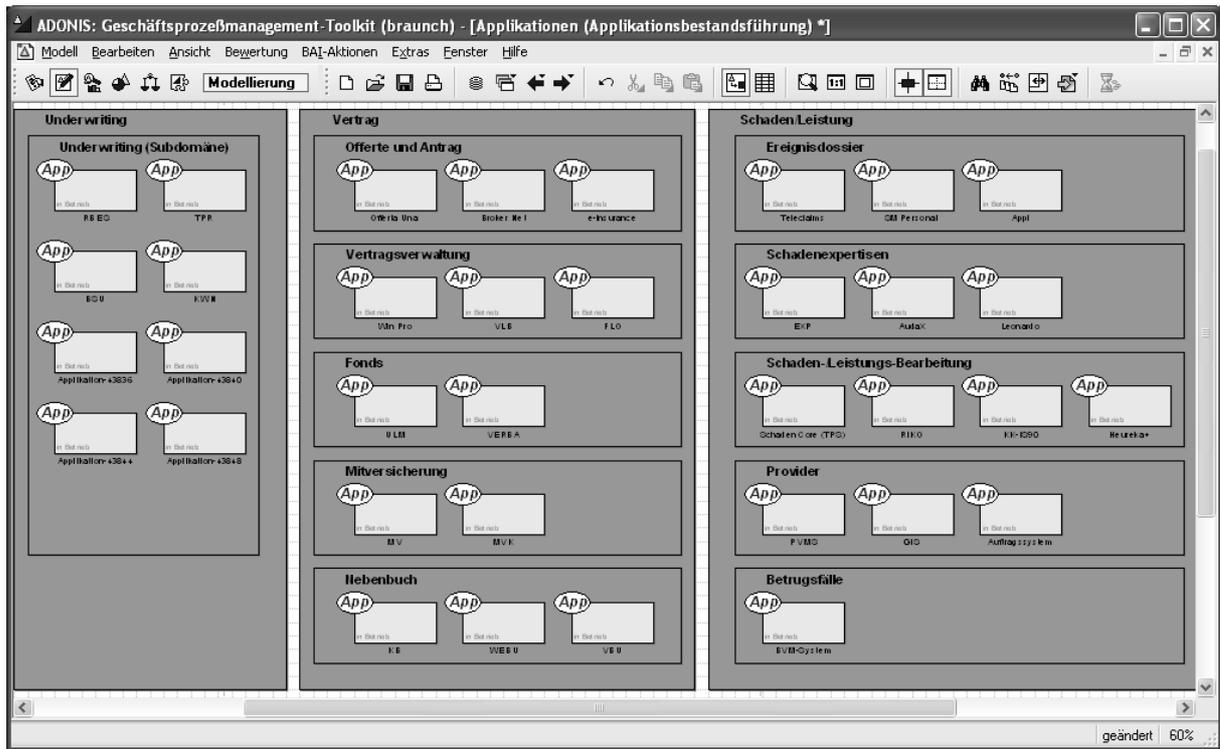


Abbildung 90: Applikationsbestandsführung

Softwarekomponenten und Plattformen

Die folgende Abbildung 91 zeigt exemplarisch einen Teil der Softwarekomponenten der Applikation „Claims Core“, die der Schadenbearbeitung dient.

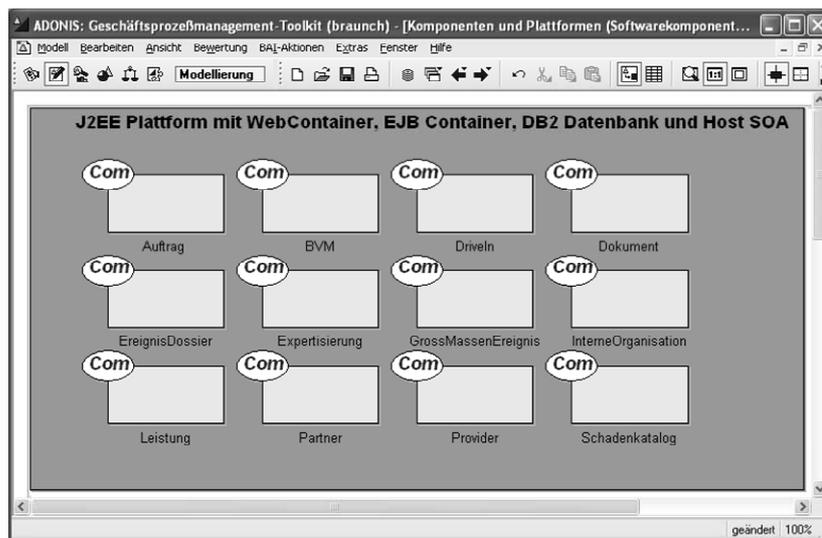


Abbildung 91: Softwarekomponenten

Die Softwarekomponenten laufen auf einer J2EE-Plattform. Als Entwicklungstechnologien kommen J2EE und Eclipse zum Einsatz. Für jede Softwarekomponente können entsprechend

dem Metamodell in Abbildung 49 die verwendete Entwicklungstechnologie, Betriebskomponente und Plattform erfasst werden. Darüber hinaus lassen sich die Schnittstellen und die verantwortliche Organisationseinheit dokumentieren.

Analyse der erstellten Modelle

Die erstellten Modelle der Unternehmensarchitektur gewinnen an Wert, wenn sie nicht nur als gemeinsame Kommunikationsbasis dienen, sondern auch die strategische Entscheidungsfindung durch modellbasierte Analysen unterstützen.⁶⁶⁰ Die modellbasierte Analyse spielt eine zentrale Rolle, wenn es um Veränderungen an Elementen der Unternehmensarchitektur geht und deren Auswirkungen auf andere Bereiche der Unternehmensarchitektur abgeschätzt bzw. antizipiert werden müssen. Aufgrund der Komplexität der Unternehmensstrukturen kann die Analyse der Modelle nicht mehr manuell erfolgen, sondern erfordert eine entsprechende Werkzeugunterstützung in Form bestimmter Abfragen und Mechanismen.

Nach der Informationsaufnahme und der Erstellung der zuvor beschriebenen Modelle konnten, darauf aufbauend, unterschiedliche Analysen definiert und durchgeführt werden. Dadurch war die Beantwortung der in Abschnitt 7.2.3 genannten Fragen möglich. So lässt sich beispielsweise die Frage beantworten, welche Geschäftsprozesse durch welche Applikationsdomänen unterstützt werden, indem die entsprechenden Referenzen ausgewertet und die Objekte in einer Beziehungsmatrix einander gegenübergestellt werden (vgl. Abbildung 92).

	Abschluss ...	Inkasso	Problemabnahme	Providermanagement	Provisionierung	Schadenbearbeitung	Triage	Verkaufsführung	Vertragscontrolling	Vertragsstellung
Betrugsfälle						x				
Ereignisdossier			x			x	x			
Kampagnen								x		
Kundenberatung								x		
Offerte und Antrag										x
Provider				x		x				
Provisionierung		x			x			x		
Schaden/Leistung Bearbeitung			x			x	x			
Schadenexperten						x				
Underwriting (Subdomäne)										x
Vertragsverwaltung	x								x	x
Vertriebsvereinbarung					x					

Abbildung 92: Beziehungstabellen (Applikationsdomäne zu Prozess)

Ebenso lässt sich die Frage beantworten, welche Prozessrollen es gibt und welche Aktivitäten diese durchführen. Ob eine bestimmte Frage beantwortet werden kann, lässt sich sehr einfach nachvollziehen, indem man die entsprechenden im Metamodell definierten Beziehungen betrachtet. Das Werkzeug ermöglicht sowohl die Auswertung von Beziehungen innerhalb eines Modells, als auch solcher mit modellübergreifenden Referenzen. Dadurch können auch Objekte in einer Beziehungsmatrix einander gegenübergestellt werden, deren Metaentitätstypen aus unterschiedlichen Ebenen und Sichten stammen. In dem Beispiel in Abbildung 92 sind

⁶⁶⁰ Vgl. Lankhorst (2005), S. 191.

dies Applikationsdomänen der Applikationsbestandsführung auf Applikationsebene, die Prozessen der Prozesslandkarte auf Organisationsebene gegenübergestellt werden.

Neben den Beziehungstabellen lassen sich auch so genannte Impact-Analysen durchführen. Grundlage dieser Analysen sind die im Metamodell definierten Beziehungen zwischen unterschiedlichen Modellen und Ebenen. Die Ausführung einer Impact-Analyse erfordert das Durchlaufen unterschiedlicher Sichten und Ebenen der Unternehmensarchitektur und die Auswertung der entsprechenden Beziehungen, um festzustellen, ob die Änderung eines Objektes sich über diese Beziehungen auf andere Objekte auswirkt. Dadurch können Auswirkungen von Änderungen an Elementen der Informationssystemebene (z.B. Ausfall eines Infrastrukturelements) auf die Organisations- und Strategieebene vorhergesagt werden (Bottom-Up). Umgekehrt lassen sich aber auch die Auswirkungen von Veränderungen an der Geschäftsstrategie (z.B. Änderung des Kundenprozesses oder Einführung einer neuen Leistung) auf die Organisations- und Informationssystemebene abschätzen.

Das folgende Beispiel illustriert, wie sich eine Änderung an einem Element auf die gesamte Unternehmensarchitektur auswirken kann. Wenn beispielsweise ein Infrastrukturelement ausfällt, sind folglich auch die darauf installierten Plattformen und die auf den Plattformen laufenden Softwarekomponenten nicht verfügbar. Davon sind bestimmte Applikationen betroffen, die mit diesen Softwarekomponenten implementiert sind. Dies hat wiederum Auswirkungen auf die Geschäftsprozesse, die von diesen Applikationen unterstützt werden. Die betroffenen Geschäftsprozesse enthalten Serviceaktivitäten, die bestimmte Kundenaktivitäten und damit Kundenprozesse unterstützen. Darüber hinaus produzieren diese Prozesse Leistungen, die Teil bestimmter Marktleistungen sein können, die an den Kunden gehen. Folgende Beziehungen werden für diese Impact-Analyse ausgewertet (vgl. Abbildung 93):

- Welche Plattformen sind auf dem ausgefallenen Server (Infrastrukturelement) installiert?
- Welche Softwarekomponenten laufen auf diesen Plattformen?
- Welche Applikationen werden mit diesen Softwarekomponenten implementiert?
- Zu welchen Applikationsdomänen gehören diese Applikationen, und welche Geschäftsprozesse werden durch diese Applikationsdomänen unterstützt?
- Welche Serviceaktivitäten werden in diesen Geschäftsprozessen durchgeführt, die wiederum welche Kundenaktivitäten und damit Kundenprozesse unterstützen?
- Welche Prozessleistungen produzieren diese Prozesse, die Teil bestimmter Marktleistungen sind, die an den Kunden bzw. an Partner gehen?

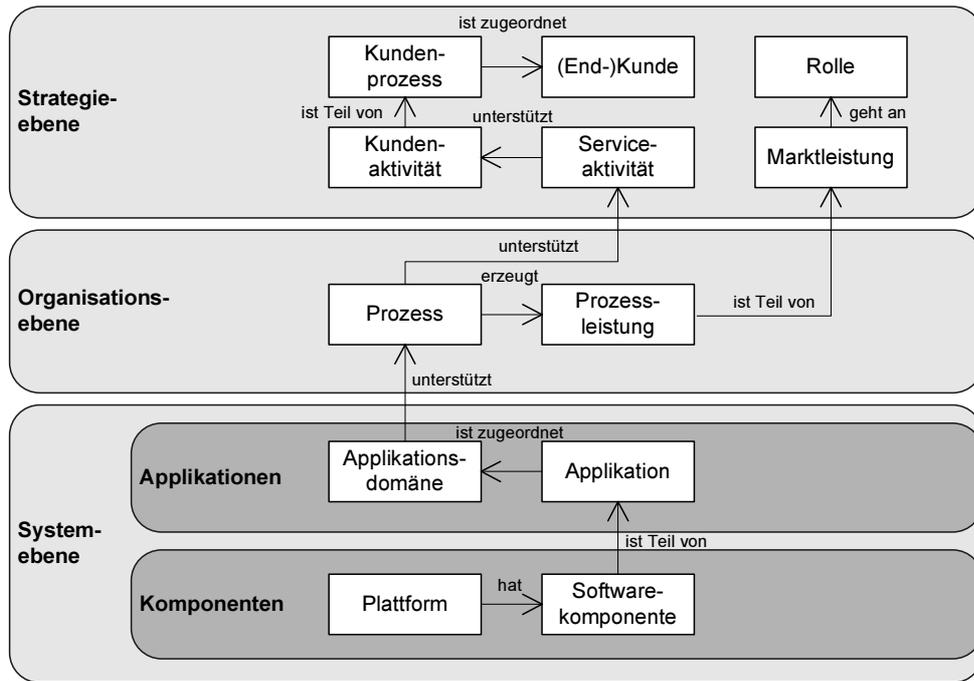


Abbildung 93: Auswertung Metamodellbeziehungen

Die folgende Abbildung 94 zeigt exemplarisch das Ergebnis einer Impact-Analyse. Dieses visualisiert die Auswirkungen des Ausfalls einer bestimmten Plattform auf unterschiedliche Bereiche der Unternehmensarchitektur.

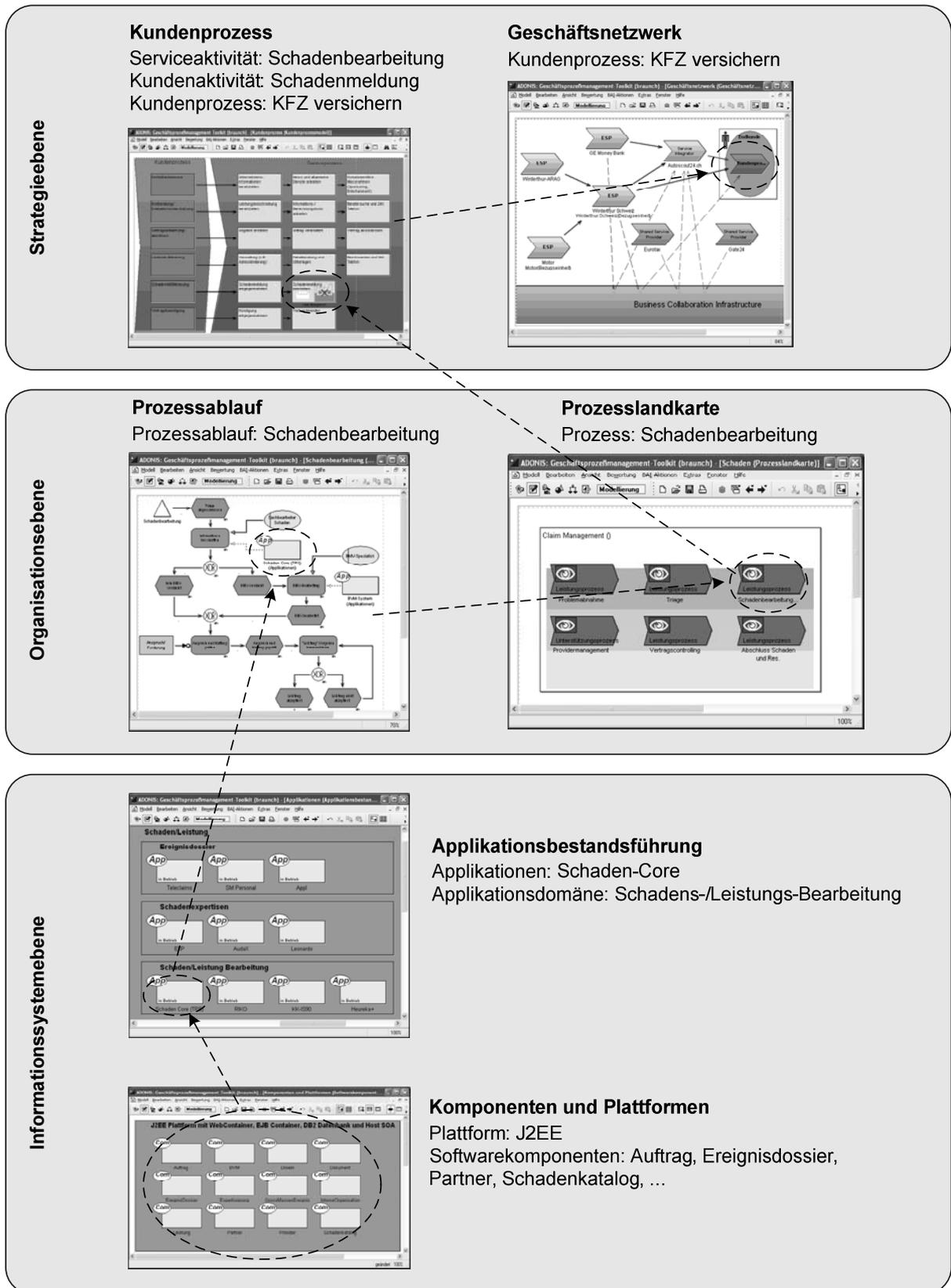


Abbildung 94: Impact-Analyse (Szenario „Plattform fällt aus“)

6.1.7 Fazit

Folgende Erkenntnisse können aus der Fallstudie bei der Winterthur Group abgeleitet werden:

- Die in der Einführung sowie in Abschnitt 2.5 genannten Aspekte stellen ein in der Praxis existierendes und relevantes Problem dar.
- Auch in der Praxis wird ein zentral entwickeltes und gepflegtes Metamodell der Unternehmensstrukturen als Lösungsansatz für diese Problemstellung gesehen.
- Der Ansatz der vorliegenden Arbeit stellt für die Praxis relevante Metaentitäts- und Beziehungstypen auf der Strategie-, Organisations- und Informationssystemebene zur Verfügung.
- Die Modelle der Geschäftsstrategie bilden Sachverhalte in der Realität ab und können mit den Modellen auf Organisationsebene in Beziehung gesetzt werden.

Anhand des Pilot-Projektes im Rahmen der Fallstudie konnten folgende Erkenntnisse gewonnen werden:

- Der entwickelte Ansatz und Prototyp kann mit geringfügigen, einfach durchführbaren unternehmensspezifischen Anpassungen in der Praxis für die Abbildung der Unternehmensstrukturen eingesetzt werden.
- Mit Hilfe des Werkzeuges können unterschiedliche Analysen auf den erstellten Modellen durchgeführt und wichtige Abhängigkeiten erkannt sowie praxisrelevante Fragen beantwortet werden. Damit ist nicht nur eine Kommunikationsbasis geschaffen, sondern die Auswirkungen von bestimmten Änderungen an der Unternehmensstruktur können von den verantwortlichen Personen besser antizipiert werden.
- Das in dieser Arbeit präsentierte Metamodell und der entwickelte Prototyp stellen einen Ansatz dar, der als Ausgangsbasis für andere Projekte in der Praxis verwendet werden kann.

Insgesamt hat die Fallstudie die Anwendbarkeit und den praktischen Nutzen des Ansatzes sowie des Werkzeuges für die Abbildung und Analyse komplexer Unternehmensstrukturen deutlich gemacht.

Darüber hinaus hat die Fallstudie Anregungen und Ideen für mögliche Erweiterungen bzw. die Integration weiterer Konzepte, wie beispielsweise ITIL und SOA, geliefert.

6.2 Metamodellbasierter Vergleich mit bestehenden Ansätzen

Eine metamodellbasierte Evaluierung untersucht und vergleicht die Metaentitätstypen und Beziehungen anhand eines so genannten Master-Metamodells. Eine derartige Analyse erlaubt die Überprüfung der Vollständigkeit und Redundanzfreiheit der verwendeten Konzepte sowie die Identifikation möglicher Spezialisierungen bzw. Generalisierungen.⁶⁶¹

Unter einem Master-Metamodell wird in dieser Arbeit analog zur Definition eines Master-Referenzmodells in Anlehnung an ROSEMANN eine Komposition von verschiedenen Metamodellen verstanden.⁶⁶² Bei dieser Art der Evaluierung wird untersucht, ob das Metamodell des in dieser Arbeit präsentierten Ansatzes die im Master-Metamodell repräsentierten Sachverhalte berücksichtigt. Grundproblem dieser Art der Evaluierung ist, dass zurzeit keine Master-Metamodelle bekannt sind⁶⁶³ und die Definition eines solchen den Rahmen dieser Arbeit sprengen würde. Deshalb wird die Komposition der Metamodelle der in Abschnitt 3.2 erläuterten verwandten Ansätze (ARIS, MEMO und SOM) als Master-Metamodell deklariert und als Grundlage für die Evaluierung verwendet.

Wird ein Metaentitätstyp des Master-Metamodells, also von ARIS, MEMO oder SOM, durch mehrere Metaentitätstypen dieses Ansatzes repräsentiert, so liegt das Problem der (ontologischen) Redundanz vor. Wird dagegen ein Metaentitätstyp des Master-Metamodells durch keinen entsprechenden Metaentitätstyp repräsentiert, so ist der Ansatz unvollständig.

Es ist zu vermuten, dass solche Defizite des zugrunde liegenden Metamodells sich auf die Qualität der Abbildung von Sachverhalten in der Praxis und somit auf die Qualität von Modellen und Referenzmodellen auswirken, die basierend auf diesem Metamodell erstellt wurden. Sie sollten folglich durch eine Gegenüberstellung der Metaentitätstypen bzw. Konzepte identifiziert und behoben werden.

6.2.1 Metamodellbasierter Vergleich mit ARIS

Die von ARIS häufig als Ebenen bezeichneten Phasen der Softwareentwicklung (Fachkonzept, DV-Konzept, Implementierung) lassen sich im Verständnis der vorliegenden Arbeit nicht als Architekturebenen interpretieren, da sie die Sicht auf das Unternehmen zu stark auf für die Softwareentwicklung relevante Sachverhalte eingrenzen. Dennoch können die wesentlichen Metaentitätstypen von ARIS den Architekturebenen des Ansatzes dieser Arbeit zugeordnet werden.

Die folgende Tabelle 27 zeigt eine Gegenüberstellung der wesentlichen Metaentitätstypen des Ansatzes der vorliegenden Arbeit mit den Metaentitätstypen bzw. Konzepten von ARIS. Dar-

⁶⁶¹ Vgl. hierzu *Fettke/Loos* (2004a), S. 10-11.

⁶⁶² Vgl. *Rosemann* (1995), S. 35.

⁶⁶³ Vgl. *Fettke/Loos* (2004a), S. 11.

in ist einerseits ersichtlich, dass das Metamodell dieser Arbeit in Bezug auf die Konzepte von ARIS weitgehend vollständig ist. Andererseits ist zu erkennen, dass zwar zusätzliche Spezialisierungen bzw. Generalisierungen von bestimmten Metaentitätstypen, aber keine redundanten Metaentitätstypen vorhanden sind.

Strategieebene		Organisationsebene	
Ansatz dieser Arbeit	ARIS	Ansatz dieser Arbeit	ARIS
Erfolgsfaktor (BSC)	Kritischer Erfolgsfaktor	Applikation	Anwendungssystem
Unternehmen Geschäftseinheit	Organisationssegment	Ereignis	Ereignis
Geschäftsnetzwerk	Wertschöpfungskette	Aktivität	Funktion manuelle Funktion Systemfunktion
Geschäftsnetzwerk-Rolle	Wertschöpfungsfunktion	Informationsobjekt Datenobjekt	Makrodatenobjekt elektr.-alphanum. Datenobjekt elektronisches Datenobjekt konventionelles Datenobjekt
Kundenprozess Serviceprozess Geschäftsprozess Leistungsprozess Unterstützungsprozess	Funktionssegment Geschäftsprozessesegment Kernprozessvariante Kernprozess Unterstützungsprozess	Operator	Operator
Kundengruppe	Kundengruppe	Organisationseinheit Primäre OE Sekundäre OE	Organisationseinheit Organisationsstruktur Organisationstyp
Marktleistung Leistungsspezifikation	Leistungsfeld	Rollenträger Organisat. Rollenträger Technischer Rollenträger	Primär menschl. Leistungsträger Primär techn. Leistungsträger
Ziel (BSC)	Unternehmensziel	Prozess	Geschäftsprozess
		Prozessleistung Sachleistung/Dienstleistung InfoDL/sonstige DL Kostenart	Leistung (Produkt) Sachleistung/Dienstleistung InfoDL/sonstige DL Kostenart Lieferung
		Prozessziel	Ziel
		Qualifikation	Qualifikation
		Prozessrolle Benutzerkonto	Rolle Benutzerklasse
		Standort	Standort
		Stelle Stellentyp	Stelle

Informationssystemebene	
Ansatz dieser Arbeit	ARIS
Applikation	Anwendungssystem
Informationsobjekt	Makrodatenobjekt
Datenobjekt Klasse	Mikrodatenobjekt Entitytyp
Assoziation	Beziehungstyp
Attribut	Attribut

Tabelle 27: Metamodellbasierter Vergleich mit ARIS

So wird bei ARIS beispielsweise eine Funktion weiter nach manueller Funktion und Systemfunktion differenziert. Zudem unterscheidet ARIS bei Leistungen zwischen Sach- und Dienstleistungen sowie weiter zwischen Informations- und sonstige Dienstleistung. Bei einem Makrodatenobjekt, welches im Ansatz dieser Arbeit mit einem Informationsobjekt korrespondiert, kann es sich um ein elektronisch-alphanumerisches, elektronisches oder konventionelles Datenobjekt handeln. Da es sich bei diesen Metaentitätstypen lediglich um Spezialisierungen handelt, könnten diese bzw. entsprechende Metaentitätstypen durch Vererbung einfach in das Metamodell des in dieser Arbeit präsentierten Ansatzes integriert werden.

Weitere Metaentitätstypen dieses Ansatzes, die in ARIS nicht enthalten sind, wurden aus Gründen der Übersichtlichkeit nicht in die Tabelle aufgenommen.

Anzumerken ist, dass das strategische Geschäftsprozessmodell und das strategische Vorgangskettendiagramm von ARIS eher der Organisationsebene zuzuordnen sind. Das strategi-

sche Geschäftsprozessmodell beschreibt Organisationssegmente und Geschäftsprozesssegmente, die später lediglich detailliert werden. Ebenso bildet das strategische Vorgangskettendiagramm den Prozessablauf in grober, aggregierter Form ab und wird später durch eine EPK detailliert. Im Wertschöpfungskettendiagramm wird der Zusammenhang von Funktionen (primäre Aktivitäten) der Unternehmung zur Leistungserstellung betrachtet. Daraus ergibt sich eine grobe Prozessstruktur, die als Grundlage für die Prozessgestaltung dient. Somit dienen diese Modelle lediglich zur Identifikation und Vorstrukturierung der Geschäftsprozesse. Andere strategische Aspekte können allerdings nicht abgebildet werden.

Die Metaentitätstypen zur Beschreibung der Ablaufplanung von Prozessen stimmen weitgehend überein, da das Metamodell der Ablaufplanung der vorliegenden Arbeit in Teilen auf der EPK von ARIS basiert. Für die Abbildung der Datenstrukturen wird bei ARIS das ER-Diagramm vorgeschlagen. Dessen zugrunde liegende Metaentitätstypen lassen sich in korrespondierende Metaentitätstypen des in dieser Arbeit verwendeten vereinfachten UML-Klassendiagramms überführen.

6.2.2 Metamodellbasierter Vergleich mit MEMO

Das Metamodell von MEMO ist im Vergleich zu dem in dieser Arbeit dargestellten Ansatz sehr abstrakt angesetzt. Daher ist eine Gegenüberstellung der einzelnen Metaentitätstypen der beiden Ansätze nur teilweise möglich. Dennoch finden sich die wesentlichen Konzepte von MEMO auch im Metamodell dieser Arbeit (vgl. Tabelle 28).

In Bezug auf MEMO ist das hier vorgestellte Metamodell somit ebenfalls vollständig. Darüber hinaus konnten auch keine wesentlichen Redundanzen identifiziert werden. Lediglich der Metaentitätstyp „Ressource“, welcher Technologien, Personen und finanzielle Ressourcen beschreibt, ist nicht auf Strategieebene im Ansatz dieser Arbeit vorhanden. Allerdings stellt sich die Frage, ob Ressourcen überhaupt bei der Strategiegestaltung betrachtet werden sollten. Diese sollten eher Teil der nachfolgenden Gestaltung der Organisationsstrukturen und Prozessabläufe sein.

Auf Strategieebene verwendet MEMO analog zu ARIS das Konzept der Wertschöpfungskette, welches eine Alternative zu den Konzepten des Geschäftsnetzwerks und des Kundenprozesses darstellt und eine andere strategische Ausrichtung des Unternehmens definiert.

Im Wesentlichen wird dabei auf den Ansatz von PORTER zurückgegriffen, der im Bereich der strategischen Unternehmensplanung eine erhebliche Bedeutung erlangt hat.⁶⁶⁴ Danach wird ein Unternehmen als ein System von Aktivitätsgruppen dargestellt, die in ihrer Abfolge die Wertkette bilden. Primäraktivitäten sind unmittelbar auf die Erstellung der Güter oder Leistungen gerichtet, die den Kunden des Unternehmens angeboten werden. Eine Aktivität be-

⁶⁶⁴ Vgl. hierzu *Porter* (1999).

schreibt eine wesentliche Funktion, die in einem Unternehmen durchzuführen ist. Sie stellt damit eine Abstraktion konkreter Geschäftsprozesse dar, die zur Erfüllung dieser Funktion benötigt werden.

Strategieebene		Organisationsebene	
Ansatz dieser Arbeit	MEMO	Ansatz dieser Arbeit	MEMO
Erfolgsfaktor (BSC)	Erfolgsfaktor	Ereignis	Ereignistyp
Geschäftseinheit	Geschäftseinheit	Informationsobjekt	Input/Output Spezifikation
Geschäftsnetzwerk	Wertschöpfungskette	Prozessleistung	Input/Output Verwendung
Kennzahl (BSC)	Kennzahl	Organisationseinheit	Organisationseinheit
Kunde	Markt	Prozess	Prozesstyp
Marktleistung	Marktsegment	Führungsprozess	Einfacher Prozesstyp
Kundenprozess	Teilmarkt	Leistungsprozess	Komplexer Prozesstyp
Kundensegment	Marktposition	Unterstützungsprozess	Prozesseinsatz
Geschäftsnetzwerk-Rolle	Wettbewerber	Stelle	Stelle
Serviceaktivität	Aktivität Primäre Aktivität Unterstützungsaktivität	Organisatorischer Rollenträger	Operative Ressource
Serviceprozess	Aktivitätsgruppe	Technischer Rollenträger	
Strategie (BSC)	Strategie		
Unternehmen	Unternehmen		
Ziel (BSC)	Unternehmensziel		
X	Ressource		

Informationssystemebene	
Ansatz dieser Arbeit	MEMO
Klasse	Klasse
Assoziation Aggregation Komposition Interaktion	Beziehung Interaktion Aggregation
Attribut	Attribut Guard Trigger Service

Tabelle 28: Metamodellbasierter Vergleich mit MEMO

Wie bereits zu Beginn von Abschnitt 4.1 ausgeführt, orientieren sich die Modelle zur Strategiegestaltung der vorliegenden Arbeit aufgrund der Positionierung im BE an der Kundenprozessorientierung und der Bildung von Geschäftsnetzwerken. Die lineare vertikale Wertschöpfungskette hat sich in vielen Branchen zu einem verzweigten Wertschöpfungsnetzwerk gewandelt, in dem unterschiedliche Unternehmen und Geschäftseinheiten Teilaufgaben in der Wertschöpfungskette übernehmen. Eine traditionelle Betrachtung und Abbildung der Wertschöpfungskette erscheint somit auch im Rahmen der Unternehmensarchitektur nicht mehr adäquat zu sein.

Auf Informationssystemebene verwendet MEMO eine Modellierungssprache zur Abbildung der Datenstrukturen, die den UML-Klassendiagrammen sehr ähnlich ist. MEMO definiert lediglich für den Metaentitätstyp „Klasse“ zusätzliche Eigenschaften wie „Trigger“ und

„Guard“. Mit Hilfe von Triggers können beispielsweise bestimmten Ereignissen im Lebenszyklus von Objekten Aktionen zugeordnet werden (z.B.: Wenn ein Mitarbeiter 50 Jahre alt wird, ist sein Jahresurlaub um zwei Tage zu erhöhen).⁶⁶⁵ Guards erlauben es, Bedingungen anzugeben, die während der Lebenszeit eines Objekts immer erfüllt sein müssen, mittels der Spezifikation einzelner Attribute allein aber nicht artikuliert werden können (z.B.: Der Verkaufspreis eines Artikels soll niemals geringer sein als der Einkaufspreis). Da allerdings der Ansatz der vorliegenden Arbeit nicht auf softwareentwicklungsrelevante Sachverhalte fokussiert, wird auf eine entsprechende Erweiterung des Metamodells verzichtet.

6.2.3 Metamodellbasierter Vergleich mit SOM

SOM definiert keine Metaentitätstypen für die Strategieebene. Ein Vergleich ist auf dieser Ebene daher nicht möglich.

Die von SOM auf der Organisations- und Informationssystemebene verwendeten Konzepte werden auch im Ansatz der vorliegenden Arbeit weitgehend abgedeckt (vgl. Tabelle 29). Auf Organisationsebene bietet SOM zusätzlich die Metaentitätstypen „Nachricht“ und „Vorbedingung“. Eine Nachricht kann allerdings auch als eine Art Informationsobjekt aufgefasst werden. Analog lässt sich eine Vorbedingung als ein Ereignis interpretieren. Damit würde es sich lediglich um weitere Spezialisierungen bereits im Ansatz dieser Arbeit vorhandener Metaentitätstypen handeln.

SOM differenziert auf Informationssystemebene die konzeptuellen Objekttypen weiter nach objektspezifischen, leistungsspezifischen und transaktionsspezifischen Objekttypen. Die Objekttypen werden entsprechend aus Diskurs-/Umweltobjekten, Leistungsspezifikationen und den Transaktionen des Geschäftsprozessmodells abgeleitet. Analog werden auch im Ansatz dieser Arbeit die Informationsobjekte aus den Modellen der Aufbau- und Ablauforganisation abgeleitet. Durch die Dokumentation der Erhebungsquelle der einzelnen Informationsobjekte werden diese folglich auch entsprechend weiter differenziert.

Da auf Informationssystemebene lediglich automatisierbare Aufgaben bzw. Aktivitäten betrachtet werden, dient die Automatisierbarkeit als ein wichtiges Kriterium zur weiteren Verfeinerung der auf Organisationsebene erstellten Modelle. Ein entsprechendes Konzept sollte deshalb auch im Ansatz dieser Arbeit integriert werden. Dies liesse sich beispielsweise recht einfach durch die Definition zusätzlicher Attribute, wie „Automatisierungsgrad“ und „Automatisierbarkeit“, für den Metaentitätstyp „Aktivität“ umsetzen. Eine Aktivität wird dann analog zu SOM als automatisierbar betrachtet, wenn ein Lösungsverfahren angegeben werden kann, für das ein Informationssystem zur Verfügung steht oder eines konstruierbar ist.

⁶⁶⁵ Vgl. hierzu *Frank* (1995), S. 6.

Organisationsebene		Informationssystemebene	
Ansatz dieser Arbeit	SOM	Ansatz dieser Arbeit	SOM
Aktivität	Aufgabe Entscheidungsaufgabe Lösungsverfahren Transformationsaufgabe	Applikation	Anwendungssystem
Ereignis	Ereignis	Geschäftsfunktion Applikationsfunktion	Automatisierbare Aufgabe Automatisierbare Transaktion Automatisierte Aufgabe Automatisierte Transaktion
Informationsobjekt	Aufgabenobjekt	X	Grad der Automatisierung
Informationsbeziehung Leistungsbeziehung	Betriebliche Transaktion	Organisationseinheit	Betriebliches Objekt
Prozessleistung	Leistungspaket	Informationsobjekt Datenobjekt Klasse	Objekttyp objektspezifisch leistungsspezifisch transaktionsspezifisch aufgaben-spezifisch
Prozessziel	Sach-/Formalziel	Beziehung Aggregation Komposition Generalisierung	Beziehung is-a-Beziehung interacts-with-Beziehung is-part-of-Beziehung
Rollenträger	Aufgabenträgertyp	Attribut	Attribut
Organisationseinheit Prozessrolle Stelle Stellentyp	Betriebliches Objekt		
X	Nachricht		
X	Vorbedingung		

Tabelle 29: Metamodellbasierter Vergleich mit SOM

6.2.4 Fazit

Der in diesem Abschnitt durchgeführte metamodellbasierte Vergleich mit bestehenden Methoden zur Unternehmensmodellierung hat gezeigt, dass deren grundlegende Konzepte auch in dem Ansatz der vorliegenden Arbeit zu finden sind und dieser folglich keine wesentlichen Lücken aufweist bzw. unvollständig ist. Ferner hat der Vergleich gezeigt, dass kein Metaentitätstyp des Master-Metamodells, also von ARIS, MEMO oder SOM, durch mehrere Metaentitätstypen dieses Ansatzes repräsentiert wird. Der entwickelte Ansatz enthält somit auch keine (ontologischen) Redundanzen.

6.3 Natürlichsprachliche und merkmalsbasierte Evaluation

In Abschnitt 2.6 wurden sowohl strukturelle als auch inhaltliche Kriterien zur Beurteilung der verwandten Ansätze definiert. Der in der vorliegenden Arbeit entwickelte Ansatz zur Modellierung der Unternehmensarchitektur wird nun im Folgenden ebenfalls anhand dieser Kriterien beurteilt. Diese Kriterien können zudem im Hinblick auf die im zuvor beschriebenen praktischen Einsatz gewonnenen Erfahrungswerte beurteilt werden.

- **Aggregation/Ganzheitlichkeit:** Um einen möglichst umfassenden Ansatz im Sinne der Unternehmensarchitektur zu entwickeln, wurde auf einer bestehenden Methode (BAI-Methode) aufgebaut und diese um Konzepte aus anderen Methoden zur Unternehmensmodellierung (ARIS, MEMO, SOM) sowie aus domänenspezifischen Ansätzen (Strategiegestaltung, Prozessmanagement, Applikationsintegration, IT-Service-Management, Autorisierung) ergänzt. Dadurch konnten die Schlüsselemente, die für strategische Fragestellungen im Schnittbereich zwischen Fachbereich und IT-Bereich relevant sind, iden-

tifiziert und integriert werden. Der im vorhergehenden Abschnitt durchgeführte metamodellbasierte Vergleich mit existierenden Methoden hat zudem gezeigt, dass deren Konzepte im Wesentlichen alle berücksichtigt wurden.

- **Strukturierung:** Es wurden zahlreiche existierende Ansätze im Hinblick auf die Einteilung in Architekturebenen und Sichten untersucht (vgl. Kapitel 3). Dabei konnten vier wesentliche Architekturebenen sowie 18 relevante Sichten (inklusive der IT-Servicesicht) identifiziert werden, die in dem Ansatz der vorliegenden Arbeit Berücksichtigung gefunden haben (vgl. Abbildung 32).
- **Konsistenz und Formalisierung:** Für die Gewährleistung der Konsistenz wurde einerseits eine klare Einteilung in Ebenen und Sichten (vgl. Abbildung 32) vorgenommen. Andererseits wurden die Schlüsselemente sowie deren ebeneninterne und ebenenübergreifende Abhängigkeiten durch ein zugrunde liegendes Metamodell (semi-)formal spezifiziert, welches mit Hilfe einer Metamodellierungssprache (UML-Klassendiagramme) dargestellt wurde. Auf jeder Ebene konnten zentrale Metaentitätstypen identifiziert werden, die die unterschiedlichen Sichten miteinander verknüpfen. Zudem wurden die Metaentitätstypen und deren Beziehungen verbal beschrieben, um mögliche Missverständnisse zu vermeiden. Die Abbildung in einer Metamodellierungsplattform hat gezeigt, dass das Metamodell ausreichend formal und konsistent sowie nicht zu abstrakt definiert ist. Ausserdem bietet der Ansatz neben dem Metamodell eine systematische Vorgehensweise für die konsistente Entwicklung der auf dem Metamodell basierenden Modelle. Die (semi-)formale Spezifikation der verwendeten Konzepte und deren Beziehungen ermöglicht darüber hinaus die Analyse der Auswirkungen von Veränderungen an bestimmten Elementen der Unternehmensarchitektur.
- **Verständlichkeit:** Die für die Abbildung der unterschiedlichen Ebenen und Sichten verwendeten Symbole und Konzepte stammen aus den jeweiligen Anwendungsdomänen, wie beispielsweise Geschäftsnetzwerk und BSC für die Modellierung der strategischen Ausrichtung, EPK für die Prozessmodellierung, UML für die Datenmodellierung. Sie sollten den Anspruchsgruppen daher ausreichend vertraut sein und ihnen das Verständnis der Modelle erleichtern. Die Modelle sollten beispielsweise sowohl von Managern, die einen groben Überblick benötigen, als auch von Softwareentwicklern, die eine Softwarekomponente implementieren und deren Einsatz bzw. Kontext kennen müssen, interpretiert werden können. Innerhalb einer Anwendungsdomäne können die Modelle und aggregierten Elemente der Unternehmensarchitektur als Ausgangsbasis für detailliertere Spezifikationen dienen. Des Weiteren sind die Metaentitätstypen und deren grafische Repräsentationen getrennt voneinander definiert, so dass gegebenenfalls für den Einsatz in einem spezifischen Unternehmensumfeld andere Symbole verwendet werden könnten.
- **Angemessenheit:** Im Sinne der Unternehmensarchitektur wurden für diesen Ansatz lediglich die Schlüsselemente im Schnittbereich zwischen Fachbereich und IT ausgewählt.

Detailinformationen sollten mit Modellierungssprachen und Werkzeugen der jeweiligen Anwendungsdomäne modelliert werden. In Abschnitt 4.8 wurden entsprechende Schnittstellen zu diesen domänenspezifischen Detailmodellen identifiziert.

- **Flexibilität:** Aus den gesammelten Erfahrungen lässt sich vermuten, dass zwischen dem Kriterium der Flexibilität und der Formalisierung/Konsistenz immer ein gewisser Ausgleich zu finden ist, da diese Kriterien in der Regel nicht zusammen vollständig erfüllt werden können. Ein hoher Formalisierungsgrad bedingt erfahrungsgemäss meist auch eine gewisse Einschränkung bei der Flexibilität. Umgekehrt bedeutet eine hohe Flexibilität meist eine geringere Formalisierung und damit weniger Gewährleistung der Konsistenz. Dennoch ist der vorliegende Ansatz trotz der relativ hohen Formalisierung aufgrund des verwendeten Metamodellierungs- und Vererbungskonzepts bis zu einem gewissen Grad flexibel einsetzbar. Zusätzliche Metaentitätstypen, Beziehungstypen und Attribute können mit geringem Aufwand in den bestehenden Ansatz integriert werden.
- **Methodenelemente:** Im Sinne des Methoden-Engineering (vgl. Abschnitt 2.1) enthält der in der vorliegenden Arbeit entwickelte Ansatz ein Vorgehensmodell (hier aufgrund der Fokussierung auf das Metamodell nur auf Level 1 beschrieben⁶⁶⁶) und Ergebnisdokumente mit einem zugrunde liegenden Metamodell. Bisher wurden allerdings für die Durchführung der Aktivitäten des Vorgehensmodells noch keine entsprechenden Rollen und keine detaillierten Techniken zur Erstellung der Ergebnisdokumente definiert.

Die folgende Tabelle 30 zeigt nochmals zusammenfassend die Erfüllung der inhaltlichen und strukturellen Charakteristika im Vergleich zu den bestehenden Ansätzen. Im Vergleich zu letzteren konnten im Hinblick auf die an einen Ansatz für die Abbildung der Unternehmensarchitektur gestellten Anforderungen wesentliche Verbesserungen erzielt werden. Dabei wurden vor allem die Kriterien der Ganzheitlichkeit, Strukturierung, Formalisierung und Konsistenz adressiert. Dennoch bestehen Verbesserungspotentiale und Erweiterungsmöglichkeiten, z.B. bei der Entwicklung eines Rollenmodells, der detaillierteren Spezifizierung von Techniken zur Erstellung der Ergebnisdokumente und der Integration weiterer Konzepte (z.B. SOA, Sicherheitsarchitektur) im Sinne des Anspruchs der Ganzheitlichkeit an einen Ansatz zur Modellierung der Unternehmensarchitektur.

⁶⁶⁶ Für eine detaillierte Beschreibung des Vorgehens und der einzelnen Techniken zur Abbildung der Unternehmensarchitektur vgl. *Choinowski et al.* (2003) und *Leist* (2004).

Ansatz	Inhaltliche Charakteristika							Strukturelle Charakteristika				
	Ganzheitlichkeit	Strukturierung	Konsistenz	Verständlichkeit	Angemessenheit	Flexibilität	Formalisierung	Vorgehen	Rollen	Ergebnisse	Techniken	Metamodell
ARIS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
MEMO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SOM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FEAF	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TOGAF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zachman	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eigener Ansatz	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				

Legende: erfüllt/vorhanden teilweise erfüllt/vorhanden nicht erfüllt/vorhanden

Tabelle 30: Merkmalbasierte Evaluierung

7 Zusammenfassung und Ausblick

Das letzte Kapitel dient der kritischen Reflexion der erarbeiteten Ergebnisse. Hierzu werden in Abschnitt 7.1 zunächst die wesentlichen Ergebnisse der Arbeit zusammengefasst und diese danach in Abschnitt 7.2 gewürdigt. Abschliessend erfolgt in Abschnitt 7.3 ein Ausblick auf weitere Forschungsthemen und Entwicklungsmöglichkeiten, die in unmittelbarem Zusammenhang mit den Ergebnissen der Arbeit stehen.

7.1 Zusammenfassung der Arbeit

Die Entwicklung einer Unternehmensarchitektur ist eine fundamentale Voraussetzung für die Bereitstellung von Informationssystemen, die mit den organisatorischen und strategischen Richtlinien des Unternehmens konsistent sind. Sowohl in der Praxis als auch in der Wissenschaft fehlt bisher ein allgemein akzeptiertes Verständnis der wesentlichen Ebenen, Elemente und Beziehungen einer Unternehmensarchitektur. Insbesondere existiert für diese Elemente und Beziehungen keine umfassende, (semi-)formale Spezifikation.

Das primäre Gestaltungsziel der vorliegenden Arbeit ist die Entwicklung eines Ansatzes zur Modellierung der Unternehmensarchitektur und dessen Abbildung in einem kommerziellen Metamodellierungswerkzeug. Im Mittelpunkt steht dabei zum einen die Definition eines ebenen- und sichtenübergreifenden Metamodells, das die grundlegenden Elemente eines Unternehmens sowie deren Beziehungen festlegt. Zum anderen wird ein Software-Prototyp mit Hilfe des Metamodellierungswerkzeuges erstellt. Damit sollen die Konsistenz und Umsetzbarkeit des entwickelten Ansatzes gezeigt werden.

Die Erhebung und Analyse aktueller Grundlagen und Ansätze der Unternehmensmodellierung in Praxis und Forschung stellt das primäre Erkenntnisziel der Arbeit dar. Diese bildet zusammen mit einer bereits bestehenden Methode die Grundlage für die Entwicklung des Metamodells. Ein weiteres Erkenntnisziel ist die Analyse kommerziell verfügbarer Modellierungswerkzeuge zur softwaretechnischen Abbildung der Unternehmensarchitektur.

Die Entwicklung des Ansatzes beginnt mit der Aufarbeitung konzeptioneller Grundlagen der Unternehmensmodellierung (Kapitel 2). Zunächst wird das Methoden- und Modellverständnis der vorliegenden Arbeit erläutert. Darauf aufbauend wird der Begriff Unternehmensarchitektur eingeführt und genauer eingegrenzt. Die Architektur eines Unternehmens besteht im Verständnis dieser Arbeit aus verschiedenen Ebenen sowie ebeneninternen und ebenenübergreifenden Sichten auf dieses Unternehmen. Jede Sicht repräsentiert einen Ausschnitt des Unternehmens für einen bestimmten Zweck und wird durch ein zugrunde liegendes Metamodell definiert. Für die Gewährleistung der Konsistenz zwischen den einzelnen Sichten sind die Identifikation der Gesamtzusammenhänge sowie deren komplexitätsreduzierende Darstellung wesentlich. Ein systematischer, methoden- und modellbasierter Ansatz kann hierbei

Unterstützung leisten. Das Business Engineering (BE) bildet aus diesem Grunde den Forschungsrahmen des Dissertationsvorhabens. Im Grundlagenteil werden daher die grundlegende Ausrichtung, die Gestaltungsebenen und Inhalte sowie das vereinfachte Metamodell des Business Engineering vorgestellt. Entsprechend dem BE-Ansatz sind in den letzten Jahren zahlreiche Methoden entwickelt worden. Darunter auch die BAI-Methode, welche den Ausgangspunkt der Arbeit darstellt und daher ebenfalls im Grundlagenteil kurz eingeführt wird. Anschliessend wird auf die Bedeutung der Werkzeugkonstruktion und -unterstützung für die Unternehmensmodellierung eingegangen sowie deren aktueller Entwicklungsstand beschrieben. Nach Einführung der relevanten Begrifflichkeiten und der Eingrenzung des Themengebiets erfolgt eine Diskussion der Ziele und treibenden Kräfte der Unternehmensarchitektur. Darauf aufbauend werden abschliessend im Grundlagenteil die Anforderungen an den zu entwickelnden Ansatz sowie die Werkzeugunterstützung abgeleitet.

Im Anschluss an die Aufarbeitung der Grundlagen erfolgt die Untersuchung methodischer Ansätze, die sich mit dem Thema Unternehmensarchitektur befassen (Kapitel 3). Dabei werden im Hinblick auf die Entwicklung eines eigenen Ansatzes insbesondere solche Ansätze berücksichtigt, die weitestgehend die im Grundlagenteil formulierten Anforderungen erfüllen. Zum einen erfolgt eine detaillierte Beschreibung und Bewertung existierender Methoden zur Unternehmensmodellierung, die primär aus der Wissenschaft stammen. Zum anderen werden aktuelle Bezugsrahmen (Frameworks) vorgestellt und bewertet, die in der Praxis am häufigsten zum Einsatz kommen und die von den meisten aktuell am Markt verfügbaren Werkzeugen unterstützt werden. Aus der Analyse geht hervor, dass keiner der bestehenden Ansätze die Anforderungen vollständig erfüllt. Auffallend ist, dass gerade die in der Praxis weit verbreiteten Bezugsrahmen die Anforderungen am wenigsten erfüllen, insbesondere die Kriterien der Ganzheitlichkeit, Konsistenz und Formalisierung.

Aufgrund dieser Schwächen wird ein eigener Ansatz vorgeschlagen, der die Stärken der bewerteten Ansätze berücksichtigt und vor allem die Kriterien der Ganzheitlichkeit, Konsistenz und Formalisierung adressiert (Kapitel 4). Der Fokus des eigenen Ansatzes liegt deshalb auf der Entwicklung eines möglichst ganzheitlichen, (semi-)formal definierten Metamodells, das insbesondere die ebenen- und sichtenübergreifenden Abhängigkeiten zwischen den grundlegenden Elementen der Unternehmensarchitektur berücksichtigt. Der Ansatz zur Modellierung der Unternehmensarchitektur umfasst die Methodenelemente Vorgehensmodell, Metamodell und Ergebnisdokument auf Strategie-, Organisations-, Applikations-, Softwarekomponenten und Datenstruktur- sowie Infrastrukturebene. Wegen der zunehmenden Bedeutung serviceorientierter Konzepte wird darüber hinaus zum einen ein Metamodell für eine IT-Servicesicht basierend auf der IT Infrastructure Library (ITIL) vorgeschlagen und mit den anderen in dieser Arbeit definierten Sichten auf die Unternehmensarchitektur integriert. Zum anderen wird die Integration einer serviceorientierten Architektur als weitere mögliche Erweiterung der Unternehmensarchitektur betrachtet. Da die detaillierte Abbildung der einzelnen Sichten nicht

im Rahmen der Unternehmensarchitektur, sondern weiterhin in domänenspezifischen Architekturmodellen und mit entsprechenden Modellierungssprachen (und Werkzeugen) erfolgen sollte, werden schliesslich die wesentlichen Schnittstellen zu diesen Detailmodellen identifiziert. Der entwickelte Ansatz bildet somit einen Bezugsrahmen, der Elemente und Konzepte aus domänenspezifischen Ansätzen miteinander in aggregierter Form integriert und Schnittstellen zu entsprechenden Detailmodellen bietet. Die Detailmodelle können somit zunächst nebenläufig gestaltet und weiterentwickelt werden. Die entwickelte Unternehmensarchitektur liefert Vorgaben, aus denen entsprechende Anpassungen resultieren, die die Konsistenz des Gesamtsystems bewahren.

Um die Konsistenz und Umsetzbarkeit des entwickelten Metamodells überprüfen zu können, wird dieses mit Hilfe einer Metamodellierungsplattform abgebildet (Kapitel 5). Dazu werden zunächst ausgewählte, kommerziell verfügbare (Meta-)Modellierungswerkzeuge dargestellt und anhand der im Grundlagenteil abgeleiteten Anforderungen bewertet. Die Entscheidung fiel zu Gunsten von ADONIS der Firma BOC Business Objects Consulting aus. Das Werkzeug deckt bereits ohne anwendungsspezifische Anpassungen alle Hauptkriterien ab. Durch sein methodenunabhängiges Metamodellierungskonzept ermöglicht ADONIS eine flexible und einfache Umsetzung sowie Anpassung der Metamodelle des in dieser Arbeit präsentierten Ansatzes. Der aus der Implementierung mit dem Metamodellierungswerkzeug entstandene Software-Prototyp wird anschliessend im Hinblick auf dessen wesentliche Komponenten und Funktionalitäten vorgestellt. Der Prototyp besteht aus einem Modelleditor für die Erstellung und Bearbeitung von Modellen, einer Analysekomponente für die Durchführung von Abfragen auf den erstellten Modellen, einer Simulationskomponente für die Durchführung von Simulationsalgorithmen sowie einer Import/Export- und Dokumentationskomponente.

Schliesslich wird darüber hinaus eine multiperspektivische Evaluation des Prototyps und des zugrunde liegenden Metamodells in Form einer Fallstudie sowie einer natürlichsprachlichen, merkmalsbasierten und metamodellbasierten Evaluierung durchgeführt (Kapitel 6). Die Fallstudie der Winterthur Group bestätigt die Praxistauglichkeit. Mit geringfügigen unternehmensspezifischen Anpassungen können der Prototyp und das Metamodell in der Praxis für die Abbildung der Unternehmensstrukturen eingesetzt werden. Die natürlichsprachliche und merkmalsbasierte Evaluierung macht deutlich, dass der entwickelte Ansatz im Vergleich zu den bereits bestehenden Ansätzen wesentliche Verbesserungen bringt und die im Grundlagenteil formulierten Anforderungen weitgehend erfüllt. Die metamodellbasierte Evaluierung untersucht und vergleicht die Metaentitätstypen und Beziehungen des entwickelten Metamodells anhand eines so genannten Master-Metamodells. Eine derartige Analyse ermöglicht die Überprüfung der Vollständigkeit und Redundanzfreiheit der verwendeten Konzepte sowie die Identifikation möglicher Spezialisierungen bzw. Generalisierungen. Als Ergebnis dieser Analyse geht hervor, dass das entwickelte Metamodell in Bezug auf das Master-Metamodell vollständig ist und keine Redundanzen aufweist.

7.2 Kritische Würdigung

Im Rahmen der multiperspektivischen Evaluierung wurde bereits eine ausführliche und kritische Diskussion der erarbeiteten Ergebnisse geführt (vgl. Kapitel 6). Da dem Dissertationsvorhaben das Design-Science-Paradigma zugrunde liegt (vgl. Abschnitt 1.3), erfolgt die Würdigung zudem anhand der Design-Science-Richtlinien, die wesentliche Anforderungen an eine Forschungsarbeit formulieren.

Das Ergebnis einer DS-Forschungsarbeit ist ein anwendbares Artefakt in Form einer Methode, eines Modells oder einer Instanz.⁶⁶⁷ Die Ergebnisse der vorliegenden Arbeit sind zum einen der Ansatz und das Metamodell für die Modellierung von Unternehmensarchitekturen und zum anderen der entwickelte (Software-)Prototyp. Da der Ansatz auf einer bestehenden Methode aufbaut, die gemeinsam mit Partnerunternehmen entwickelt wurde, ist das entwickelte Artefakt anwendbar. Die Fallstudie hat darüber hinaus die potenzielle Praxistauglichkeit des Prototyps und des Metamodells demonstriert.

Das Ziel der DS-Forschung ist die Entwicklung technologiebasierter Lösungen für wichtige und relevante Probleme.⁶⁶⁸ Die Relevanz der vorliegenden Arbeit ergibt sich aus den Zielen und treibenden Kräften der Unternehmensarchitektur (vgl. Abschnitt 2.5) sowie den Defiziten bestehender Ansätze (vgl. Kapitel 3). Bisher existiert kein Ansatz, der alle relevanten Bereiche und Elemente der Unternehmensarchitektur konsistent abbildet. Die Arbeit adressiert dieses Problem durch die Entwicklung eines ebenen- und sichtenübergreifenden Metamodells sowie eines geeigneten (Software-)Werkzeuges.

Die Anwendung, Effektivität und Qualität des Artefakts müssen stringent und nachvollziehbar dargelegt werden.⁶⁶⁹ Die Qualität des entwickelten Ansatzes wird durch die Ableitung und Berücksichtigung von Anforderungen sichergestellt (vgl. Abschnitt 2.6), die auf Basis der Grundlagen und Herausforderungen sowie der Ziele und treibenden Kräfte der Unternehmensarchitektur ermittelt werden. Die Konstruktion eines Prototyps zeigt die Konsistenz und ausreichende Formalisierung des Metamodells. Das Fallbeispiel demonstriert dessen Praxistauglichkeit. Darüber hinaus erfolgen eine metamodellbasierte sowie natürlichsprachliche und merkmalsbasierte Evaluation.

Die DS-Forschung muss klar nachvollziehbare und nachprüfbar Ergebnisse in Form von Artefakten oder Beiträgen zur Wissensbasis liefern.⁶⁷⁰ Der Forschungsbeitrag ist sichergestellt, wenn die Defizite bestehender Ansätze und Werkzeuge adressiert werden konnten. Die vorliegende Arbeit entwickelt zum einen ein Artefakt in Form eines Metamodells und eines Prototyps. Darüber hinaus liefert die Arbeit einen detaillierten Vergleich bestehender Ansätze

⁶⁶⁷ Vgl. *Hevner et al.* (2004), S. 82.

⁶⁶⁸ *Hevner et al.* (2004), S. 84.

⁶⁶⁹ *Hevner et al.* (2004), S. 85.

⁶⁷⁰ *Hevner et al.* (2004), S. 87.

und eine Bewertung aktuell verfügbarer Modellierungswerkzeuge. Die multiperspektivische Evaluation zeigt zudem, wie Artefakte der DS-Forschung evaluiert werden können.

Die DS-Forschung verlangt bei der Konstruktion und Bewertung des Artefakts die Einhaltung stringenter, forschungsmethodischer Grundsätze.⁶⁷¹ Die Stringenz der Forschung wird durch die Anwendung der in der Wirtschaftsinformatik allgemein akzeptierten Forschungsmethoden Deduktion, Modellierung sowie Entwicklung und Test eines Prototyps sichergestellt. Dabei wird auf die existierende Wissensbasis zurückgegriffen, insbesondere im Rahmen der Grundlagen und der Diskussion bereits existierender Ansätze. Darüber hinaus wird bei der Entwicklung des Ansatzes auf Literaturquellen zurückgegriffen, die bestimmte Teilbereiche der Unternehmensarchitektur abdecken, wie z.B. Strategiegestaltung, Prozessmanagement, Servicemanagement und Applikationsintegration.

Die DS-Forschung ist ein Suchprozess. Die endgültige Problemlösung wird durch einen iterativen Prozess erreicht.⁶⁷² Die Erkenntnisse aus dem Praxistest und der multiperspektivischen Evaluation sowie das Feedback von Vertretern der Partnerunternehmen im Rahmen von Workshops fließen in einem iterativen Prozess in die Verbesserung des Prototyps und Metamodells ein. Eine optimale Lösung kann in der DS-Forschung nicht bestimmt werden. Durch den iterativen Prozess kann aber zumindest von einer guten Lösung ausgegangen werden.

Die Ergebnisse der DS-Forschung müssen sowohl einem technologieorientierten als auch einem managementorientierten Publikum vermittelt werden.⁶⁷³ Das Metamodell wird detailliert spezifiziert und in einem Prototyp abgebildet, so dass es von Fachkräften in der Praxis eingesetzt werden kann. Die Beschreibung der Ziele und treibenden Kräfte der Unternehmensarchitektur vermitteln auch Führungskräften die Bedeutung der erzielten Ergebnisse. Die Ergebnisse werden zudem in der Forschungsgemeinschaft veröffentlicht und zur Diskussion gestellt.

Neben der Erfüllung der zuvor beschriebenen DS-Richtlinien ist auch das Kriterium der „Allgemeingültigkeit“ kritisch zu diskutieren. Die als Grundlage verwendete BAI-Methode wurde ursprünglich für die Finanzdienstleistungsbranche entwickelt. Es lassen sich deren Konzepte aber auch weitgehend auf andere Branchen übertragen. Zudem wurde sie in dieser Arbeit um Konzepte aus bestehenden (branchenunabhängigen) Ansätzen erweitert. Es wird daher an den in dieser Arbeit entwickelten Ansatz ein branchenübergreifender Allgemeingültigkeitsanspruch gestellt. Da der Allgemeingültigkeitsanspruch einer Methode, eines Referenzmodells oder eines Metamodells nur eingeschränkt erzielt und überprüft werden kann,⁶⁷⁴ muss dieser

⁶⁷¹ Hevner *et al.* (2004), S. 87.

⁶⁷² Hevner *et al.* (2004), S. 88.

⁶⁷³ Hevner *et al.* (2004), S. 90.

⁶⁷⁴ Vgl. hierzu z.B. vom Brocke (2003b), S. 31f. und Wortmann (2005), S. 100.

durch den Einsatz des Prototyps und des Metamodells in zahlreichen weiteren Unternehmen verschiedener Branchen überprüft werden.

Ein weiterer Aspekt, den es zu diskutieren und überprüfen gilt, sind die im Grundlagenteil formulierten Ziele und treibenden Kräfte einer Unternehmensarchitektur. Als eines der wesentlichen Ziele der Unternehmensarchitektur wurde insbesondere das IT-Business-Alignment herausgestellt. Die vorliegende Arbeit stellt nicht den Anspruch, das Problem des IT-Business-Alignment gelöst zu haben. Dennoch können Fragestellungen des IT-Business-Alignment auf Basis des entwickelten Ansatzes besser untersucht und beurteilt werden. Der Ansatz bietet zum einen eine nützliche und wichtige Grundlage für die modellbasierte Analyse von Abhängigkeiten und Wechselwirkungen zwischen der strategischen Ausrichtung, den Prozessen und Organisationsstrukturen sowie den Informationssystemen und der zugrunde liegenden Infrastruktur eines Unternehmens. Zudem bietet er eine gemeinsame Kommunikationsbasis für Mitarbeiter aus unterschiedlichen Unternehmensbereichen.

7.3 Zukünftiger Forschungs- und Entwicklungsbedarf

Die vorliegende Arbeit bildet eine Grundlage für zahlreiche weitere wissenschaftliche Aktivitäten sowie für die Weiterentwicklung des in dieser Arbeit präsentierten Ansatzes und Prototyps. Die folgenden Ansatzpunkte fassen den zukünftigen Forschungs- und Entwicklungsbedarf zusammen:

- **Quantitative Analyse:** Bisher wurde die quantitative Analyse der Unternehmensarchitektur sowohl in der Praxis als auch in der Wissenschaft nicht in Betracht gezogen, da davon ausgegangen wurde, dass diese Detailinformationen erfordert, die bei der Dokumentation der Unternehmensarchitektur nicht oder lediglich in aggregierter Form erfasst werden. LANKHORST ist aber beispielsweise der Meinung, dass auch auf einem eher globalen, aggregierten Niveau die Unternehmensarchitektur quantitativ analysiert werden kann und sollte, insbesondere im Hinblick auf die Abhängigkeiten zwischen den unterschiedlichen Architekturebenen.⁶⁷⁵ So erfordern beispielsweise die Geschäftsprozesse eine bestimmte Performance der unterstützenden Informationssysteme. Umgekehrt beeinflusst die Performance der Informationssysteme die Durchführung der Geschäftsprozesse und damit die Reaktionszeit auf die Kundenprozesse. Analog zur Entwicklung von der Softwarearchitektur hin zur Unternehmensarchitektur könnten folglich auch Konzepte für die Analyse von Software (Software Testing, Simulation) daraufhin untersucht werden, inwieweit sie auf die Analyse der Unternehmensarchitektur übertragen werden können. Dabei spielen auch die Schnittstellen zu den operativen Systemen eine wichtige Rolle. Die quantitative Analyse liesse sich beispielsweise auf Basis bestimmter Kennzahlen durchführen. Diese könnten zunächst operativ gesammelt (z. B. Antwortzeiten der Applikationen, Ausfallzei-

⁶⁷⁵ Vgl. z.B. *Lankhorst* (2005), S. 193-194.

ten, Userbeschwerden, Service-Anfragen etc.) und anschliessend mit Hilfe der durch die Unternehmensarchitektur vorgegebenen Struktur aggregiert und aufbereitet werden. Die Unternehmensarchitektur würde somit als eine Art Analysestruktur dienen.

- **Kostenanalyse:** Aus der Fallstudie mit der Winterthur Group ging unter anderem hervor, dass in der Praxis die Beantwortung der Frage nach den Kosten der IT für einen bestimmten Geschäftsprozess eine grosse Bedeutung hat. Weitere Arbeiten könnten untersuchen, inwieweit sich diese Fragestellung auf Basis des in dieser Arbeit präsentierten Unternehmensarchitektur-Ansatzes beantworten lässt.
- **Evaluierung und Weiterentwicklung:** Wünschenswert wären darüber hinaus weitere Anwendungen des Ansatzes und des entwickelten Prototyps in der Praxis, um diese auf eine breitere Fallstudienbasis zu stellen. Die dabei gesammelten Erfahrungen könnten im Sinne eines iterativen Forschungs- und Entwicklungsprozesses in die Weiterentwicklung des Prototyps und des zugrunde liegenden Metamodells einfließen. Um die in dieser Arbeit ansatzweise durchgeführte multiperspektivische Evaluierung zu ergänzen, könnte zusätzlich eine Evaluierungsmethode der fremddisziplinären Perspektive angewendet werden. FETTKE/LOOS schlagen dafür beispielsweise eine Evaluierung auf Basis der Bunge-Methode vor.⁶⁷⁶
- **Architekturprozesse:** Um das im Grundlagenteil angesprochene IT-Business-Alignment zu erreichen, sind neben einem ebenenübergreifenden Metamodell auch entsprechende Prozesse zur Bewirtschaftung und Kommunikation der erstellten Modelle und ihrer Elemente notwendig. Ziel dieser Prozesse ist es, die auf Basis des Metamodells entwickelten Modelle dauerhaft effizient und effektiv miteinander zu koordinieren. HAFNER hat in seiner Arbeit eine Methode für das Management der IS-Architektur entwickelt.⁶⁷⁷ Ein zukünftiger Forschungsbedarf besteht in der systematischen Überprüfung dieser Methode im Hinblick auf ihre Übertragung auf gesamte Unternehmensarchitekturen und deren Integration mit dem in dieser Arbeit präsentierten Ansatz bzw. dessen Metamodell. Mit zunehmender Reife könnte anschliessend die Werkzeugunterstützung der Architekturprozesse mit dem in dieser Arbeit entwickelten Prototyp betrachtet werden.
- **IT-Servicesicht:** Eine konsistent entwickelte und dokumentierte Unternehmensarchitektur bildet eine wertvolle Grundlage für das IT-Service-Management. IT-Verantwortliche können mit ihrer Hilfe einen klaren Blick auf die Infrastruktur und Applikationen, die unterstützten Geschäftsprozesse sowie auf die unterschiedlichen Abhängigkeiten zwischen diesen Elementen erhalten. Davon profitieren nahezu alle in ITIL definierten Kernprozesse. In Abschnitt 4.4 wurde bereits ein erster Vorschlag zur Integration einer IT-Servicesicht präsentiert, der in nachfolgenden Arbeiten verfeinert werden könnte.

⁶⁷⁶ Vgl. hierzu z.B. *Fettke/Loos* (2003) und *Rosemann/Green* (2002).

⁶⁷⁷ Vgl. *Hafner* (2005).

- **Methodenbestandteile:** Der Fokus der vorliegenden Arbeit lag primär auf der Definition eines ebenenübergreifenden Metamodells. Um diesen Methoden-Ansatz im Sinne des Methoden-Engineering zu einer kompletten Methode auszubauen, sind die Entwicklung eines Rollenmodells sowie die detaillierte Spezifizierung von Techniken erforderlich. Als Grundlage für die Entwicklung des Rollenmodells könnten beispielsweise die in dem Framework der IEEE⁶⁷⁸ definierten Stakeholder und Viewpoints dienen. Darüber hinaus liesse sich die Integration weiterer Konzepte (z.B. SOA, Sicherheitsarchitektur) betrachten, um dem Anspruch der Ganzheitlichkeit noch besser gerecht zu werden.
- **Schnittstellen:** Die Entwicklung und Pflege der Unternehmensarchitektur wird auch in Zukunft nicht durch ein einziges Werkzeug unterstützt werden. Stattdessen werden weiterhin verschiedene, domänenspezifische Modellierungssprachen, -techniken und -werkzeuge in einem Unternehmen zum Einsatz kommen. Spezielle Werkzeuge für die Unternehmensmodellierung, wie der in dieser Arbeit entwickelte Prototyp, bilden lediglich die Beziehungen zwischen den aggregierten Elementen der domänenspezifischen Modelle ab und bieten darauf aufbauend spezifische Analysemechanismen. Die Integration mit domänenspezifischen Werkzeugen ist folglich von entscheidender Bedeutung für den Erfolg und die Akzeptanz des entwickelten Werkzeuges in der Praxis. In Abschnitt 4.8 wurden bereits einige Schnittstellen zu domänenspezifischen Modellierungssprachen und Werkzeugen identifiziert und konzeptionell spezifiziert. Diese könnten durch die Implementierung entsprechender Import/Export-Funktionalitäten und/oder die Definition von Referenzen auf die korrespondierenden Modelle und Objekte in den operativen Systemen technisch umgesetzt werden.

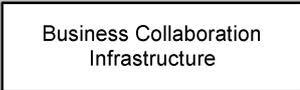
Wie die obigen Ausführungen zeigen, besteht im Bereich der Unternehmensmodellierung und Werkzeugunterstützung noch erheblicher Forschungs- und Entwicklungsbedarf. Die vorliegende Arbeit kann als Grundlage für weitere Forschungsarbeiten genutzt werden. Da die Akzeptanz und Bedeutung der Unternehmensarchitektur in der Praxis weiterhin wächst,⁶⁷⁹ wird sie auch in Zukunft ein relevantes Themengebiet der angewandten Forschung sein.

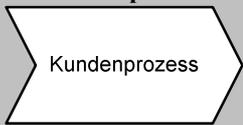
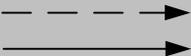
⁶⁷⁸ Vgl. *IEEE* (2000), S. 5 und *Hafner* (2005), S. 29.

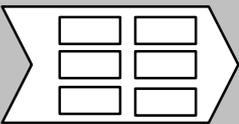
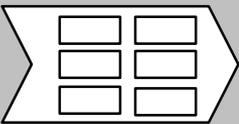
⁶⁷⁹ Vgl. *IFEAD* (2005), S. 33.

Anhang A: Dokumentation der Abbildung der Metamodelle in ADONIS

A.1. Strategieebene

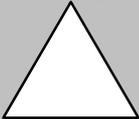
Modelltyp „Geschäftsnetzwerk“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Service Integrator 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
	Leistungsmodell	1:1-Referenz auf Leistungsmodell
Shared Service Provider 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
	Leistungsmodell	1:1-Referenz auf Leistungsmodell
Exclusive Service Provider 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
	Leistungsmodell	1:1-Referenz auf Leistungsmodell
Public Service 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
	Leistungsmodell	1:1-Referenz auf Leistungsmodell
Business Bus (BCI) 	Name	Text
	Beschreibung	Text, mehrzeilig

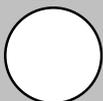
Kundenprozess 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Kundenprozess	1:1-Referenz auf Kundenprozessmodell
Endkunde 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Kontaktpräferenz	Enumeration (Internet, Telefon, Filiale, etc.)
Kundensegment (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
bezieht Leistung (von Rolle zu Rolle) 	Art	Enumeration (standardisiert, individualisiert)
	Farbe	Kontextabhängig
	Linie	Kontextabhängig (direkt, indirekt)

Modelltyp „Kundenprozessmodell“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Kundenprozess (Schwimmbahn) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Kundenbedürfnis	Text
	Kudentyp	Text
	(End-)Kunde	1:1-Referenz auf Kunde in Geschäftsnetzwerkmodell
	Kontaktpräferenz	Enumeration (Internet, Telefon, Filiale, etc.)
Serviceprozess (Schwimmbahn) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Vertriebsweg	Enumeration (Internet, Telefon, Filiale, etc.)
	Reputation	Text
	Problemlösung	Text, mehrzeilig
	Marktleistungen	1:n-Referenz auf eine oder mehrere Ausprägung/en des Leistungsmodells

Serviceaktivität 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
	Referenzierte Prozesse	1:n-Referenz auf Prozesse in Prozesslandkarte
Kundenaktivität 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
bedarf (von Kundenaktivität zu Serviceaktivität) 		

Modelltyp „Leistungsmodell“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Dimension (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
Ausprägung 	Name	Text
	Beschreibung	Text, mehrzeilig

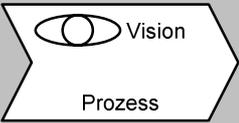
Modelltyp „Zielmodell (BSC)“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Strategie 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
Kritischer Erfolgsfaktor 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Externe Dokumentation	1:1-Referenz auf Word- oder Excel-Dokument

Massnahme (in Poolmodell) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
Ziel 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Massnahmen	1:n-Referenz auf eine oder mehrere Massnahme/n im Massnahmenmodell (Pool)
Kennzahl 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
Perspektive 	Name	Text
	Beschreibung	Text, mehrzeilig
	...	
beeinflusst (von Erfolgsfaktor zu Strategie) 		
hat (von Ziel zu Erfolgsfaktor) 		

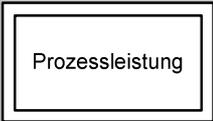
Modelltyp „Massnahmen“ (Pool)		
(Beziehungs-)Klasse	Attribut	Attributtyp
(Prozess-)Leistung 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf OE in Aufbauorganisation
	Ziele	1:n-Referenz auf ein oder mehrere Ziel/e der BSC

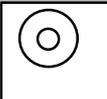
Aggregation 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
ist Vorgänger von (von Massnahme zu Massnahme) 		

A.2. Organisationsebene

Modelltyp „Prozesslandkarte“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Prozess 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Typ	Enumeration (Leistungs-, Unterstützungs-, Führungsprozess)
	(Prozess-)Leistungen	1:n-Referenz auf eine oder mehrere Prozessleistung/en im (Prozess-)Leistungsmodell (Pool)
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation
	Prozessführung	1:1-Referenz auf Modell der Prozessführung
	Subprozess	1:1-Referenz auf Prozess der Prozesslandkarte oder EPK
	Leistungen (Prozessvision)	Text, mehrzeilig
	Ablauf (Prozessvision)	Text, mehrzeilig
	Erfolgsfaktoren (Prozessvision)	Text, mehrzeilig
	IT-Unterstützung (Prozessvision)	Text, mehrzeilig
Bezugseinheit (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Bezugseinheit	1:1-Referenz auf Organisationseinheit in Aufbauorganisation

erbringt/bezieht Leistung (von Prozess zu Prozess) 		
erbringt/bezieht Information (von Prozess zu Prozess) 		

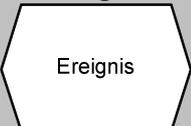
Modelltyp „Leistungen“ (Pool)		
(Beziehungs-)Klasse	Attribut	Attributtyp
(Prozess-)Leistung 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Typ	Enumeration (Sach-, Informationsdienstleistung, sonstige Dienstleistung)
	Leistungsspezifikation	1:n-Referenz auf eine oder mehrere Ausprägung/en im (Markt-)Leistungsmodell auf Strategieebene
	Externe Leistungsspezifikation	1:1-Referenz auf Word- oder Excel-Dokument
	Informationsobjekte	1:n-Referenz auf eines oder mehrere Informationsobjekte
	Kostenart	Enumeration (Kostenarten)
ist übergeordnet (von Leistung zu Leistung) 		

Modelltyp „Prozessführung“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Kritischer Erfolgsfaktor (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Erfolgsfaktor	1:1-Referenz auf Erfolgsfaktor in BSC
Prozessziel 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig

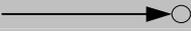
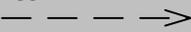
Führungsgrösse 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Kennzahl	1:1-Referenz auf Kennzahl in BSC
	Messsystem	1:1-Referenz auf externe Anwendung
bestimmt (von Prozessziel zu Führungsgrösse)		

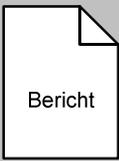
Modelltyp „Aufbauorganisation“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Organisationseinheit 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Standort	Text
	Typ	Enumeration (Unternehmen, externe oder interne Geschäftseinheit)
	Prozesse	1:n-Referenz auf einen oder mehrere Prozess/e der Prozesslandkarte
Stelle 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Stellentyp	Enumeration
	Person	1:1-Referenz auf Person in Aufbauorganisation
	E-Mail	Text
	Telefon	Text
	Standort	Text
	Qualifikationen	Tabelle
Prozessrolle 	Name	Text
	Beschreibung	Text, mehrzeilig
	Standort	Text
	Qualifikationen	Tabelle
	Kompetenzen	Tabelle

Person 	Name	Text
	Qualifikationen	Tabelle
	Stellvertreter	1:n-Referenz auf eine oder mehrere Stelle/n in der Aufbauorganisation
ist übergeordnet (von OE zu OE) 		
gehört zu (von Stelle zu OE) 		
nimmt wahr (von Rollenträger zu Rolle) 		
kommuniziert mit (von OE zu OE) 		

Modelltyp „Ablauforganisation“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Aktivität 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Prozessablauf	1:1-Referenz auf Subprozess
Prozessstart 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Auslöser	Text, mehrzeilig
	Ergebnis	Text, mehrzeilig
Ereignis 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig

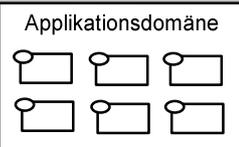
<p>Applikation (Referenz)</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenzierte Applikation	1:1-Referenz auf Applikation in App-Bestandsführung
	Datensicht	1:1-Referenz auf UML-Klassendiagramm
	Steuerungssicht	1:1-Referenz auf EPK
<p>Prozessrolle (Referenz)</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierter Rollenträger	1:1-Referenz auf Rolle, Stelle oder Organisationseinheit in Aufbauorganisation
<p>Operator</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
<p>Prozessleistung (Referenz)</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierte Leistung	1:1-Referenz auf eine Leistung in Leistungen (Pool)
<p>Informationsobjekt (Referenz)</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenziertes Informationsobjekt	1:1-Referenz auf Informationsobjekt in Informationsobjektmodell
<p>Prozesswegweiser</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierter Prozess	1:1-Referenz auf EPK
<p>Kennzahl (Referenz)</p> 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenzierte Kennzahl	1:1-Referenz auf Kennzahl in Prozessführung

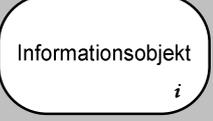
ist Nachfolger/Vorgänger von (von Ereignis zu Aktivität/Operator)	Übergangsbedingung	Text, mehrzeilig
	Übergangswahrscheinlichkeit	Text
	Kommentar	Text, mehrzeilig
führt aus (von Rollenträger zu Aktivität) 		
ist Input für (von IO zu Aktivität) 		
ist Output von (von IO zu Aktivität) 		
unterstützt (von Applikation zu Aktivität) 		
ist zugeordnet (von Führungsgrösse zu Aktivität) 		

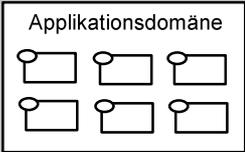
Modelltyp „Informationslandkarte“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Bericht 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Informationsobjekte	1:n-Referenz auf ein oder mehrere Informationsobjekt/e
	Messsystem	1:1-Referenz auf externe Anwendung
Stelle 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierte Stelle	1:1-Referenz auf Stelle in Aufbauorganisation

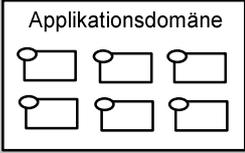
konsumiert/produziert (von Stelle zu Bericht) - - - - ->	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig

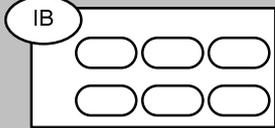
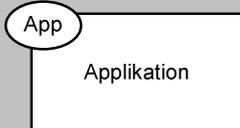
A.3. Applikationsebene

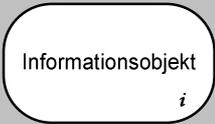
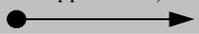
Modelltyp „Applikationsbestandsführung“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Applikation 	Name	Text
	Applikationsbezeichnung	Text
	Applikations-ID	Text
	Informationsobjekte	1:n-Referenz auf ein oder mehrere Informationsobjekt/e im Informationsobjektmodell
	Geschäftsfunktionen	1:n-Referenz auf eine oder mehrere Geschäftsfunktion/en im Geschäftsfunktionenmodell
	Life-Cycle-Status	Enumeration (in Betrieb, in Planung, ...)
	Owner	1:1-Referenz auf OE in Aufbauorganisation
	unterstützt	1:n-Referenz auf einen oder mehrere Prozess/e in Prozesslandkarte
	Anzahl User	Text
	Softwarekomponenten	1:n-Referenz auf eine oder mehrere Softwarekomponent/en in Komponenten und Plattformen
Applikationsdomäne (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Unterstützte Prozesse	1:n-Referenz auf einen oder mehrere Prozess/e in Prozesslandkarte
	Owner	1:1-Referenz auf Organisationseinheit in Aufbauorganisation

Modelltyp „Informationsobjektmodell“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Informationsobjekt 	Name	Text
	Beschreibung	Text, mehrzeilig
	Datenobjekte	1:n-Referenz auf ein oder mehrere Datenobjekt/e
	Begrifflichkeiten	Tabelle
	Erhebungsquellen	1:1-Referenz auf Prozess oder Organisationseinheit

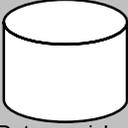
Applikationsdomäne (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Unterstützte Prozesse	1:n-Referenz auf einen oder mehrere Prozess/e in Prozesslandkarte
	Owner	1:1-Referenz auf Organisationseinheit in Aufbauorganisation

Modelltyp „Geschäftsfunktionsmodell“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Geschäftsfunktion 	Organisationseinheit	Referenz auf Org.einheit in Aufbauorganisation
	Name	Text
	Beschreibung	Text, mehrzeilig
Applikationsdomäne (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Unterstützte Prozesse	1:n-Referenz auf einen oder mehrere Prozess/e in Prozesslandkarte
	Owner	1:1-Referenz auf Organisationseinheit in Aufbauorganisation

Modelltyp „Integrationsarchitektur“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Beziehungsstruktur 	Name	Text
	Beschreibung	Text, mehrzeilig
Integrationsbereich 	Name	Text
	Beschreibung	Text, mehrzeilig
	Anwendungsfeld	Text, mehrzeilig
	Anwendungsschnittstellen	Text, mehrzeilig
Applikation (Referenz) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenzierte Applikation	1:1-Referenz auf Applikation in App.-Bestandsführung
	Datensicht	1:1-Referenz auf UML-Klassendiagramm
	Steuerungssicht	1:1-Referenz auf EPK

Informationsobjekt (Referenz) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenziertes IO	1:1-Referenz auf IO in Informationsobjektmodell
Geschäftsfunktion (Referenz) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenzierte Geschäftsfunktion	1:1-Referenz auf Geschäftsfunktion in Geschäftsfunktionenmodell
Leistung/Organisationseinheit/Prozess (Referenz) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenz	1:1-Referenz auf Leistung, Organisationseinheit oder Prozess
wird abgeleitet aus (von Integrationsbereich zu Beziehungsstruktur) 		
ist zugeordnet (von Integrationsbereich zu Applikation) 		
hat Beziehung mit (zw. Informationsobjekt, Geschäftsfunktion oder LOP) 	Art	Enumeration (Leistungs-, Informations-, Daten-, Controllingbeziehung)

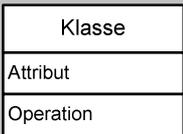
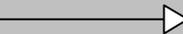
A.4. Softwarekomponenten- und Datenstrukturebene

Modelltyp „Komponenten und Plattformen“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Softwarekomponente 	Name	Text
	Beschreibung	Text, mehrzeilig
	Entwicklungstechnologie	Text, mehrzeilig
	Betriebskomponente	Text, mehrzeilig
	Infrastrukturelement	1:1-Referenz auf Infrastrukturelement
	Applikationen	1:n-Referenz auf eine oder mehrere Applikation/en der Bestandsführung
	Datenobjekte	1:n-Referenz auf eines oder mehrere Datenobjekt/e des Datenobjektmodells
	Owner	1:1-Referenz auf OE in Aufbauorganisation
	Lieferant	Text, mehrzeilig
	Hersteller	Text, mehrzeilig
Plattform (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Infrastrukturelement	1:1-Referenz auf Infrastrukturelement
	Owner	1:1-Referenz auf OE in Aufbauorganisation
Datenspeicher 	Name	Text
	Beschreibung	Text, mehrzeilig
	Tabellen	1:1-Referenz auf externe Anwendung (z.B. Excel)
	Datenobjekte	1:n-Referenz auf ein oder mehrere Datenobjekt/e
	Owner	1:1-Referenz auf OE in Aufbauorganisation
Schnittstelle 	Name	Text
	Beschreibung	Text, mehrzeilig
	Datenobjekte	1:n-Referenz auf ein oder mehrere Datenobjekt/e
	Owner	1:1-Referenz auf OE in Aufbauorganisation
Betriebskomponente (Aggregation) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Owner	1:1-Referenz auf OE in Aufbauorganisation

Entwicklungstechnologie  Entwicklungs- technologie	Name	Text
	Beschreibung	Text, mehrzeilig
	Owner	1:1-Referenz auf OE in Aufbauorganisation
Kennzahl (Referenz) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierte Führungsgrößen	1:n-Referenz auf eine oder mehrere Führungsgrösse/n der Prozessführung
Datenfluss (zw. SW-Komponente/ Datenspeicher und Schnittstelle) 		
wird erstellt mit (zw. SW-Komponente und Entwicklungstechnologie) 		

Modelltyp „Datenobjekte“		
(Beziehungs-)Klasse	Attribut	Attributtyp
Datenobjekt 	Name	Text
	Beschreibung	Text, mehrzeilig
	Datenspeicher	1:1-Referenz auf Datenspeicher in Komponenten und Plattformen
	Softwarekomponenten	1:n-Referenz auf eine oder mehrere SW-Komponent/en in Komponenten und Plattformen
	Detaillierung	1:1-Referenz auf Datenstrukturmodell (UML-Klassendiagramm)
Datenspeicher (Referenz) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Referenzierter Datenspeicher	1:1-Referenz auf Datenspeicher in Komponenten und Plattformen

ist zugeordnet (von Datenobjekt zu Datenobjekt) 		

Modelltyp „Datenstrukturmodell“ (UML-Klassendiagramm)		
(Beziehungs-)Klasse	Attribut	Attributtyp
Klasse 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Attribute	Text, mehrzeilig
	Operationen	Text, mehrzeilig
	Source Code (Doku)	1:1-Referenz auf externe Anwendung
Aggregation (von Klasse zu Klasse) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kardinalität	Text
Komposition 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kardinalität	Text
Generalisierung (von Klasse zu Klasse) 	Name	Text
	Beschreibung	Text, mehrzeilig
Assoziation (von Klasse zu Klasse) 	Name	Text
	Beschreibung	Text, mehrzeilig
	Richtung	Enumeration
	Kardinalität	Text

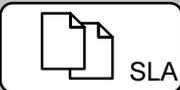
A.5. Infrastrukturebene

Modelltyp „Infrastrukturmodell“ (Produktionsarchitektur)		
(Beziehungs-)Klasse	Attribut	Attributtyp
Infrastrukturelement 	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Typ	Enumeration (Desktop, Server, Peripherie, ...)
	Softwarekomponenten	1:n-Referenz auf eine oder mehrere SW-Komponent/en
	Plattformen	1:n-Referenz auf eine oder mehrere Plattform/en

	Systemeigenschaften	Tabelle
	Hardwarekomponenten	Tabelle
	Owner	1:1-Referenz auf OE in Aufbauorganisation
	Standort	Text
Kommunikationselement  Kommunikations- element	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Typ	Enumeration (Switch, Router, Multiplexer, ...)
	Systemeigenschaften	Tabelle
	Standort	Text
ist verbunden mit (von I/K-Element zu I/K- Element)	Name	Text
	Beschreibung	Text, mehrzeilig
	Kardinalität	Text

A.6. IT-Services

Modelltyp „IT-Servicemodell“		
(Beziehungs-)Klasse	Attribut	Attributtyp
IT-Service (Aggregation)  IT-Service	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Service Level Requirements	Tabelle
	Infrastrukturelemente	1:n-Referenz auf ein oder mehrere Infrastrukturelement/e
	Softwarekomponenten	1:n-Referenz auf eine oder mehrere SW-Komponent/en
	IT-Service-Prozesse	1:n-Referenz auf einen oder mehrere IT-Service-Prozess/e in (IT-)Prozesslandkarte
	Dienstleister	Tabelle
	Owner	1:1-Referenz auf OE in Aufbauorganisation
	Servicekatalog	1:1-Referenz auf externe Anwendung
	(Prozess-)Leistung	1:1-Referenz auf Leistung in Leistungen (Pool) auf Organisationsebene
Kommunikationselement (Referenz)  Kommunika- tionselement	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenziertes Kommunikationselement	1:1-Referenz auf K-Element in Infrastrukturmodell

Infrastrukturelement (Referenz)  Infrastrukturelement	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenziertes Infrastrukturelement	1:1-Referenz auf I-Element in Infrastrukturmodell
Softwarekomponente  Komponente	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierte Komponente	1:1-Referenz auf SW-Komponente in Komponenten und Plattformen
IT-Service-Prozess (Referenz)  IT-Service-Prozess	Name	Text
	Beschreibung	Text, mehrzeilig
	Kommentar	Text, mehrzeilig
	Referenzierter Prozess	1:1-Referenz auf Prozess in Prozesslandkarte auf Organisationsebene
	Input	Text, mehrzeilig
	Output	Text, mehrzeilig
	Owner	1:1-Referenz auf OE in Aufbauorganisation
	IT-Dienstleister	Text
 SLA	Name	Text
	Beschreibung	Text, mehrzeilig
	Service Levels	Tabelle (Service-Zeiten, Verfügbarkeit, Performance, Support-Zeiten, ...)
definiert (von SLA zu IT-Service) 	Name	Text
	Beschreibung	Text, mehrzeilig
erzeugt/verwendet (von IT-Service-Prozess zu IT-Service) 	Name	Text
	Beschreibung	Text, mehrzeilig

Anhang B: Darstellung einer Auswahl kommerziell verfügbarer Modellierungswerkzeuge

Die Darstellung des jeweiligen Werkzeugs beinhaltet eine tabellarische Kurzcharakteristik des Anbieters sowie eine Beschreibung des Funktionsumfangs und der Systemarchitektur. Die Untersuchung basiert auf Anbieterhomepages sowie Produktbroschüren, Whitepaper, Präsentationen oder Evaluierungsversionen der Lösungen, die im Internet frei zugänglich sind oder auf Anfrage zur Verfügung gestellt wurden. Darüber hinaus wurden von Gartner veröffentlichte Berichte für die Auswahl und Bewertung der Werkzeuge herangezogen. Die Analyse wurde im Mai 2005 abgeschlossen.

Um die einzelnen Werkzeuge einheitlich und konsistent beschreiben zu können, wird ein qualitativer Bezugsrahmen verwendet. Die Darstellung der Werkzeuge erfolgt anhand der folgenden Charakteristika:⁶⁸⁰

- **Anbieter:** Charakterisiert den Anbieter des Werkzeugs anhand von Eckdaten und beschreibt dessen Position im Markt.
- **Unterstützte Methoden und Bezugsrahmen (Frameworks):** Ein wichtiges Kriterium für die Beurteilung eines Werkzeuges stellen die unterstützten (Modellierungs-)Methoden (z.B. UML 2.0 oder BPMN 1.0) und/oder Bezugsrahmen dar. Diese legen fest, welche Bereiche der Unternehmensarchitektur mit Hilfe des Werkzeuges abgebildet und welche Analysen und Operationen auf den erstellten Modellen durchgeführt werden können.
- **Modellierungsoberfläche:** Die Modellierungsoberfläche dient dem Entwurf, der Entwicklung, der Pflege sowie der Verarbeitung von Modellen der Unternehmensarchitektur. In der Regel werden die Modelle in graphischer Form repräsentiert. Zusätzliche Informationen zu den Modellen können meist in textueller Form den graphischen Repräsentationen beigelegt werden. Die Qualität der Modellierungsoberfläche (z.B. Benutzerfreundlichkeit, Strukturierung, Übersichtlichkeit) ist ein wichtiges Kriterium für die Beurteilung eines Werkzeuges.
- **Erweiterbarkeit und Anpassbarkeit:** Dieses Kriterium beurteilt, inwieweit das Werkzeug an die individuellen Anforderungen eines bestimmten Unternehmens angepasst werden kann. Ein Werkzeug kann beispielsweise an ein unternehmensindividuelles Umfeld angepasst werden, indem neue Methoden/Bezugsrahmen hinzugefügt oder bereits unterstützte Methoden/Bezugsrahmen modifiziert werden. Einige Werkzeuge verfügen zudem über eine vereinfachte Programmieroberfläche, um die Funktionen und Mechanismen des

⁶⁸⁰ Vgl. *Schekkerman* (2004), S. 202-206.

Werkzeugs zu erweitern. Einige wenige ermöglichen darüber hinaus die Anpassung oder Erweiterung der zugrunde liegenden Metamodelle.

- **Repository und Systemarchitektur:** Die meisten auf dem Markt verfügbaren Werkzeuge verwenden ein Repository in Form einer relationalen Datenbank zur Speicherung der entworfenen Modelle. Die Funktionen des Repositorys haben einen grossen Einfluss auf die gesamte Funktionalität des Werkzeugs. Die Systemarchitektur eines Werkzeugs beschreibt dessen softwaretechnische Struktur und Implementierung. In der Regel kann zwischen einer „single user/single client“ oder einer „simple two-tier client/server“ Struktur unterschieden werden.

Casewise Corporate Modeler

Anbieter

Das Unternehmen Casewise wurde 1989 gegründet und hat ihren Firmensitz in Mount Laurel (USA) sowie in London (UK). Ihr Werkzeug Corporate Modeler⁶⁸¹ stammt ursprünglich aus dem Bereich der Geschäftsprozessmodellierung und -analyse. Neuere Versionen des Tools erfassen nicht mehr nur die Geschäftsprozesse, sondern auch andere Sachverhalte, wie Personen, welche die Geschäftsprozesse durchführen, Standorte, an denen die Prozesse auftreten, sowie die verwendeten Applikationen, Daten und die eingesetzte Hardware. Laut Gartner liegt die Stärke des Werkzeugs in seinen graphischen Fähigkeiten und seinen Möglichkeiten, die IT-Architektur eines Unternehmens mit Artefakten der Geschäftsarchitektur zu verknüpfen. Es bietet die Möglichkeit, Daten mit einer Reihe unterschiedlicher führender Design-Werkzeuge auszutauschen. Um seine Position am Markt verbessern zu können, müsse Casewise vor allem die Administrationsfunktionalitäten sowie die Metamodellunterstützung erweitern.⁶⁸²

Unterstützte Methoden/Bezugsrahmen

Der Corporate Modeler verwendet standardmässig ein proprietäres Framework, das auf dem Zachman Framework basiert. Es bildet einen Ausgangspunkt für die Modellierung, indem es dem Benutzer Vorlagen und erprobte Beispiele zur Verfügung stellt.

Modellierungsoberfläche

Der Corporate Modeler bietet eine weitgehend intuitive, individuell anpassbare Modellierungsoberfläche. Die Objekte werden gut dargestellt und können beliebig in Grösse und Position verändert und ausgerichtet werden. Semi-transparente Objekte ermöglichen die Modellierung mehrerer Ebenen in Prozess- und Beziehungslandkarten. Eine Baumstruktur ermöglicht die einfache Navigation entsprechend den unterschiedlichen Detaillierungsebenen. Ein Statusfenster macht den Benutzer auf mögliche Konsistenzprobleme während der Modellierung aufmerksam.

⁶⁸¹ Vgl. im Folgenden *Casewise* (2005b), *Casewise* (2005a) und *Casewise* (2005c).

⁶⁸² Vgl. *James* (2005b).

Erweiterbarkeit und Anpassbarkeit

Die Modellierungsobjekte, deren Eigenschaften sowie deren Beziehungen untereinander können individuell angepasst werden. Dadurch ist das Repository leicht erweiterbar und es kann ein angepasster Bezugsrahmen als Grundlage für die Modellierung mit dem Werkzeug verwendet werden.

Repository und Systemarchitektur

Das Repository des Corporate Modeler wird durch eine skalierbare relationale Datenbank realisiert. Diese speichert Diagrammobjekte mit zugeordneten Informationen sowie Modelle. Einmal erstellte Diagrammobjekte (Ressourcen, Prozesse, Daten-Entitäten etc.) werden automatisch im Repository gespeichert und können in jedem beliebigen Diagramm wieder verwendet werden. Dadurch lassen sich Redundanzen vermeiden. In einer Multi-User-Umgebung können die Benutzer auf die Objekte gemeinsam zugreifen. Das Repository ermöglicht das kollaborative Arbeiten mehrerer Benutzer an zentral gespeicherten Modellen von unterschiedlichen Standorten aus. Offline-Arbeiten an Modellen können mit dem Repository synchronisiert werden. Ein Zugriffs- und Rechtesystem sowie eine Reihe von Sicherheitsoptionen verhindern den unbefugten Zugriff auf die in der zentralen Datenbank gespeicherten Modelle.

ARIS Design Platform

Anbieter

Die IDS Scheer AG wurde 1984 gegründet und hat ihren Hauptsitz in Saarbrücken. Weltweit beschäftigt das Unternehmen 2200 Mitarbeitende. IDS Scheer ist ein Marktführer im Bereich der Geschäftsprozessmodellierung und -analyse. Das von IDS Scheer angebotene Werkzeug ARIS Process Platform⁶⁸³ ist sehr mächtig und bietet umfangreiche Funktionalitäten. Dadurch ist es in der Anwendung sehr komplex. Aufgrund seiner Herkunft ist ARIS vor allem für die Unternehmensmodellierung geeignet, bei der die Geschäftsprozesse im Mittelpunkt stehen. Um seine Marktführerschaft im Bereich Unternehmensmodellierung ausbauen zu können, müsse IDS Scheer laut Gartner die Komplexität seines Werkzeuges für weniger erfahrene Unternehmen reduzieren, um diesen einen leichteren Einstieg zu ermöglichen.⁶⁸⁴

Unterstützte Ansätze

Die ARIS Design Platform bietet Unterstützung für das Zachman Framework sowie für die Bezugsrahmen FEAF, TEAF und TOGAF (vgl. Abschnitt 3.3). Des Weiteren können verschiedene Bezugsrahmen integriert verwendet werden.

Modellierungsoberfläche

Der ARIS Designer dient der Modellierung von Geschäftsprozessen. Das Werkzeug bietet zahlreiche Navigationsmöglichkeiten, wodurch komplexe Sachverhalte in übersichtliche Teilbereiche gegliedert und grosse Modelle vollständig überblickt werden können. Die Organisa-

⁶⁸³ Vgl. Im Folgenden *IDS-Scheer (2005a)*, *IDS-Scheer (2005c)* und *IDS-Scheer (2005b)*.

⁶⁸⁴ Vgl. *James (2005b)*.

tions-, Daten- und Funktionssicht ermöglichen unterschiedliche Blickwinkel auf die Geschäftsprozesse. Der Modellierer erhält zudem Unterstützung durch Modellierungsfunktionen, wie beispielsweise Mehrfachplatzierung von Objekten, benutzerdefinierte Objektdarstellung, automatische Layoutgenerierung, mehrstufige Undo-Funktion, etc. Der Explorer bietet Funktionen zum einfachen Erzeugen, Umbenennen, Verschieben und Löschen von Gruppen, Modellen und Objekten.

Erweiterbarkeit und Anpassbarkeit

ARIS kann bis zu einem gewissen Grad individuell angepasst werden. Komplexere Anpassungen müssen aber in der Regel durch Serviceleistungen von IDS Scheer oder dessen Partnern vorgenommen werden. Unternehmen können Modelltypen, Objekttypen, Symbole, Attributtypen und Attributtypengruppen umbenennen. Für Objekttypen können eigene Symbole angelegt werden. Ausserdem können bestimmte Attribute individuell konfiguriert und beliebigen Attributtypgruppen zugeordnet werden. Konfigurationsfilter dienen der Anpassung der ARIS-Methode an die Anforderungen bestimmter Benutzer.

Repository und Systemarchitektur

Der ARIS Web Designer ermöglicht den standort- und plattformunabhängigen Einsatz des Designers. Die Drei-Ebenen-Architektur besteht aus einem browserfähigen Front-End, einem Applikations- und einem Datenbankserver. Dadurch können die Unternehmensabläufe verteilt und unter Zugriff auf ein zentrales Repository gestaltet werden. In einer Multi-User-Umgebung können die Benutzer auf die Objekte gemeinsam zugreifen und diese bearbeiten.

Alfabet SITM

Anbieter

Die Alfabet Meta-Modeling AG mit Sitz in Berlin wurde im Jahre 1995 gegründet. Sie entwickelt und vertreibt Software-Lösungen und Services für das strategische Informationsmanagement über komplexe Geschäftsarchitekturen. Im August 2004 wurde Alfabet von Forrester als Anbieter von Change-Management-Lösungen der dritten Generation identifiziert. Das von der Firma Alfabet vertriebene Produkt SITM (Strategic IT Management)⁶⁸⁵ ist ein strategisches Planungs-Werkzeug für das IT-Management, mit welchem sich die IT-Architektur einer Unternehmung detailliert beschreiben lässt.

Unterstützte Ansätze

Alfabet verwendet in seiner Standardlösung das proprietäre Framework SITM (Strategic IT Management). Dieses unterscheidet eine Geschäftsebene (Business Layer), Applikationsebene (Application Layer) sowie eine physische Ebene (Physical Layer).

⁶⁸⁵ Vgl. im Folgenden *Alfabet* (2005b) und *Alfabet* (2005a).

Modellierungsoberfläche

Die Informationen im Repository können über ein Web-Interface visualisiert und bearbeitet werden. Unterschiedliche Baumstrukturen ermöglichen die datenzentrierte, hierarchische Navigation innerhalb des Repositorys. Rollenspezifische Darstellungen erleichtern den Benutzern die Suche nach den für sie wichtigen Informationen. Zusätzlich werden einfache und erweiterte Funktionen für die Suche nach Objekten angeboten. Für die einzelnen Objekte des Repositorys werden automatisch benutzerdefinierte Berichte generiert.

Erweiterbarkeit und Anpassbarkeit

Grundsätzlich können die wesentlichen Elemente eines Unternehmens modelliert und inventarisiert werden. Das zugrunde liegende Datenmodell bietet zudem eine gewisse eingeschränkte Flexibilität, wenn zusätzliche Attribute oder Elemente hinzugefügt werden sollen. Eine spezielle Anpassung an das unternehmensindividuelle Umfeld muss in der Regel in Zusammenarbeit mit Alfabet erfolgen.

Repository und Systemarchitektur

Das Repository (Logical IT Inventory) stellt die zentrale Komponente von Alfabets Lösung dar. Es erfasst die aktuelle IT-Landschaft des Unternehmens und liefert Informationen, auf deren Grundlage zukünftige IT-Projekte und Planungsaufgaben durchgeführt werden können. In dem Repository können Informationen über Organisationsstrukturen, Prozesse, Applikationen, Infrastrukturkomponenten sowie Standorte einheitlich und konsistent inventarisiert werden. Darüber hinaus ermöglicht es die Abbildung logischer Zusammenhänge und Abhängigkeiten zwischen den IT-relevanten Objekten der Geschäftsarchitektur. Das Repository erlaubt den Zugriff von mehreren Personen über das Internet. Änderungen können nur von Benutzern mit entsprechenden Schreib- und Leserechten durchgeführt werden. Die Rechte können für jedes Objekt separat vergeben werden.

Alfabet bietet, aufbauend auf dem Repository, weitere Module an, wie z.B. die Module „Application Architecture Management“, „Business Demand Management“ und „Enterprise Architecture Management“. Diese leisten Unterstützung bei der Planung von Projekten, indem zukünftige Änderungen in der Applikations-, Technologie- und Prozesslandschaft mit Projekten assoziiert werden können. Dadurch können Strategien entwickelt und deren Umsetzung vorangetrieben und überwacht werden.

Das Werkzeug verwendet für die Umsetzung des Repository eine relationale Datenbank (MS SQL-Server oder Oracle). Die Bewirtschaftung der Daten und Informationen kann über ein Web-Interface oder einen Windows-Client erfolgen. Es bietet eine Rechte- und Zugriffsverwaltung, durch die die Applikations- und Technologieverantwortlichen die Möglichkeit haben, die Daten dezentral zu pflegen und auf einem aktuellen Stand zu halten.

Adaptive Enterprise Architecture Manager

Anbieter

Adaptive ist ein vor allem in Nordamerika und Europa tätiger Software-Hersteller, dessen Produkte das IT-Management bei der Verwaltung der architektur-relevanten Objekte und der Planung und Koordination von Veränderungen unterstützen. Die von Adaptive angebotene Produktpalette besteht aus den drei Modulen „Enterprise Architecture Manager“, „Business Process Manager“ und „IT Portfolio Manager“. ⁶⁸⁶ Diese können entweder einzeln oder zusammen in einem Unternehmen eingesetzt werden.

Des Weiteren können Informationen mit einer Reihe von Modellierungswerkzeugen ausgetauscht werden. Das Werkzeug bietet allerdings nur einfache grafische Darstellungsmöglichkeiten für Diagramme, z.B. basierend auf Microsoft Visio. ⁶⁸⁷

Unterstützte Ansätze

Adaptive war an der Spezifikation des Federal Enterprise Architecture Framework (FEAF) (vgl. Abschnitt 3.3.2) aktiv beteiligt. Folglich unterstützt auch der Enterprise Architecture Manager dieses Framework. Das Referenzmodell von Adaptive, welches die Struktur der Informationen über eine Organisation beschreibt, gilt damit als Industriestandard für die Modellierung von Organisationsstrukturen bei US-amerikanischen Regierungsbehörden. FEAF kann aber auch für die Abbildung anderer komplexer Organisationen verwendet werden.

Modellierungsoberfläche

Da es sich bei dem Produkt um ein reines Repository zur Speicherung von Unternehmensinformationen handelt, besitzt es keine eigene Modellierungsoberfläche. Stattdessen verwendet es aber aktuelle Modellierungsstandards, um mit zahlreichen anderen Modellierungswerkzeugen interagieren zu können, wie z.B. Microsoft Visio, IBM Rational Rose, Popkin System Architect oder Sybase Power Designer. Daten können somit importiert, transformiert und integriert sowie letztendlich auch wieder exportiert werden.

Erweiterbarkeit und Anpassbarkeit

Der Adaptive Designer bietet eine Reihe von Werkzeugen, die Adaptive für die Entwicklung und Anpassung ihrer Produkte verwendet. Mit einer Lizenz für den Designer können aber auch Unternehmen selbst die Produkte von Adaptive auf ihre individuellen Bedürfnisse anpassen. Mit Hilfe des Designers ist es möglich, Adaptive anzupassen und zu erweitern, indem Klassen, Attribute und Beziehungen im Metamodell hinzugefügt oder geändert werden. Dadurch können weitere, für das Unternehmen wichtige Informationen in dem Repository gespeichert werden. Zusätzlich können neue oder alternative Sichten erstellt werden, um festzulegen, wie Informationen über das Unternehmen dargestellt werden sollen und wie zwischen diesen navigiert werden kann.

⁶⁸⁶ Vgl. im Folgenden *Adaptive* (2005b), *Adaptive* (2005a), *Adaptive* (2005c) und *Adaptive* (2005d).

⁶⁸⁷ Vgl. *James* (2005b).

Repository und Systemarchitektur

Die Stärke des Enterprise Architecture Manager ist das Repository, welches ursprünglich von Unisys entwickelt wurde und von vielen Anwendern verwendet wird. Das Repository bietet zahlreiche Administrationsfunktionen, wie beispielsweise die Möglichkeit, Diskussionsbeiträge und Definitionen mit den Diagrammen abzuspeichern. Es unterstützt aktuelle Standards, wie beispielsweise UML oder CWM. Diese können auch erweitert oder miteinander integriert werden. Daneben ermöglicht es den Austausch von Modellen mit anderen Werkzeugen, die den XML-basierten XMI-Standard unterstützen. Für die Speicherung der Daten verwendet das Repository gängige relationale Datenbanken. Bei unterschiedlichen Versionen eines Objektes müssen lediglich die Änderungen gespeichert werden, wodurch sich die Zugriffszeit auf die Datenbank verringert.

EAM ist ein rein web-basiertes Tool, welches lediglich einen Browser voraussetzt. Der Benutzer kann über den Web-Browser alle wichtigen Funktionen nutzen, wie z.B. Modelle anzeigen und ändern. Die Modelle werden im Repository gespeichert. Mit Hilfe des Designers können Sichten auf die Metamodelle verändert oder neu definiert werden.

Popkin Enterprise Architect

Anbieter

Die Firma Popkin wurde 1988 gegründet und hat ihren Firmensitz in New York.⁶⁸⁸ Popkin bietet für die Unternehmensmodellierung das Werkzeug System Architect an.⁶⁸⁹ Das Werkzeug bietet eine grosse Palette an unterschiedlichen Diagrammtypen und ist damit vor allem für Anwender interessant, die Wert auf gute Visualisierungsfähigkeiten legen. Diese werden ergänzt durch umfangreiche, aber damit auch komplexe Funktionalitäten für die Erstellung von Berichten und Präsentationen. System Architect bietet Im- und Exportfunktionen für zahlreiche Design-Tools und kann Informationen in unterschiedlichen Formaten austauschen, wie z.B. XML und CSV. Gartner prognostiziert, dass Popkin mit seinem Produkt mittelfristig in einer Marktführerposition bleiben wird.⁶⁹⁰

Unterstützte Ansätze

Der Enterprise Architect von Popkin unterstützt bereits vordefinierte und von der Industrie akzeptierte Frameworks, wie beispielsweise das Zachman Framework oder The Open Group Architecture Framework (TOGAF). Unternehmen können aber auch ein auf ihre Anforderungen angepasstes Framework verwenden. Das Werkzeug bietet zudem Unterstützung für die objekt- und komponentenbasierte Modellierung mit UML sowie für die relationale Datenmodellierung mit Entity-Relationship-Diagrammen.

⁶⁸⁸ Die Firma Popkin wurde am 18. April 2005 durch die Firma Telelogic aufgekauft.

⁶⁸⁹ Vgl. im Folgenden *Popkin* (2003) und *Popkin* (2004).

⁶⁹⁰ Vgl. *James* (2005b).

Modellierungsoberfläche

Die graphische Oberfläche bietet viele Möglichkeiten für das Zeichnen und Anzeigen von Diagrammen, welche die Geschäftsarchitektur, die Applikationen oder das Datenbankdesign repräsentieren. Der Browser des System Architect zeigt dem Benutzer die gespeicherten Diagramme und Definitionen in einer Baumstruktur an. Diese können von dort aus direkt geöffnet und editiert werden. Filterfunktionen erleichtern die Suche nach bestimmten Diagrammen und Definitionen. Ein weiteres Hilfsmittel zur einfachen Navigation bietet der Framework-Browser. Dieser ermöglicht die Visualisierung der erstellten und im Repository gespeicherten Modelle und Artefakte entsprechend dem verwendeten Framework. Beim Anklicken einer bestimmten Zelle des Frameworks wird eine Liste mit Diagrammen angezeigt, die dieser Zelle zugeordnet sind. Jeder Diagrammtyp besitzt eine eigene Toolbox, in der die methodenspezifischen Symbole angezeigt werden.

Erweiterbarkeit und Anpassbarkeit

Das zugrunde liegende Metamodell des System Architect kann für die Abbildung unternehmensspezifischer Informationen angepasst werden. Das Metamodell kann um neue Diagrammtypen, Symbole, Definitionen sowie neue Attribute erweitert werden. Zusätzlich ist es möglich, die Darstellung der in den Diagrammen abgebildeten Informationen anzupassen.

Repository und Systemarchitektur

Alle Modelle, Diagramme und Definitionen, die mit dem System Architect erzeugt werden, werden in einem zentralen, mehrbenutzerfähigen Repository, genannt Encyclopedia, abgelegt. Das Repository basiert auf einer relationalen Datenbank (MS SQL Server). Der System Architect wird in der Regel in einem Multiuser-Umfeld eingesetzt, kann aber auch als Stand-alone-Installation auf einem Client verwendet werden.

BOC ADONIS

Anbieter

Die BOC Information Technologies Consulting GmbH wurde 1995 in Wien als Spin-off der BPMS- (Business Process Management Systems-) Gruppe der Abteilung Knowledge Engineering der Universität Wien gegründet. Ausgehend vom Stammsitz in Wien entstanden weitere Landesgesellschaften in Berlin, Madrid, Dublin, Athen und Warschau. Die BOC ist ein Beratungs- und Softwarehaus, welches sich auf Strategie-, Geschäftsprozess- und IT-Management spezialisiert hat. Derzeit beschäftigt die BOC über 100 Mitarbeiter. Das Geschäftsprozessmanagement-Werkzeug ADONIS unterstützt die Modellierung, Analyse, Simulation und Evaluation von Geschäftsprozessen.

Unterstützte Ansätze

ADONIS stellt standardmässig die ADONIS-Standard-Methode zur Verfügung. Diese ist eine universelle Modellierungstechnik zur Abbildung von Abläufen und Organisationsstrukturen. Die Methode ist branchenunabhängig für die Modellierung, Analyse, Dokumentation und

Optimierung einsetzbar. Die zur Verfügung stehenden Modelltypen ermöglichen eine integrierte und konsistente Darstellung von strategischen bis hin zu umsetzungsrelevanten Informationen. ADONIS bietet eine Vielzahl weiterer fertig ausgearbeiteter Methoden, wie z.B. LOVEM, EPK, UML 2.0 und BPMN.

Modellierungsoberfläche

Die in ADONIS erstellten Modelle und Objekte besitzen Attribute, in denen qualitative und quantitative Informationen wie Beschreibungen, Kommentare, Zeiten, Kosten, Referenzen auf andere Modelle oder externe Dokumente gespeichert werden. Der Modelleditor ist intuitiv zu bedienen und bietet umfangreiche Funktionalitäten. Er bietet beispielsweise die Möglichkeit schnell und einfach zwischen den einzelnen Modellen zu navigieren, externe Grafiken einzufügen, unterschiedliche Ansichtsmodi auszuwählen und Grafiken in einer Vielzahl von Dateiformaten zu exportieren. Der Explorer ist ein Sichtfenster, in dem die ADONIS-Modelle und deren Statusinformation in einer Baumstruktur angezeigt werden. Er ermöglicht den Zugriff auf einzelne Funktionen und Sichten.

Erweiterbarkeit und Anpassbarkeit

ADONIS basiert auf einem methodenunabhängigen Metamodellierungsansatz. Dieser ermöglicht eine leichte Anpassung ohne Programmieraufwand sowohl im Bereich Modellierungstechniken als auch bei den Modell auswertenden Funktionen. Aufgrund der Offenheit und Methodenunabhängigkeit lassen sich unterschiedliche Aspekte eines Unternehmens abbilden, wie z.B. Geschäftsprozesse, Organisationsstrukturen, IT-Systeme und Qualitätssicherungsmaßnahmen.

Repository und Systemarchitektur

Für das Repository kann ein gängiges relationales Datenbanksystem verwendet werden. ADONIS kann entweder als Stand-alone-Installation oder in einer Client/Server-Architektur verwendet werden.

Anhang C: Ansprechpartner bei der Winterthur Group

Die Fallstudie wurde auf Basis von Interviews und Dokumentenanalysen zusammengestellt. Vertiefende Nachbesprechungen der Interviews wurden telefonisch oder per E-Mail durchgeführt. Folgende Ansprechpartner waren an der Erhebung der Fallstudie beteiligt:

Dario Bee

MBA, Head Business Analysis Claims

Winterthur Versicherungen

General Guisan-Str. 40

CH-8401 Winterthur

dario.bee@winterthur.ch

www.winterthur.ch

Urs Bernet

IT-Architekt

Winterthur Versicherungen

General Guisan-Str. 40

CH-8401 Winterthur

urs.a.bernet@winterthur.ch

www.winterthur.ch

Fiorenzo Maletta

IT Architect, Head Architecture Coaching & Information Architecture

Winterthur Versicherungen

General Guisan-Str. 40

CH-8401 Winterthur

fiorenzo.maletta@winterthur.ch

www.winterthur.ch

Literatur

Adams (2005)

Adams, P.: Poll Shows Integration Increasing Among IT Asset Management Tools, <http://www.gartner.com> (Zugriff: 02.05.2006).

Adaptive (2005a)

Adaptive: Business Process Manager, <http://www.adaptive.com/links/papers.html> (Zugriff: 04.05.2005).

Adaptive (2005b)

Adaptive: Enterprise Architecture Manager, <http://www.adaptive.com/links/papers.html> (Zugriff: 04.05.2005).

Adaptive (2005c)

Adaptive: IT Portfolio Manager, <http://www.adaptive.com/links/papers.html> (Zugriff: 04.05.2005).

Adaptive (2005d)

Adaptive: White Paper – The Road to Enterprise Architecture, <http://www.adaptive.com/links/papers.html> (Zugriff: 04.05.2005).

Alfabet (2005a)

Alfabet: White Paper – Alfabet`s Logical IT Inventory, http://www.alfabet.de/alfabet_e/whitepapers.html (Zugriff: 04.05.2005).

Alfabet (2005b)

Alfabet: White Paper – Planning IT from demand to budget, http://www.alfabet.de/alfabet_e/whitepapers.html (Zugriff: 04.05.2005).

Alpar et al. (2005)

Alpar, P., Grob, H.-L., Weimann, P., Winter, R.: Anwendungsorientierte Wirtschaftsinformatik, 4. Aufl., Vieweg, Braunschweig/Wiesbaden, 2005.

Anaya/Ortiz (2005)

Anaya, V., Ortiz, A.: How enterprise architectures can support integration, in: Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems (IHIS), Oldenburg, 2005, S. 25-30.

Arsanjani (2004)

Arsanjani, A.: Service-oriented modeling and architecture – How to identify, specify, and realize services for your SOA, <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/> (Zugriff: 21.04.2006).

Aziz et al. (2005)

Aziz, S., Obitz, T., Modi, R., Sarkar, S.: Enterprise Architecture – A Governance Framework – Part I – Embedding Architecture into the Organization, Infosys, 2005.

Balzert (2000)

Balzert, H.: Lehrbuch der Software-Technik, Software-Entwicklung, 2. Aufl., Spektrum Akademischer Verlag, Heidelberg/Berlin, 2000.

Baumöl et al. (2005)

Baumöl, U., Österle, H., Winter, R.: Business Engineering in der Praxis, in: *Österle, H., Winter, R. (Hrsg.): Business Engineering in der Praxis*, Springer, Heidelberg, 2005, S. 1-13.

Becker (2004a)

Becker, J.: Referenzmodellierung - Aktuelle Methoden und Modelle, in: *Wirtschaftsinformatik*, 46(2004)5, S. 325-326.

Becker (2004b)

Becker, J.: Referenzmodellierung – Aktuelle Methoden und Modelle, in: *Wirtschaftsinformatik*, 46(2004)5, S. 325-326.

Becker et al. (2002a)

Becker, J., Algermissen, L., Delfmann, P., Knackstedt, R.: Referenzmodellierung, in: *Das Wirtschaftsstudium*, 30(2002)11, S. 1294-1298.

Becker/Delfmann (2004)

Becker, J., Delfmann, P.: Referenzmodellierung – Grundlagen, Techniken und domänenbezogene Anwendungen, Physica, Heidelberg, 2004.

Becker et al. (2002b)

Becker, J., Delfmann, P., Knackstedt, R., Kuropka, D.: Konfigurative Referenzmodellierung, in: *Becker, J., Knackstedt, R. (Hrsg.): Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung*, Physica, Heidelberg, 2002, S. 25-144.

Becker et al. (2001)

Becker, J., Holten, R., Knackstedt, R., Neumann, S.: Konstruktion von Methodiken – Vorschläge für eine begriffliche Grundlegung und domänenspezifische Anwendungsbeispiele, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Münster, Münster, 2001.

Becker et al. (2003)

Becker, J., Holten, R., Knackstedt, R., Niehaves, B.: Forschungsmethodische Positionierung in der Wirtschaftsinformatik – Epistemologische, ontologische und linguistische Leitfragen, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Münster, Münster, 2003.

Bernet (2005)

Bernet, U.: WGR Enterprise Architecture Metamodel, Winterthur Group, Winterthur, 2005.

Blechar (2005)

Blechar, M.: Magic Quadrant for OOA&D Tools, <http://www.gartner.com> (Zugriff: 28.04.2006).

Blechar/Sinur (2006)

Blechar, M., Sinur, J.: Magic Quadrant for Business Process Analysis Tools, <http://www.gartner.com> (Zugriff: 28.04.2006).

Böh/Meyer (2004)

Böh, A., Meyer, M.: IT-Balanced Scorecard – Ein Ansatz zur strategischen Ausrichtung der IT, in: *Zarnekow, R., Brenner, W., Grohmann, H. (Hrsg.): Informationsmanagement – Konzepte und Strategien für die Praxis*, 2004, S. 105-124.

Böhm et al. (2002)

Böhm, T., Junginger, M., Krcmar, H.: Modular Service Architectures: A Concept and Method for Engineering IT Services, in: *Proceedings of 36th Hawaii International Conference on System Sciences (HICSS'03)*, 2002.

BPMI (2002)

BPMI: Business Process Management Notation (BPMN) 1.0 Specification, (Zugriff: 18.07.2005).

Braun et al. (2004)

Braun, C., Hafner, M., Wortmann, F.: Methodenkonstruktion als wissenschaftlicher Erkenntnisansatz, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2004.

Braun/Winter (2005a)

Braun, C., Winter, R.: Classification of Outsourcing Phenomena in Financial Services, in: *Proceedings of the 13th European Conference on Information Systems*, Regensburg, 2005.

Braun/Winter (2005b)

Braun, C., Winter, R.: A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform, in: *Proceedings of the Workshop on Enterprise Modelling and Information Systems Architectures*, Klagenfurt, 2005, S. 64-79.

Braun et al. (2005)

Braun, C., Wortmann, F., Hafner, M., Winter, R.: Method Construction – A Core Approach to Organizational Engineering, in: *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC)*, Santa Fe, 2005, S. 1295-1299.

Brecht (2002)

Brecht, L.: Process Leadership – Methode des informationssystemgestützten Prozessmanagement, Verlag Dr. Kovac, Hamburg, 2002.

Brenner (1995)

Brenner, C.: Techniken und Metamodell des Business Engineering, Dissertation der Universität St. Gallen, St. Gallen, 1995.

Brenner (1994)

Brenner, W.: Grundzüge des Informationsmanagements, Springer, Berlin, 1994.

Brinkkemper (1996)

Brinkkemper, S.: Method engineering – engineering of information systems development methods and tools, in: *Information and Software Technology*, 38(1996)4, S. 275-280.

Buhl et al. (1999)

Buhl, H. U., Visser, V., Will, A.: Virtualisierung des Bankgeschäfts, in: *Wirtschaftsinformatik*, 41(1999)2, S. 116-123.

Bundesbank (2004)

Bundesbank: Internationale Konvergenz der Kapitalmessung und Eigenkapitalanforderungen, http://www.bundesbank.de/download/bankenaufsicht/pdf/eigenkapitalempfehlung_de.pdf (Zugriff: 04.04.2006).

Casewise (2005a)

Casewise: Casewise Modeler Suite Brochure, <http://www.casewise.com/downloads/presentations/> (Zugriff: 04.06.2005).

Casewise (2005b)

Casewise: Corporate Modeler, <http://www.casewise.com/products/corporate-modeler/index.php> (Zugriff: 04.06.2005).

Casewise (2005c)

Casewise: Corporate Modeler 10 Datasheet, <http://www.casewise.com/downloads/presentations/> (Zugriff: 04.06.2005).

Cheesman/Daniels (2000)

Cheesman, J., Daniels, J.: UML Components – A Simple Process for Specifying Component-Based Software, Addison-Wesley, 2000.

Chen (1976)

Chen, P. P.-S.: The Entity-Relationship Model – Toward a Unified View of Data, in: *ACM Transactions on Database Systems*, 1(1976)1, S. 9-36.

Choinowski et al. (2003)

Choinowski, S., Leist, S., Winter, R., Zellner, G.: BAI Methode, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2003.

CIO-Council (1996)

CIO-Council: Clinger-Cohen-Act, <http://www.defenselink.mil/nii/org/cio/doc/CCA-Book-Final.pdf> (Zugriff: 04.04.2006).

Coldewey (2000)

Coldewey, J.: Produktsysteme im Versicherungsbereich, in: *Proceedings of the Object Oriented Programming Conference (OOP2000)*, München, 2000.

Commerce (2000)

Commerce, O. o. G.: IT Infrastructure Library, The Stationary Office, London, 2000.

Dan et al. (2004)

Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., Youssef, A.: Web services on demand – WSLA-driven automated management, in: IBM Systems Journal, Special Issue on Utility Computing, 43(2004)1, S. 136-158.

Dinner/Kolber (2005)

Dinner, W., Kolber, A.: Zachman, Basel II and Sarbanes-Oxley, in: DM Review, Oktober, 2005, S. 43-47.

Dinter/Bucher (2006)

Dinter, B., Bucher, T.: Business Performance Management, in: *Chamoni, P., Gluchowski, P. (Hrsg.): Analytische Informationssysteme – Business Intelligence-Technologien und -Anwendungen*, Springer, Berlin et al., 2006, S. 23-50.

FEA (1999)

FEA: Federal Enterprise Architecture Framework, https://secure.cio.noaa.gov/hpcc/docita/files/federal_enterprise_arch_framework.pdf (Zugriff: 04.06.2005).

FEA (2001)

FEA: A Guide to Federal Enterprise Architecture, http://www.eaframeworks.com/FEAF/a_practical_guide_to_federal_enterprise_architecture.pdf (Zugriff: 04.06.2005).

Ferraiolo et al. (2001)

Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, R. D., Chandramouli, R.: Proposed NIST Standard for Role-Based Access Control, in: ACM Transactions on Information and System Security (TISSEC), 4(2001)3, S. 224-274.

Ferstl/Sinz (1995)

Ferstl, O. K., Sinz, E. J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen, in: Wirtschaftsinformatik, 37(1995)3, S. 209-220.

Ferstl/Sinz (1998)

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

Ferstl/Sinz (2001)

Ferstl, O. K., Sinz, E. J.: Grundlagen der Wirtschaftsinformatik, Oldenbourg, München, 2001.

Fettke/Loos (2002)

Fettke, P., Loos, P.: Methoden zur Wiederverwendung von Referenzmodellen – Übersicht und Taxonomie, in: Tagungsband zur 6. Fachtagung Referenzmodellierung 2002 (RefMod 2002), Nürnberg, 2002, S. 9-33.

Fettke/Loos (2003)

Fettke, P., Loos, P.: Ontological evaluation of reference models using the bunge-wand-weber model, in: Proceedings of the Ninth Americas Conference on Information Systems, Atlanta, 2003, S. 2944-2955.

Fettke/Loos (2004a)

Fettke, P., Loos, P.: Entwicklung eines Bezugsrahmens zur Evaluierung von Referenzmodellen – Langfassung eines Beitrages, Arbeitsbericht, Information Systems & Management, Universität Mainz, Mainz, 2004.

Fettke/Loos (2004b)

Fettke, P., Loos, P.: Referenzmodellierungsforschung, in: Wirtschaftsinformatik, 46(2004)5, S. 331-340.

Firesmith et al. (1998)

Firesmith, D., Henderson-Sellers, B., Graham, I.: Open Modeling Language Reference Manual, SIGS Books, 1998.

Frank (1994)

Frank, U.: Multiperspektivische Unternehmensmodellierung – Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung, Oldenbourg, München, 1994.

Frank (1995)

Frank, U.: MEMO – Objektorientierte Unternehmensmodellierung zum gemeinsamen Entwurf optimierter Geschäftsprozesse und hochintegrierter Anwendungssysteme, in: Objekt-Spektrum, Nr. 6, 1995, S. 43-47.

Frank (1998a)

Frank, U.: The MEMO Meta-Metamodel, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Koblenz-Landau, Koblenz, 1998.

Frank (1998b)

Frank, U.: The MEMO Object Modeling Language (MEMO-OML), Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Koblenz-Landau, Koblenz, 1998.

Frank (1999a)

Frank, U.: Applying the MEMO-OML – Guidelines and Examples, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Koblenz-Landau, Koblenz, 1999.

Frank (1999b)

Frank, U.: Eine Architektur zur Spezifikation von Sprachen und Werkzeugen für die Unternehmensmodellierung, in: Proceedings der Fachtagung Modellierung betrieblicher Informationssysteme (MobIS), Bamberg, 1999, S. 154-169.

Frank (2002)

Frank, U.: Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages, in: Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35), Honolulu, 2002, S. 72.

Frank/Van Laak (2003)

Frank, U., Van Laak, B.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Koblenz-Landau, Koblenz, 2003.

Frese (1995)

Frese, E.: Grundlagen der Organisation. Konzept – Prinzipien – Strukturen, 6. Aufl., Gabler, Wiesbaden, 1995.

Frie/Auth (2001)

Frie, T., Auth, G.: Kopplung operativer (horizontaler) Applikationen mit dem Data Warehouse, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2001.

Geissler (2004)

Geissler, C.: Compliance Management?, in: Harvard Business Manager, Nr. 2, 2004, S. 17.

Grant (1998)

Grant, R. M.: Contemporary strategy analysis – Concepts, techniques, applications. A guide for instructors, Blackwell Publishers, 1998.

Greenfield/Short (2003)

Greenfield, J., Short, K.: Software factories – Assembling applications with patterns, models, frameworks and tools, in: Companion of the 18th annual ACM SIGPLAN conference on object-oriented programming, systems, languages, and applications, Anaheim, 2003, S. 16-27.

Greiffenberg (2003)

Greiffenberg, S.: Methoden als Theorien der Wirtschaftsinformatik, in: *Uhr, W., Esswein, W., Schoop, E. (Hrsg.):* Wirtschaftsinformatik 2003 Band II, Physica-Verlag, Heidelberg, 2003, S. 947-968.

Gutzwiller (1994)

Gutzwiller, T.: Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica, Heidelberg, 1994.

Hafner (2005)

Hafner, M.: Entwicklung einer Methode für das Management der Informationssystemarchitektur im Unternehmen, Dissertation der Universität St. Gallen, St. Gallen, 2005.

Hafner et al. (2006)

Hafner, M., Schelp, J., Winter, R.: Berücksichtigung des Architekturmanagements in serviceorientierten IT-Managementkonzepten am Beispiel von ITIL, in: *Schelp, J., Winter, R. (Hrsg.):* Integrationsmanagement, Springer, Berlin et al., 2006, S. 99-121.

Hahn (1997)

Hahn, D.: US-amerikanische Konzepte strategischer Unternehmensführung, in: *Hahn, D., Taylor, B. (Hrsg.):* Strategische Unternehmensplanung – strategische Unternehmensführung – Stand und Entwicklungstendenzen, Physica, Heidelberg, 1997, S. 144-164.

Hammer/Champy (1993)

Hammer, M., Champy, J.: Reengineering the Corporation – A Manifest for Business Revolution, Allen & Unwin, 1993.

Harengel (2000)

Harengel, J.: Die Balanced Scorecard als Instrument des Banken-Controlling, Dissertation der Universität Konstanz, Konstanz, 2000.

Hegering et al. (1999)

Hegering, H.-G., Abeck, S., Neumair, B.: Integrated Management of Networked Systems – Concepts, Architectures, and Their Operational Application, Morgan Kaufmann, 1999.

Heinrich (2000)

Heinrich, B.: Dimensionen zur Beschreibung eines Geschäftsmodells für Kreditinstitute im Bereich Privatkunden, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2000.

Heinrich (2002a)

Heinrich, B.: Das Geschäftsmodell als Instrument zur Positionierung des Unternehmens, in: *Leist, S., Winter, R. (Hrsg.):* Retail Banking im Informationszeitalter – Integrierte Gestaltung der Geschäfts-, Prozess- und Applikationsebene, Springer, Berlin et al., 2002, S. 53-72.

Heinrich (2002b)

Heinrich, B.: Die konzeptionelle Gestaltung des Multichannel-Vertriebs anhand von Kundenbedürfnissen, in: *Leist, S., Winter, R. (Hrsg.):* Retail Banking im Informationszeitalter – Integrierte Gestaltung der Geschäfts-, Prozess- und Applikationsebene, Springer, Berlin et al., 2002, S. 73-92.

Heinrich/Winter (2004)

Heinrich, B., Winter, R.: A Strategy Modelling Technique for Financial Services, in: Proceedings of the 12th European Conference on Information Systems, Turku, 2004.

Heinrich (1993)

Heinrich, L. J.: Wirtschaftsinformatik – Einführung und Grundlegung, Oldenbourg, München, 1993.

Heinrich (2002c)

Heinrich, L. J.: Informationsmanagement – Planung, Überwachung und Steuerung der Informationsinfrastruktur, 7. Aufl., Oldenbourg, München, 2002.

Henderson/Thomas (1992)

Henderson, J. C., Thomas, J.: Aligning Business and Information Technology Domains – Strategic Planning in Hospitals, in: Hospital and Health Services Administrative, 37(1992)1, S. 71-87.

Henderson/Venkatraman (1993)

Henderson, J. C., Venkatraman, N.: Strategic alignment – Leveraging information technology for transforming organizations, in: IBM Systems Journal, 32(1993)1, S. 4-16.

Hess (1996)

Hess, T.: Entwurf betrieblicher Prozesse, Dissertation der Universität St. Gallen, St. Gallen, 1996.

Hess/Brecht (1996)

Hess, T., Brecht, L.: State of the Art des Business Process Redesign – Darstellung und Vergleich bestehender Methoden, 2. Aufl., Gabler, Wiesbaden, 1996.

Hesse et al. (1992)

Hesse, W., Merbeth, G., Frölich, R.: Software-Entwicklung, Vorgehensmodelle, Projektführung, Produktverwaltung, in: Handbuch der Informatik, Oldenbourg-Verlag, München, 1992.

Hevner et al. (2004)

Hevner, A. R., March, S. T., Park, J.: Design Science in Information Systems Research, in: MIS Quarterly, 28(2004)1, S. 75-105.

Heym (1993)

Heym, M.: Methoden-Engineering – Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme, Rosch-Buch, Hallstadt, 1993.

Hochstein/Hunziker (2003)

Hochstein, A., Hunziker, A.: Serviceorientierte Referenzmodelle des IT-Managements, in: HMD – Praxis der Wirtschaftsinformatik, 40(2003)232, S. 46-56.

Hochstein et al. (2004)

Hochstein, A., Zarnekow, R., Brenner, W.: Serviceorientiertes IT-Management nach ITIL – Möglichkeiten und Grenzen, in: HMD – Praxis Der Wirtschaftsinformatik, 41(2004)239, S. 68-76.

Holten et al. (1997)

Holten, R., Striemer, R., Weske, M.: Ansätze zur Entwicklung von Workflow-basierten Anwendungssystemen, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität Münster, Münster, 1997.

Horváth&Partners (2005)

Horváth&Partners: Balanced-Scorecard-Studie 2005, <http://www.horvath-partners.com/hp3/1709153/2336048.html> (Zugriff: 24.02.2006).

IBM-Corporation (1984)

IBM-Corporation: Business Systems Planning – Information Systems Planning Guide, IBM-Form GE20-0527-4, 4. Aufl., Atlanta, 1984.

IDS-Scheer (2005a)

IDS-Scheer: ARIS Process Platform, <http://www.ids-scheer.com/germany/products/23019> (Zugriff: 04.06.2005).

IDS-Scheer (2005b)

IDS-Scheer: Präsentation ARIS Design Platform, http://www.ids-scheer.com/germany/products/aris_design_platform/25646 (Zugriff: 04.05.2005).

IDS-Scheer (2005c)

IDS-Scheer: White Paper ARIS Design Platform, http://www.ids-scheer.com/germany/products/aris_design_platform/29482 (Zugriff: 04.06.2005).

IEEE (2000)

IEEE: IEEE Recommended Practice for Architectural Description of Software Intensive Systems, IEEE Std 1471-2000, 2000.

IFEAD (2004)

IFEAD: Trends in Enterprise Architecture 2004: How are Organisations Progressing?, <http://www.enterprise-architecture.info/Images/EA%20Survey/EA%20Survey%202004%20IFEAD.PDF> (Zugriff: 04.05.2005).

IFEAD (2005)

IFEAD: Trends in Enterprise Architecture 2005 – How are Organisations Progressing?, <http://www.enterprise-architecture.info/Images/EA%20Survey/Enterprise%20Architecture%20Survey%202005%20IFEAD%20v10.pdf> (Zugriff: 20.02.2006).

IFEAD (2006)

IFEAD: Enterprise Architecture Tools Overview 2006, http://www.enterprise-architecture.info/EA_Tools.htm (Zugriff: 08.06.2006).

IMG (1997)

IMG: PROMET BPR – Methodenhandbuch für den Entwurf von Geschäftsprozessen, Version 2.0, Information Management Group, St. Gallen, 1997.

IMG (1999)

IMG: PROMET STP – Methodenhandbuch für die System- und Technologieplanung, Release 1.0, Information Management Group, St. Gallen, 1999.

Infosys (2005)

Infosys: Infosys Enterprise Architecture Survey, <http://www.infosys.com/services/systemintegration/ea-survey/ea-survey-executive-summary.pdf> (Zugriff: 05.05.2006).

Jacobson et al. (1994)

Jacobson, I., Ericsson, M., Jacobson, A.: The Object Advantage – Business Process Reengineering With Object Technology, Addison-Wesley, Wokingham, 1994.

James (2004a)

James, G.: Architecture Frameworks – Tool Support, www.gartner.com (Zugriff: 19.04.2005).

James (2004b)

James, G.: Market Scope – Enterprise Architecture Tool Market, www.gartner.com (Zugriff: 19.04.2005).

James (2005a)

James, G.: Magic Quadrant for Enterprise Architecture Tools, www.gartner.com (Zugriff: 19.04.2005).

James (2005b)

James, G.: Vendor Details for the 4Q04 Enterprise Architecture Tool Magic Quadrant, www.gartner.com (Zugriff: 19.04.2005).

Jeckle et al. (2004a)

Jeckle, M., Rupp, C., Hahn, J., Zenger, B., Queins, S.: UML 2.0 – Evolution oder Degeneration?, in: Objekt Spektrum, Nr. 3, 2004, S. 12-19.

Jeckle et al. (2004b)

Jeckle, M., Rupp, C., Hahn, J., Zengler, B., Queins, S.: UML 2 glasklar, Hanser, München, 2004.

Jonkers et al. (2004)

Jonkers, H., Lankhorst, M., van Buuren, R., Hoppenbrouwers, S., Bonsangue, M., van der Torre, L.: Concepts for Modelling Enterprise Architectures, in: International Journal of Cooperative Information Systems, 13(2004)3, S. 257-287.

Jonscher/Dittrich (1994)

Jonscher, D., Dittrich, K.: Realisierung von Sicherheitsstrategien mit Hilfe flexibler Zugriffskontrollmechanismen, in: *Bauknecht, K., Dittrich, K. (Hrsg.):* Sicherheit in Informationssystemen, vdf, Zürich, 1994, S. 23-52.

Jung (2004)

Jung, E.: Ein unternehmensweites IT-Architekturmodell als erfolgreiches Bindeglied zwischen der Unternehmensstrategie und dem operativen Bankgeschäft, in: Wirtschaftsinformatik, 46(2004)4, S. 313-314.

Junginger et al. (2000)

Junginger, S., Kühn, H., Strobl, R., Karagiannis, D.: Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation, in: Wirtschaftsinformatik, 42(2000)5, S. 392-401.

Kaiser (2000)

Kaiser, T. M.: Methode zur Konzeption von Intranets, Dissertation der Universität St. Gallen, St. Gallen, 2000.

Kalakota/Robinson (2001)

Kalakota, R., Robinson, M.: e-Business 2.0 – Roadmap for success, 2. Aufl., Addison Wesley, Boston et al., 2001.

Kaplan/Norton (1996)

Kaplan, R. S., Norton, D. P.: Using the Balanced Scorecard as a Strategic Management System, in: Harvard Business Review, January, 1996, S. 75-85.

Kaplan/Norton (1997)

Kaplan, R. S., Norton, D. P.: Balanced Scorecard, Schäffer-Poeschel, Stuttgart, 1997.

Karagiannis/Kühn (2002)

Karagiannis, D., Kühn, H.: Metamodelling Platforms, in: Proceedings of the Third International Conference on Electronic Commerce and Web Technologies (EC-Web), Aix-en-Provence, 2002, S. 182-196.

Keller et al. (1992)

Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage „Ereignisgesteuerter Prozessketten (EPK)“, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität des Saarlandes, Saarbrücken, 1992.

Kelly/Smolander (1996)

Kelly, S., Smolander, K.: Evolution and issues in metaCASE, in: Information and Software Technology, 38(1996)4, S. 161-266.

Kern et al. (2002)

Kern, A., Kuhlmann, M., Schaad, A., Moffett, J. D.: Observations on the role life-cycle in the context of enterprise security management, in: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, New York, 2002, S. 43-51.

Kitchenham et al. (1995)

Kitchenham, B., Pickard, L., Pfleeger, S. L.: Case Studies for Method and Tool Evaluation, in: IEEE Software, 12(1995)4, S. 52-62.

König et al. (1996)

König, W., Heinzl, A., Rumpf, M., von Poblitzki, A.: Zur Entwicklung der Forschungsmethoden und Theoriekerne der Wirtschaftsinformatik in den nächsten zehn Jahren – Eine kombinierte Delphi- und AHP-Untersuchung, in: *Heilmann, H., Heinrich, L. J., Roithmayer, F. (Hrsg.):* Information Engineering, München et al., 1996, S. 36-65.

Krcmar (1990)

Krcmar, H.: Bedeutung und Ziele von Informationssystemarchitekturen, in: Wirtschaftsinformatik, 32(1990)5, S. 395-402.

Krcmar (2003)

Krcmar, H.: Informationsmanagement, 3. Aufl., Springer, Berlin, 2003.

Kremer (2004)

Kremer, S.: Information Retrieval in Portalen – Gestaltungselemente, Praxisbeispiele und Methodenvorschlag, Dissertation der Universität St. Gallen, St. Gallen, 2004.

Kühn/Bayer (2004)

Kühn, H., Bayer, F.: Integration Approaches for Metamodelling Platforms, in: Proceedings of the Workshop on "Enterprise Modeling and Ontology – Ingredients for Interoperability", in Conjunction with the Fifth International Conference on Practical Aspects of Knowledge Management (PAKM 2004), Vienna, 2004, S. 51-59.

Kühn et al. (2003)

Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise Model Integration, in: Proceedings of the Fourth International Conference on Electronic Commerce and Web Technologies (EC-Web), Prague, 2003, S. 379-392.

Kühn et al. (1999)

Kühn, H., Junginger, S., Petersen, C.: Re-Use in Business Process Management based on Metamodeling and Model Views, in: Proceedings of the Eighth International Conference on Human-Computer Interaction, Munich, 1999, S. 173-174.

Kühn/Karagiannis (2005)

Kühn, H., Karagiannis, D.: Strategie-, Prozess- und IT-Management – Ein Pattern-orientierter Integrationsansatz, in: *Ferstl, O. K., Sinz, E. J., Eckert, S., Isselhorst, T. (Hrsg.):* Wirtschaftsinformatik 2005, Physica, Heidelberg, 2005, S. 1483-1502.

Kühn et al. (2004)

Kühn, H., Murzek, M., Bayer, F.: Horizontal Business Process Model Interoperability using Model Transformation, in: Proceedings of the Workshop on Interoperability of Enterprise Systems at ECOOP 2004, Oslo, 2004.

Lankes et al. (2005)

Lankes, J., Matthes, F., Wittenburg, A.: Softwarekartographie – Systematische Darstellung von Anwendungslandschaften, in: *Ferstl, O. K., Sinz, E. J., Eckert, S., Isselhorst, T. (Hrsg.):* Wirtschaftsinformatik 2005, Physica Verlag, Heidelberg, 2005, S. 1443-1462.

Lankhorst (2005)

Lankhorst, M.: Enterprise Architecture at Work, Springer, Berlin/Heidelberg, 2005.

Legner (1999)

Legner, C.: Benchmarking informationssystemgestützter Geschäftsprozesse, Dissertation der Universität St. Gallen, St. Gallen, 1999.

Lehner (1995)

Lehner, F.: Grundfragen und Positionierung der Wirtschaftsinformatik, in: *Lehner, F., Hildebrand, K., Maier, R. (Hrsg.):* Wirtschaftsinformatik, Hanser, München/Wien, 1995, S. 1-72.

Leidecker/Bruno (1984)

Leidecker, J., Bruno, A.: Identifying and Using Critical Success Factors, in: Long Range Planning, 17(1984)1, S. 23-32.

Leist (2002)

Leist, S.: Bankenarchitektur des Informationszeitalters - Zielsetzung und Gestaltungsebenen, in: *Leist, S., Winter, R. (Hrsg.): Retail Banking im Informationszeitalter*, Springer, Berlin, 2002, S. 4-28.

Leist (2004)

Leist, S.: Methoden zur Unternehmensmodellierung – Vergleich, Anwendungen und Diskussionen der Integrationspotenziale, Habilitation der Universität St. Gallen, St. Gallen, 2004.

Leist/Winter (2000)

Leist, S., Winter, R.: Finanzdienstleistungen im Informationszeitalter – Vision, Referenzmodell und Transformation, in: *Belz, C., Bieger, T. (Hrsg.): Dienstleistungskompetenz und innovative Geschäftsmodelle*, Thexis, St.Gallen, 2000, S. 150-166.

Leist/Winter (2002)

Leist, S., Winter, R.: Retail Banking im Informationszeitalter, Springer, Berlin et al., 2002.

Lorenz (1995)

Lorenz, K.: Methode, in: *Mittelstrass, J. (Hrsg.): Enzyklopädie Philosophie und Wissenschaftstheorie*, Stuttgart, 1995, S. 876-879.

Luftman et al. (1993)

Luftman, J. N., Lewis, P. R., Oldach, S. H.: Transforming the enterprise – The alignment of business and information technology strategies, in: *IBM Systems Journal*, 32(1993)1, S. 198-221.

Luftman et al. (1999)

Luftman, J. N., Papp, R., Brier, T.: Enablers and inhibitors of business-IT alignment, in: *Communications of the Association for Information Systems (AIS)*, 1(1999)3, S. 2-32.

Lühns (2001)

Lühns, J.-C.: Strategische Unternehmensführung im Kontext hoher Marktturbulenz – Entwicklung eines Systematisierungsmodells am Beispiel von Netzwerkbranchen, Deutscher Universitäts Verlag, Wiesbaden, 2001.

Marr/Neely (2001)

Marr, B., Neely, A.: Balanced Scorecard Software Report, Infoedge Inc., 2001.

Masak (2005)

Masak, D.: Moderne Enterprise Architekturen, Springer, Berlin/Heidelberg, 2005.

Mayerl et al. (2005)

Mayerl, C., Link, S., Racke, M., Popescu, S., Vogel, T., Mehl, O., Abeck, S.: Methode für das Design von SLA-fähigen IT-Services, in: *Proceedings der GI/ITG-Fachtagung Kommunikation in verteilten Systemen (KiVS)*, Kaiserslautern, 2005, S. 271-282.

Mesarovic et al. (1970)

Mesarovic, M. D., Macko, D., Takahara, Y.: Theory of Hierarchical, Multilevel Systems, Academic Press, New York/London, 1970.

Mintzberg (1987)

Mintzberg, H.: The strategy concept I: five p's for strategy, in: California Management Review, Herbst, 1987, S. 11-24.

Müller-Stewens/Lechner (2003)

Müller-Stewens, G., Lechner, C.: Strategisches Management – Wie strategische Initiativen zum Wandel führen, 2. Aufl., Schäffer-Poeschel, Stuttgart, 2003.

Nadler et al. (1992)

Nadler, D. A., Gerstein, M. S., Shaw, R. B.: Organizational Architecture – Designs for Changing Organizations, Jossey-Bass, San Francisco, 1992.

Nieschlag et al. (1994)

Nieschlag, R., Dichtl, E., Hörschgen, H.: Marketing, Duncker&Humblot, Berlin, 1994.

NIST (1993)

NIST: Integration Definition For Function Modeling (IDEF0), <http://www.idef.com/pdf/idef0.pdf> (Zugriff: 11.05.2006).

Nuseibeh et al. (1996)

Nuseibeh, B., Finkelstein, A., Kramer, S.: Method engineering for multi-perspective software development, in: Information and Software Technology, 38(1996)4, S. 267-274.

Nüttgens/Rump (2002)

Nüttgens, M., Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK), in: Proceedings des GI-Workshops Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen (Promise 2002), Bonn, 2002, S. 64-77.

Oestereich et al. (2004)

Oestereich, B., Weiss, C., Schröder, C., Weilkiens, T., Lenhard, A.: Objektorientierte Geschäftsprozessmodellierung mit der UML, Dpunkt-Verlag, Heidelberg, 2004.

OMG (2003a)

OMG: Model Driven Architecture (MDA) FAQ, http://www.omg.org/mda/faq_mda.htm (Zugriff: 04.05.2005).

OMG (2003b)

OMG: UML 2.0 Superstructure Specification, <http://www.omg.org/docs/ptc/03-08-02.pdf> (Zugriff: 06.06.2005).

OMG (2005)

OMG: UML 2.0 Superstructure Specification, <http://www.omg.org/docs/formal/05-07-04.pdf> (Zugriff: 24.03.2006).

Opengroup (2002)

Opengroup: The Open Group Architecture Framework "Enterprise Edition" Version 8.1, <http://www.opengroup.org/architecture/togaf8-doc/arch/> (Zugriff: 05.06.2005).

Orlikowski/Barley (2001)

Orlikowski, W. J., Barley, S. R.: Technology and Institutions – What Can Research on Information Technology and Research on Organizations Learn From Each Other?, in: *MIS Quarterly*, 25(2001)2, S. 145-165.

Österle (1995)

Österle, H.: Business Engineering – Prozess- und Systementwicklung, 2. Aufl., Springer, Berlin et al., 1995.

Österle (2003)

Österle, H.: Geschäftsmodell des Informationszeitalters, in: *Österle, H., Winter, R. (Hrsg.): Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters*, Springer, Berlin et al., 2003, S. 22-43.

Österle/Blessing (2003)

Österle, H., Blessing, D.: Business Engineering Modell, in: *Österle, H., Winter, R. (Hrsg.): Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters*, Springer, Berlin et al., 2003, S. 65-86.

Österle et al. (2001)

Österle, H., Fleisch, E., Alt, R.: Business Networking – Shaping Collaboration Between Enterprises, 2. Aufl., Springer, Heidelberg, 2001.

Österle/Winter (2003)

Österle, H., Winter, R.: Business Engineering, in: *Österle, H., Winter, R. (Hrsg.): Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters*, Springer, Berlin et al., 2003, S. 3-20.

Popkin (2003)

Popkin: Working with System Architect V9 Repository, http://www.popkin.com/customers/customer_service_center/downloads/documentation.htm (Zugriff: 04.05.2005).

Popkin (2004)

Popkin: User Guide, http://www.popkin.com/customers/customer_service_center/downloads/documentation.htm (Zugriff: 04.05.2005).

Porter (1999)

Porter, M. E.: Wettbewerbsvorteile – Spitzenleistungen erreichen und behaupten, Campus Verlag, Frankfurt a.M., 1999.

Rechenberg/Pomberger (1999)

Rechenberg, P., Pomberger, G.: Informatik-Handbuch, 2. Aufl., Hanser, München/Wien, 1999.

Reichmann (1997)

Reichmann, T.: Controlling mit Kennzahlen und Managementberichten – Grundlagen einer systemgestützten Controlling-Konzeption, 5. Aufl., Vahlen, München, 1997.

Rockart (1979)

Rockart, J.: Chief executives define their own data needs, in: Harvard Business Review, 57(1979)2, S. 81-93.

Rosemann (1995)

Rosemann, M.: Erstellung und Integration von Prozessmodellen – Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Dissertation der Universität Münster, Münster, 1995.

Rosemann (1996)

Rosemann, M.: Komplexitätsmanagement in Prozessmodellen – Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Gabler, Wiesbaden, 1996.

Rosemann/Green (2002)

Rosemann, M., Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs, in: Information Systems, 27(2002)2, S. 75-91.

Rosemann/zur Mühlen (1997)

Rosemann, M., zur Mühlen, M.: Modellierung der Aufbauorganisation in Workflow-Management-Systemen – Kritische Bestandsaufnahme und Gestaltungsvorschläge, in: Proceedings des EMISA-Fachgruppentreffens, Aachen, 1997, S. 78-84.

Rupprecht (2002)

Rupprecht, J.: Datensicherheit im Data Warehousing – Grundlagen, Zugriffskontrolle, Fallbeispiele, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2002.

Samarati/de Capitani di Vimercati (2002)

Samarati, P., de Capitani di Vimercati, S.: Access Control – Policies, Models and Mechanisms, in: *Focardi, R., Gorrieri, R. (Hrsg.): Foundations of Security Analysis and Design – Tutorial Lectures*, Springer, Berlin, 2002.

Sarbanes-Oxley (2002)

Sarbanes-Oxley: Sarbanes-Oxley Act of 2002,
<http://www.law.uc.edu/CCL/SOact/soact.pdf> (Zugriff: 04.04.2006).

Scheer (1998)

Scheer, A.-W.: ARIS – Vom Geschäftsprozess zum Anwendungssystem, Springer, Berlin et al., 1998.

Scheer (2001)

Scheer, A.-W.: ARIS – Modellierungsmethoden, Metamodelle, Anwendungen, 4. Aufl., Springer, Berlin/Heidelberg, 2001.

Scheer/Jost (2002)

Scheer, A.-W., Jost, W.: ARIS in der Praxis – Gestaltung, Implementierung und Optimierung von Geschäftsprozessen, Springer, Berlin/Heidelberg, 2002.

Scheer/Thomas (2005)

Scheer, A.-W., Thomas, O.: Geschäftsprozessmodellierung mit der ereignisgesteuerten Prozesskette, in: *Das Wirtschaftsstudium*, 34(2005)8-9, S. 1069-1078.

Schekkerman (2004)

Schekkerman, J.: How to Survive in the Jungle of Enterprise Architecture Frameworks – Creating or Choosing an Enterprise Architecture Framework, 2. Aufl., Trafford Publishing, Victoria, 2004.

Schelp (2000)

Schelp, J.: Konzeptionelle Modellierung mehrdimensionaler Datenstrukturen analyseorientierter Informationssysteme, Dissertation der Ruhr-Universität Bochum, Bochum, 2000.

Schelp et al. (2004)

Schelp, J., Hafner, M., Winter, R.: Architekturmanagement als Basis effizienter und effektiver Produktion von IT-Services, in: *Hmd – Praxis der Wirtschaftsinformatik*, 41(2004)237, S. 54-66.

Schelp/Schwinn (2005)

Schelp, J., Schwinn, A.: Extending the Business Engineering Framework for Application Integration Purposes, in: *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC)*, New York, 2005, S. 1333-1337.

Schemm et al. (2006)

Schemm, J., Heutschi, R., Vogel, T., Wende, K., Legner, C.: Serviceorientierte Architekturen – Einordnung im Business Engineering, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2006.

Schlitt (2003)

Schlitt, M.: Grundlagen und Methoden für Interpretation und Konstruktion von Informationssystemmodellen, Dissertation der Universität Bamberg, Bamberg, 2003.

Scholz (1998)

Scholz, C.: Von der Netzwerkkooperation zur Virtualisierung, in: *Winand, U., Nathusius, K. (Hrsg.): Unternehmensnetzwerke und virtuelle Organisationen*, Schaeffel-Poeschel, Stuttgart, 1998, S. 95-108.

Schönsleben/Leuzinger (1996)

Schönsleben, P., Leuzinger, R.: Innovative Gestaltung von Versicherungsprodukten – Flexible Industriekonzepte in der Assekuranz, Gabler Verlag, Wiesbaden, 1996.

Schütte (1998)

Schütte, R.: Grundsätze ordnungsmässiger Referenzmodellierung, Gabler, Wiesbaden, 1998.

Schütte et al. (1999)

Schütte, R., Siedentopf, J., Zelewski, S.: Wirtschaftsinformatik und Wissenschaftstheorie – Grundpositionen und Theoriekerne, Arbeitsbericht, Institut für Produktion und industrielles Informationsmanagement, Universität Essen, Essen, 1999.

Schwinn (2005)

Schwinn, A.: Entwicklung einer Methode zur Applikationsintegration in heterogenen Informationssystemen, Dissertation der Universität St. Gallen, St. Gallen, 2005.

Schwinn/Winter (2005)

Schwinn, A., Winter, R.: Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration, in: *Ferstl, O. K., Sinz, E. J., Eckert, S., Isselhorts, T. (Hrsg.):* Wirtschaftsinformatik 2005, Physica, Bamberg, 2005, S. 587-606.

Senger/Österle (2004)

Senger, E., Österle, H.: PROMET Business Engineering Case Studies (BECS) Version 2.0, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2004.

Seufert (2001)

Seufert, S.: Die Zugriffskontrolle, Dissertation der Universität Bamberg, Bamberg, 2001.

Siegenthaler/Schwinn (2005)

Siegenthaler, A., Schwinn, A.: Modellierung von Integrationsaspekten in Applikationslandschaften, in: *Schelp, J., Winter, R. H. (Hrsg.):* Integrationsmanagement – Planung, Bewertung und Steuerung von Applikationslandschaften, Springer, Berlin et al., 2005, S. 203-230.

Sinz (1999a)

Sinz, E. J.: Architektur von Informationssystemen, in: *Rechenberg, P., Pomberger, G. (Hrsg.):* Informatik-Handbuch, Hanser, München/Wien, 1999, S. 1035-1046.

Sinz (1999b)

Sinz, E. J.: Architektur von Informationssystemen, in: *Rechenberg, P., Pomberger, G. (Hrsg.):* Informatik-Handbuch, Hanser, München, Wien, 1999, S. 1035-1046.

Sledgianowski/Luftman (2005)

Sledgianowski, D., Luftman, J. N.: IT-Business Strategic Alignment Maturity – A Case Study, in: *Journal of Cases on Information Technology*, 7(2005)2, S. 102-120.

Spewak/Hill (1993)

Spewak, S. H., Hill, S. C.: Enterprise Architecture Planning – Developing a Blueprint for Data, Applications and Technology, John Wiley & Sons, New York, 1993.

Stachowiak (1973)

Stachowiak, H.: Allgemeine Modelltheorie, Springer, New York, Wien, 1973.

Stahlknecht/Hasenkamp (1999)

Stahlknecht, P., Hasenkamp, U.: Einführung in die Wirtschaftsinformatik, Springer, Berlin, 1999.

Stojanovic (2005)

Stojanovic, Z.: A Method for Component-Based and Service-Oriented Software Systems Engineering, Dissertation der Delft University of Technology, Delft, 2005.

Strahringer (1998)

Strahringer, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips, in: Modellierung 98, Proceedings; Bericht Nr. 6/98-I, Angewandte Mathematik und Informatik, Münster, 1998, S. 15-20.

Strauch (2002)

Strauch, B.: Entwicklung einer Methode für die Informationsbedarfsanalyse im Data Warehousing, Dissertation der Universität St. Gallen, St. Gallen, 2002.

Strauch/Winter (2002)

Strauch, B., Winter, R.: Vorgehensmodell für die Informationsbedarfsanalyse im Data Warehousing, in: *von Maur, E., Winter, R. (Hrsg.): Vom Data Warehouse zum Corporate Knowledge Center*, Physica, Heidelberg, 2002, S. 359-378.

ter Doest/Lankhorst (2004)

ter Doest, H., Lankhorst, M.: Tool Support for Enterprise Architecture, Telematica Instituut, Enschede, 2004.

Teubner (1999)

Teubner, R. A.: Organisations- und Informationssystemgestaltung – Theoretische Grundlagen und integrierte Methoden, DUV, Wiesbaden, 1999.

Ulrich (2001)

Ulrich, H.: Systemorientiertes Management – Das Werk von Hans Ulrich, Haupt, Bern et al., 2001.

van Hee (1994)

van Hee, K. M.: Information Systems Engineering – A Formal Approach, Cambridge University Press, Cambridge, 1994.

Varveris/Harrison (2005)

Varveris, L., Harrison, D.: Building Enterprise Architectures with TOGAF, Telelogic, 2005.

Vogler (2003)

Vogler, P.: Prozess- und Systemintegration – Umsetzung des organisatorischen Wandels in Prozessen und Informationssystemen, Habilitation der Universität St. Gallen, St. Gallen, 2003.

Völter (2003)

Völter, M. W., E.: Model Driven Architecture – Ein neuer Ansatz für Wiederverwendung?, <http://www.voelter.de/data/articles/mda.pdf> (Zugriff: 04.05.2005).

vom Brocke (2003a)

vom Brocke, J.: Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen, Logos Verlag, Berlin, 2003.

vom Brocke (2003b)

vom Brocke, J.: Referenzmodellierung – Gestaltung und Verteilung von Konstruktionsprozessen, Logos Verlag, Berlin, 2003.

W3C (2004a)

W3C: Web Services Architecture, W3C Working Group Note 11 February 2004, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/> (Zugriff: 20.02.2006).

W3C (2004b)

W3C: Web Services Glossary, W3C Working Group Note 11 February 2004, <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/> (Zugriff: 20.02.2006).

Weber et al. (1997)

Weber, J., Kummer, S., Grossklaus, A., Nippel, H., Warnke, D.: Methodik und Generierung von Logistik-Kennzahlen, in: Betriebswirtschaftliche Forschung und Praxis, Nr. 4, 1997, S. 438-454.

Winter (2003a)

Winter, R.: An Architecture Model for Supporting Application Integration Decisions, in: Proceedings of the 11th European Conference on Information Systems (ECIS), Naples, 2003.

Winter (2003b)

Winter, R.: Conceptual Modeling of Business Networks and Business Strategies, in: Proceedings of the 16th Bled eCommerce Conference on eTransformation, Bled, 2003, S. 551-568.

Winter (2003c)

Winter, R.: Methodische Unterstützung der Strategiebildung im Retail Banking, in: BIT – Banking and Information Technology, 4(2003)2, S. 49-58.

Winter (2003d)

Winter, R.: Modelle, Techniken und Werkzeuge im Business Engineering, in: *Österle, H., Winter, R. (Hrsg.): Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters*, Springer, Berlin et al., 2003, S. 87-118.

Winter (2004a)

Winter, R.: Modellierung auf Prozessebene, Executive MBA in Business Engineering, St. Gallen, 2004.

Winter (2004b)

Winter, R.: Modellierung auf Strategieebene, Executive MBA in Business Engineering, St. Gallen, 2004.

Winter (2004c)

Winter, R.: Modellierung auf Systemebene, Executive MBA in Business Engineering, St. Gallen, 2004.

Winter (2005)

Winter, R.: Unternehmensarchitektur und Integrationsmanagement für Finanzdienstleister, in: *Sokolovsky, Z., Löschenkohl, S. (Hrsg.): Handbuch Industrialisierung der Finanzwirtschaft*, Gabler, Wiesbaden, 2005, S. 575-599.

Winter (2006)

Winter, R.: Ein Modell zur Visualisierung der Anwendungslandschaft als Grundlage der Informationssystem-Architekturplanung, in: *Schelp, J., Winter, R. (Hrsg.): Integrationsmanagement*, Springer, Berlin et al., 2006, S. 1-29.

Winter/Fischer (2006)

Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture, Arbeitsbericht, Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2006.

Winter/Schelp (2005)

Winter, R., Schelp, J.: Dienst-Orientierung im Business Engineering, in: *Hmd – Praxis der Wirtschaftsinformatik*, 42(2005)241, S. 45-54.

Winter/Strauch (2004)

Winter, R., Strauch, B.: Information Requirements Engineering for Data Warehouse Systems, in: *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC)*, Zypern, 2004, S. 1359-1365.

Winterthur (2005)

Winterthur: Geschäftsbericht 2005, Winterthur Group, Winterthur, 2005.

WKWI (1994)

WKWI: Profil der Wirtschaftsinformatik, in: *Wirtschaftsinformatik*, 34(1994)1, S. 80-81.

Wortmann (2005)

Wortmann, F.: Entwicklung einer Methode für die unternehmensweite Autorisierung, Dissertation der Universität St. Gallen, St. Gallen, 2005.

Yin (2002)

Yin, R. K.: Case Study Research – Design and Methods, 3. Aufl., Sage Publications, London, 2002.

Zachman (1987)

Zachman, J. A.: A Framework for Information Systems Architecture, in: *IBM Systems Journal*, 26(1987)3, S. 276-292.

Zachman/Sowa (1992)

Zachman, J. A., Sowa, J. F.: Extending and formalizing the framework for information systems architecture, in: *IBM Systems Journal*, 31(1992)3, S. 590-616.

Zarnekow et al. (2005)

Zarnekow, R., Brenner, W., Pilgram, U.: Integriertes Informationsmanagement – Strategien und Lösungen für das Management von IT-Dienstleistungen, Springer, Heidelberg, 2005.

Zarnekow et al. (2004)

Zarnekow, R., Hochstein, A., Brenner, W.: ITIL als de-facto Standard für IT Service Management, in: *Hirzel, M. (Hrsg.):* Prozessmanagement in der Praxis, HLP, Frankfurt am Main, 2004.

Zelewski (1999)

Zelewski, S.: Grundlagen, in: *Corsten, H., Reiss, M. (Hrsg.):* Betriebswirtschaftslehre, München/Wien, 1999, S. 1-125.

Zellner (2003)

Zellner, G.: Leistungsprozesse im Kundenbeziehungsmanagement, Dissertation der Universität St. Gallen, St. Gallen, 2003.

Lebenslauf

Persönliche Daten

Name, Vorname	Braun, Christian
Geboren am	10. Juni 1977
Geburtsort	München, Deutschland
Familienstand	ledig
Staatsangehörigkeit	Deutsch

Ausbildung und Wehrdienst

1983 – 1987	Grundschule in München
1987 – 1996	Erasmus-Grasser-Gymnasium München
1996 – 1997	Grundwehrdienst bei der Deutschen Bundeswehr
1997 – 2002	Studium der Informatik an der Ludwig-Maximilians-Universität München
2002 – 2005	Doktorandenstudium an der Universität St. Gallen, Institut für Wirtschaftsinformatik
2006 – 2007	Forschungsaufenthalt an der Santa Clara University (Kalifornien, USA) finanziert durch ein Nachwuchsstipendium des Schweizerischen Nationalfonds

Berufliche Tätigkeiten

1998 – 2002	Studentische Hilfskraft und Kursleiter in der Abteilung für Informations- und Kommunikationstechnologie der Ludwig-Maximilians-Universität München
2002 – 2006	Wissenschaftlicher Mitarbeiter am Institut für Wirtschaftsinformatik der Universität St. Gallen, Lehrstuhl Prof. Dr. Robert Winter