

Rahmenwerk zur integrativen Gestaltung von Services

MODELLGETRIEBENE SERVICEBESCHREIBUNG

Von der Wirtschaftswissenschaftlichen Fakultät
der Universität Leipzig
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades
DOCTOR RERUM POLITICARUM

DR. RER. POL.

vorgelegt

von Dipl.-Inform.Wirt. Christoph Augenstein
geboren am: 01. April 1980 in Pforzheim

Gutachter: Prof. Dr. A. Ludwig
Prof. Dr. B. Franczyk

Tag der Verleihung: 03.06.2016

Bibliographische Beschreibung

Augenstein, Christoph

Rahmenwerk zur integrativen Gestaltung von Dienstleistungen

Universität Leipzig, Dissertation

192 Seiten, 270 Referenzen, 31 Abbildungen, 9 Tabellen, 12 Definitionen, 3 Anlagen

Referat:

Die vorliegende Arbeit befasst sich mit der modellgetriebenen Servicebeschreibung, einem Ansatz zur integrativen Beschreibung bzw. Modellierung von Services. Ausgehend von der Prämisse, dass die Natur von Services nur schwer zu erfassen ist und eine Disziplinen-übergreifende Positiv-Definition nicht erreicht werden kann, soll mit dem Ansatz eine Synthese bisher vorhandener Modellierungsansätze erfolgen. Das Ziel liegt dabei nicht in einem erneuten Versuch, ein vollständiges Modell zur Beschreibung von Services zu entwickeln, sondern vielmehr vorhandene Perspektiven so zu integrieren, dass ein vollständiges Bild als eine Art Mosaik entsteht. Den Kern der Arbeit bildet das Service Modeling Framework. Als Rahmenwerk umfasst es Anforderungen und Restriktionen für die Arbeit mit unterschiedlichen Servicemodellen, definiert Methoden zur Integration und bietet Werkzeuge, mit deren Hilfe die darin enthaltenen Konzepte umgesetzt werden.

Ziel der Konstruktion des Rahmenwerks ist es Nutzer in die Lage zu versetzen mittels modellgetriebener Verfahren eine Zusammenführung von Modellen zu ermöglichen. Auf Basis einer fachlichen, nicht nur syntaktischen Beschreibung von Beziehungen zwischen Modellen und Modellelementen sollen Zusammenhänge modelliert werden, die einen Informationsaustausch zwischen Modellen realisieren. Dadurch werden Abhängigkeiten zwischen Modellen explizit formuliert oder aber die Entwicklung neuer Modelle auf Basis bereits bestehender Modelle vorangetrieben.

Der Beitrag dieser Arbeit besteht in der Erarbeitung der notwendigen Konzepte und in der Bereitstellung geeigneter Verfahren sowie Werkzeugen zur Umsetzung. Insbesondere stellt diese Arbeit einen Metamodell-basierten Ansatz zur Verfügung, mit dem Modelle über ein Domänen-neutrales Basismetamodell zueinander in Beziehung gesetzt werden können. Zugehörige Werkzeuge, wie Editoren, zeigen eine prototypische Umsetzbarkeit.

Vorwort

Je schneller man unterwegs ist, umso stärker dehnt sich die Zeit und umso stärker kontrahiert der Raum. Obwohl meine übliche Geschwindigkeit als relativ gemütlich beschrieben werden kann, ist die Zeit beim Schreiben dennoch nur langsam voran geschritten.

Es sind die letzten Sätze dieser Arbeit und ich bin froh darüber. Sie führen mir aber noch einmal den Entstehungsprozess dieser Arbeit vor Augen: Die ersten Schritte im LSB-Projekt, mit dem geforderten Ziel eines universellen Servicemodells für die Logistik, die darauffolgende Analyse einer ganzen Reihe von so genannten Universalmodellen, viele interessante Gespräche über Modelle und Generatoren und die daraus resultierende Idee alles miteinander zu verbinden.

Diese Schritte und die vielen anregenden Diskussionen waren nur mit den vielen Kollegen und Freunden möglich, mit denen ich in dieser Zeit zu tun hatte. Insbesondere möchte ich Bogdan Franczyk danken, der mir die Chance gegeben hat dies hier zu schreiben, nachdem es zunächst nicht nach einer Fortsetzung aussah sowie André Ludwig, der mich im LSB-Projekt aufgenommen hat und der schließlich als Betreuer die Richtung der Arbeit maßgeblich beeinflusst hat. In diesem Zusammenhang muss auch noch die Review-Fähigkeit von Jan Löhe genannt werden, dem ich, zu meinem Leidwesen, einmal ein Paper zum „drüberschauen“ gegeben habe und dessen Anmerkungen wohl so umfassend wie der Text selbst waren.

Das Ganze wäre jedoch nicht möglich gewesen ohne die fortwährende Unterstützung meiner Eltern Marion und Jürgen Augenstein. Angefangen mit dem recht spontanen Entschluss nach Leipzig zu ziehen und während der ganzen, darauffolgenden Zeit standen sie immer mit Rat und vor allem auch Tat zur Seite. Die „gelegentlich“ aufkommende Frage nach dem Ende, bzw. ob ich überhaupt schon angefangen hätte, hab ich jedes Mal als sportlich gemeinte Motivation gedeutet. Ich hoffe das war richtig so...

Übersicht

Bibliographische Beschreibung	II
Vorwort	III
Übersicht	IV
Inhaltsverzeichnis	V
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	X
Definitionen	XI
Abkürzungsverzeichnis	XII
1 Einleitung	1
Teil I Grundlagen	12
2 Modellierung	13
3 Services	30
4 Verwandte Ansätze	42
Teil II Service Modeling Framework	54
5 Modellgetriebene Servicebeschreibung	56
6 Multiperspektivische Servicemodellierung	67
7 Werkzeuggestützte Servicegestaltung	83
Teil III Anwendung und Validierung	93
8 Anwendungsdomäne Logistik	94
9 Validierung und Evaluation	105
10 Schlussbetrachtung und Diskussion	129
Literaturverzeichnis	XIV
Anhang	XXXV
Wissenschaftlicher Werdegang	LVII
Selbständigkeitserklärung	LVIII

Inhaltsverzeichnis

Bibliographische Beschreibung	II
Vorwort	III
Übersicht	IV
Inhaltsverzeichnis	V
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	X
Definitionen	XI
Abkürzungsverzeichnis	XII
1 Einleitung	1
1.1 Gegenstand und Motivation	1
1.2 Forschungsziel	5
1.3 Forschungsdesign	7
1.4 Aufbau der Arbeit	10
Teil I Grundlagen	12
2 Modellierung	13
2.1 Modellbegriff	15
2.1.1 Modellierungsparadigma	16
2.1.2 Modellarten.....	17
2.2 Modellierungsmethode	17
2.3 Metamodellierung	19
2.4 Modelltransformationen	21
2.5 Modellgetriebene Entwicklung	24
2.6 Zusammenfassung	26
Exkurs I: Referenzmodellierung.....	27
3 Services	30
3.1 Servicebegriff	32
3.1.1 Einordnung	33
3.1.2 Definition.....	34
3.2 Spezifische Servicemodellierungsansätze	38
4 Verwandte Ansätze	42

4.1	Vergleichsbasis.....	42
4.2	Ansätze.....	45
4.3	Zusammenfassung	52
Teil II Service Modeling Framework.....		54
5	Modellgetriebene Servicebeschreibung	56
5.1	Einführung	56
5.2	Anforderungen.....	59
5.2.1	Anforderungen aus der modellgetriebenen Entwicklung	59
5.2.2	Anforderungen aus dem Service Engineering	61
5.2.3	Anforderungen aus der Serviceorientierung.....	62
5.3	Restriktionen.....	64
5.3.1	Annahmen.....	65
5.3.2	Voraussetzungen.....	66
6	Multiperspektivische Servicemodellierung.....	67
6.1	Modellhierarchie.....	68
6.2	Modellkomposition	70
6.3	Modellbasis.....	71
6.3.1	Basismetamodell.....	72
6.3.2	Abgeleitete Modelle	76
6.4	Modellierungsmethode.....	78
7	Werkzeuggestützte Servicegestaltung	83
7.1	Anforderungen.....	83
7.2	Persistente Modellierung	85
7.3	Sprach-basierte Servicemodellierung.....	86
7.4	Modellintegration	88
Teil III Anwendung und Validierung.....		93
8	Anwendungsdomäne Logistik	94
8.1	Trends in der Logistik.....	96
8.2	Logistiksysteme und Supply Chain Management	98
8.3	Betreiber- und Geschäftsmodelle für Logistiksysteme	101

9	Validierung und Evaluation	105
9.1	Wissenschaftliche Validierung	105
9.2	Servicemodellierung für Logistiksysteme	106
9.2.1	Grundlagen	106
	Exkurs II: Servicemodellierung in LSEM	108
9.2.2	Fallbeispiel	112
9.3	Prototypische Validierung	115
9.3.1	Basisfunktionen (SMF-Core)	117
9.3.2	Modellzugriff (SMF-Model)	118
9.3.3	Service Repository (SMF-Explorer).....	119
9.3.4	Service Editor (SMF-Editor)	121
9.3.5	Modellintegration (SMF-Core)	123
9.3.6	Umsetzung des Fallbeispiels	125
9.4	Bewertung	126
10	Schlussbetrachtung und Diskussion	129
10.1	Zusammenfassung	129
10.2	Forschungsbeitrag	130
10.3	Schlussbetrachtung	132
	Literaturverzeichnis	XIV
	Anhang	XXXV
	Wissenschaftlicher Werdegang	LVII
	Selbständigkeitserklärung	LVIII

Abbildungsverzeichnis

Abbildung 1-1 - Angrenzende Forschungsgebiete	4
Abbildung 1-2: Schematische Kapitelübersicht	10
Abbildung 2-1: Modellierungskonzepte und -konstrukte einer Modellierungsmethode nach (Karagiannis und Kühn 2002)	18
Abbildung 2-2: Sprachbasiertes Metaisierungsprinzip nach (Strahringer 1998)	20
Abbildung 2-3: Modelltransformation - Elemente und Beziehungen	22
Abbildung 6-1: Modellhierarchie	69
Abbildung 6-2: Ableitung des Basismodells (angelehnt an (Frank, Klein et al. 1998))	72
Abbildung 6-3: Basismetamodell	74
Abbildung 6-4: Integrierter Entwicklungsansatz.....	77
Abbildung 6-5: Modellierungsprimitive.....	77
Abbildung 6-6: Modellkomponenten nach (Stahl, Völter et al. 2007).....	78
Abbildung 6-7: Modellierungsvorgehen	80
Abbildung 7-1: Beispiel eines modellierten Service	87
Abbildung 7-2: Modellabhängigkeiten.....	89
Abbildung 8-1: Marktsegmentierung in der Logistik (nach (Schmitt 2006, S. 34))	97
Abbildung 8-2: Typisierung von Logistikdienstleistern (nach (Arnold, Isermann et al. 2008, S. 979))	102
Abbildung 8-3: Spannungsfeld von Leistungskomplexität und Kundenorientierung (nach (Arnold, Isermann et al. 2008, S. 979)).....	103
Abbildung 9-1: Lebenszyklus eines Service	109
Abbildung 9-2: LSEM-Lebenszyklus.....	110
Abbildung 9-3: Zeitliche Grobplanung des 4PL	113
Abbildung 9-4: Simulationsmodell (Ausschnitt) des Beispielszenarios in ED.....	115
Abbildung 9-5: Vereinfachte Darstellung des Ecore-Metamodells (Ausschnitt).....	116
Abbildung 9-6: Package-Abhängigkeiten zwischen Komponenten.....	117

Abbildung 9-7: Quellcodegenerierung für Modellzugriff.....	119
Abbildung 9-8: Repository-Komponente	120
Abbildung 9-9: Kommunikation zwischen Komponenten.....	120
Abbildung 9-10: Editor-Komponente.....	121
Abbildung 9-11: Properties-Komponente	122
Abbildung 9-12: Datenstruktur zur Informationsmodellierung	125
Abbildung 9-13: Modellierung von Beziehungen zwischen Modellen.....	126

Tabellenverzeichnis

Tabelle 1-1: Forschungsziele im Überblick	9
Tabelle 2-1: Morphologischer Kasten zu Transformationseigenschaften.....	23
Tabelle 3-1: Serviceeigenschaften.....	37
Tabelle 3-2: Ansätze zur Servicemodellierung	40
Tabelle 4-1: Vergleichsmerkmale zur Abgrenzung.....	45
Tabelle 4-2: Eigenschaften verwandter Ansätze	51
Tabelle 7-1: Transformationseigenschaften im Rahmen des SMF	90
Tabelle 9-1: Kundenanforderung (links), Anforderung an Teildienstleistung (rechts).....	114
Tabelle 9-2: Bewertung SMF	127

Definitionen

Definition 1-1: Forschungsziel	5
Definition 1-2: Basishypothesen	5
Definition 1-3:Forschungsfrage 1.....	6
Definition 1-4:Forschungsfrage 2.....	6
Definition 1-5:Forschungsfrage 3.....	6
Definition 1-6:Forschungsfrage 4.....	7
Definition 2-1: Modellbegriff.....	16
Definition 2-2: Modellierungsmethode	19
Definition 2-3: Modelltransformation	22
Definition 3-1: Service	38
Definition 3-2: Servicemodell	41
Definition 9-1: LSEM.....	109

Abkürzungsverzeichnis

3PL	3 rd Party Logistics
4PL	4 th Party Logistics
AG	Aktiengesellschaft
API	Application Programming Interface
BPMN	Business Process Modeling and Notation
BPEL	Business Process Execution Language
CIM	Computational-independent model
CRUD	Create, Read, Update, Delete
CSM	Common Service Model
DSL	Domänenspezifische Sprache
EMF	Eclipse Modeling Framework
EMOF	Essential Meta Object Facility
EPK	ereignisgesteuerte Prozesskette
Ggf.	Gegebenenfalls
GmbH	Gesellschaft mit beschränkter Haftung
GoM	Grundsätze ordnungsmäßiger Modellierung
HGB	Handelsgesetzbuch
LLP	Lead Logistics Provider
LSEM	Logistik Service Engineering & Management
IuK	Informations- und Kommunikationstechnologien
IT	Informationstechnologie
JSR	Java Specification Request
M2M	Modell-zu-Modell
M2T	Modell-zu-Text
MDA	Model-Driven Architecture
MDE	Model-Driven Engineering
MDSD	Model-Driven Software Development
MOF	Meta Object Facility
o. B. d. A	ohne Beschränkung der Allgemeinheit
OCL	Object Constraint Language
OMG	Object Management Group
PIM	Platform-independent model
PSM	Platform-specific model

s.	siehe
SCM	Supply Chain Management
SCOR-Modell	Supply Chain Operations Reference-Modell
SE	Service Engineering
SMF	Service Modeling Framework
SOA	Serviceorientierte Architektur
TUL	Transport, Umschlag, Lager (logistische Grundfunktionen)
UFO	Unified Foundation Ontology
UML	Unified Modeling Language
URI	Universal Resource Identifier
USDL	Universal Service Description Language
u.U.	unter Umständen
vgl.	vergleiche
WI	Wirtschaftsinformatik
XMI	XML Metadata Interchange
XML	Extensible Markup Language
z. B.	zum Beispiel

1 Einleitung

1.1 Gegenstand und Motivation

“Service-oriented thinking is one of the fastest growing paradigms in IT, with relevance to accounting, finance, supply chain management and operations, strategy and marketing”

(Demirkan, Kauffman et al. 2008, S. 358)

– Und –

“Despite its importance, the level of professionalism in developing services cannot match the level of expertise in product development.” (Steen, Iacop et al. 2012, S. 2)

Das Angebot von Services, produktbegleitend oder direkt, nimmt neben dem eigentlichen Sachgut eine immer größere Bedeutung ein, die Entwicklung und Gestaltung von Services hat jedoch noch nicht die notwendige Reife. Diese Aussage fasst die beiden oben stehenden Zitate zusammen. Ein möglicher Grund für mangelnde Reife besteht darin, dass Servicekonzepte, die Entwicklung und Gestaltung von Services zwar schon seit langem diskutiert werden (vgl. (Shostack 1984)), je nach Forschungsdisziplin die Gewichtung von Konzepten und gestalterischen Merkmalen teilweise stark variiert. Zu den wichtigsten Vertretern dieser Disziplinen zählen die Dienstleistungsmodellierung¹ (Thomas und Nüttgens 2010) sowie das Service Engineering (SE, (Bullinger und Scheer 2006; Fähnrich und Opitz 2006)) oder die interdisziplinär ausgerichtete Science of Services (Chesbrough und Spohrer 2006; Spohrer, Maglio et al. 2007; Glushko 2008; Spohrer und Maglio 2008). Während die beiden ersten einen gestalterischen, ingenieurmäßigen Ansatz zur Entwicklung von Services beschreiben, liegt der Fokus der Service Science in einer übergreifenden Definition von Services, Service Systemen (Maglio, Srinivasan et al. 2006; Spohrer, Vargo et al. 2008) und der Interaktion zwischen den Beteiligten (Ramaswamy 2006; Payne, Storbacka et al. 2008; Ramaswamy 2008). Somit stehen sich konstruktions- und konzeptorientierte Disziplinen gegenüber, wenngleich jede Disziplin auch den gegensätzlichen Part nicht vollständig vernachlässigt.

Die Entwicklung von Services, insbesondere die strukturierte und ingenieurmäßige, wiegt nach Meinung des Autors schwerer im Vergleich zur konzeptuellen Erarbeitung der Servicedomäne. Eine Fokussierung auf die konzeptuelle Erfassung erscheint auch nach heutigem Stand nicht sinnvoll für diese Arbeit, da zu wichtigen Konzepten noch immer kein Konsens gefunden wurde (vgl. Kapitel 3). Im Zuge des verstärkten Angebots von hybriden

¹ Zur Unterscheidung der Begriffe Dienstleistung und Service s. 3.1

Produkten (Sachgut inkl. zugehöriger Services) oder gar eigenständigen Services erscheint ein pragmatischer Ansatz, zur Gestaltung und Entwicklung von Services besser geeignet als eine rein konzeptuelle Erarbeitung der Domäne.

Die Forschungsdisziplin, die die ingenieurmäßige Entwicklung am stärksten betont ist das SE (vgl. auch (Meyer und Böttcher 2012)), weshalb sich diese Arbeit die Konzepte und Vorgehensweisen des SE zu Nutze macht. Als besonders wichtig erachtet werden der Gebrauch von Modellen und die Einnahme mehrerer Perspektiven zur Gestaltung und Verwaltung von Services. Das SE nutzt diese, um mit einer definierten Vorgehensweise sukzessiv verschiedene Modelle zu entwickeln, die in ihrer Gesamtheit einen Service charakterisieren. Allerdings erscheint die Zahl der möglichen Perspektiven (Produkt, Prozess und Ressourcenmodelle (vgl. (Scheer, Grieble et al. 2006, S. 32)) oder Potenzial, Prozess, Ergebnis und Marktperspektive (vgl. (Meyer und Böttcher 2012, S. 9)) als nicht ausreichend bzw. die Festlegung einer geschlossenen Menge von Perspektiven als unvorteilhaft. Dies zeigt sich auch in der großen Bandbreite verfügbarer Servicemodelle, die abhängig von Anwendungsdomäne und konkretem Forschungsansatz entwickelt wurden (vgl. (Nardi, de Almeida Falbo et al. 2013) oder auch 3.2). Gerade die Anwendungsdomäne selbst spielt eine wichtige Rolle bei der Auswahl und Kombination von Perspektiven, respektive Modellen.

Diese Arbeit soll aber nicht einen erneuten Versuch darstellen Perspektiven und zugehörige Modelle zu entwickeln. Die bislang erarbeiteten Ansätze sollen nicht aufgrund der Wahl einer spezifischen Anwendungsdomäne oder Problems als ungenügend beschrieben werden. Dadurch würde sich lediglich die Zahl teils überlappender Modellierungsansätze erhöhen, ohne jedoch einen wesentlichen Beitrag zur Servicebeschreibung zu liefern. Stattdessen soll eine Möglichkeit geschaffen werden, relevante Perspektiven miteinander zu verbinden, wodurch zum einen existierende Ansätze wiederverwendet werden können und zum anderen ein zusätzlicher Nutzen aus der Wiederverwendung bereits bestehenden Wissens gezogen werden kann. Das Model-driven Engineering (MDE), also die modellgetriebene Entwicklung, kann dazu die notwendigen Mittel bereitstellen. Als Überbegriff steht sie für die Nutzung von Modellen zur Abstraktion und zur Darstellung von alternativen Perspektiven auf Sachverhalte (vgl. (Kent 2002, S. 289)). MDE eignet sich deshalb sehr gut zur Verbindung von Modellierungsansätzen in einem *integrativen Gestaltungsansatz*, da ein wesentliches Element die Transformation von Modellen darstellt. Die von der OMG standardisierte Model-driven Architecture (MDA, vgl. (Gruhn, Pieper et al. 2006)) ist ein Beispiel mit deren Hilfe Software durch sukzessive Verfeinerung aus bestehenden Modellen generiert werden soll (s. dazu auch die unterschiedlichen MDA-Ebenen CIM, PIM und

PSM). Bestehende Transformationsansätze, als Kernstück der MDE (vgl. (Calegari und Szasz 2013)), und deren Umsetzung haben mittlerweile einen Reifegrad erreicht, der es erlaubt, Modelltransformationen unter den gegebenen Restriktionen auf nahezu beliebige Modelle anzuwenden (vgl. (Amrani, Dingel et al. 2012)). Somit können einerseits die Vorgehensweisen der SE adaptiert und andererseits Perspektiven mit Hilfe der MDE integriert werden.

Den Rahmen für die Konzepte der Arbeit bildet die Serviceorientierung (SO). Obwohl nicht neu, wird sie im ersten Zitat zu Beginn als das wichtigste derzeit bestehende Paradigma genannt. Sie steht für eine stetige Veränderung bzw. Neuentwicklung von Services, wodurch die Notwendigkeit eines Gestaltungsansatzes nochmals betont wird. Die Orientierung, also die ständige und somit dynamische Ausrichtung an einer Zielgröße oder Maxime (Coenen 2001), führt zu einer ständigen Neuausrichtung oder Anpassung von Services gegenüber einem oder mehreren Nutzern. Aus betriebswirtschaftlicher Sicht lässt sich die SO daher folgerichtig auch als Spezialfall einer Kundenorientierung und damit als Spezialfall einer Marktorientierung definieren (Bruhn und Stauss 2010, S. 38). Unter diesem Gesichtspunkt ist es dann nur zwingend, dass durch Anwendung der SO Erwartungen, Anforderungen und Wünsche des Kunden im Vordergrund stehen und die Anpassung des eigenen Angebots die Maxime unternehmerischen Handelns darstellt.

Die Gestaltung von Services wird aus konzeptueller Sicht durch die SO unterstützt. Sie erlaubt es Aktivitäten und Ressourcen in wiederverwendbare Bausteine, den Services, zu kapseln (Heutschi 2007; Vogel 2009). Der besondere Vorteil besteht darin, dass Services mit ihrer, nicht zwingend nur informationstechnischen, Schnittstelle das „wie“ und „was“ eines Service sehr klar voneinander trennen, ja sogar das „wie“ in der Regel vom Nutzer verbergen. Der Service als Kern der SO stellt das generelle Objekt des Austauschprozesses dar. Sachgüter eines Unternehmens sind in diesem Zusammenhang nur als Vehikel bzw. Hilfsmittel der Interaktion zu verstehen (Bruhn und Stauss 2010, S. 446), eine Sichtweise, die der Service Dominant Logic (Lusch, Vargo et al. 2008; Vargo, Maglio et al. 2008) aus dem Bereich der Service Science sehr stark ähnelt. Der Kontext dieser Arbeit ist dementsprechend durch die Konzepte der drei genannten Gebiete gegeben und in Abbildung 1-1 grafisch dargestellt.

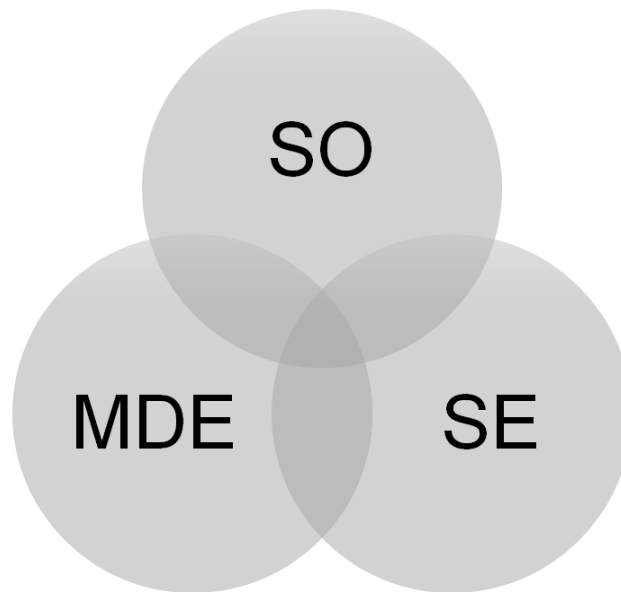


Abbildung 1-1 - Angrenzende Forschungsgebiete

Der Forschungsgegenstand dieser Arbeit ist die Verbindung von serviceorientierten Konzepten und modellgetriebenen Entwicklungsansätzen. Erklärtes Ziel dieser Arbeit ist es, ein Rahmenwerk zu entwickeln, mit dessen Hilfe unterschiedliche Modellierungsansätze bzw. deren Modelle zu einem einheitlichen Gesamtbild verbunden werden können. Dazu sollen Konzepte, Methoden und Werkzeuge geschaffen werden, die Modelle aus bestehenden Arbeiten des SE, der Dienstleistungsmodellierung und der Service Science innerhalb der für sie gedachten Kontexte wiederverwenden. Die Anwendung von modellgetriebenen Verfahren soll dazu beitragen, die daraus resultierenden, unterschiedlichen Perspektiven zu verknüpfen. Im Gegensatz zu modellgetriebenen Ansätzen aus der Softwareentwicklung steht jedoch nicht die Implementierung im Vordergrund, sondern eine integrative Beschreibung mittels unterschiedlicher Modelle, die jeweils einzelne Perspektiven eines Service repräsentieren. Die entscheidende Zielstellung der Arbeit ist es darzulegen, wie Informationen, die zur Servicedefinition und Serviceerbringung benötigt werden und die in unterschiedlichen Modellen enthalten sind, wiederverwendet werden können, ohne diese mehrmals aufnehmen zu müssen.

Daraus ergeben sich spezifische Herausforderungen für diese Arbeit. Es ist zu untersuchen, was im Sinne dieser Arbeit unter Services und unter Servicemodellen zu verstehen ist und wie sich unterschiedliche Perspektiven, durch Modelle repräsentiert, so miteinander verbinden lassen, dass ein konsistentes Gesamtbild entsteht. Die Anwendung und Evaluierung des Ansatzes wird am Beispiel der Logistik gezeigt, da die Arbeit im Kontext des Forschungsprojekts Logistik Service Engineering & Management (LSEM) entstand. Die Logistik ist jedoch nicht integraler Bestandteil des Ansatzes. Der nächste Abschnitt geht zu-

nächst tiefer in die Zielstellungen dieser Arbeit ein und beleuchtet die zentralen Fragestellungen.

1.2 Forschungsziel

Ziel dieser Arbeit ist es, Modelle zur Beschreibung von Services so miteinander zu verbinden, dass die darin beschriebenen Inhalte mehrfach wiederverwendet werden können. Somit wird der Dokumentationszweck der Modelle durch verstärkte Wiederverwendung gefördert. Darüber hinaus sollen bereits entwickelte Modellierungsansätze stärker operationalisiert, d.h. in mehreren Phasen der Servicegestaltung, -entwicklung und -bereitstellung eingebunden werden können. Die Herausforderung besteht darin, den Ansatz so zu konzipieren, dass diese unterschiedlichen Modellierungsansätze konsistent in ein Gesamtbild eines Service integriert werden können. Dazu notwendig ist die Entwicklung eines Rahmenwerkes, das sowohl die Einbindung, d.h. Nutzung, unterschiedlichster Modelle als auch die Wiederverwendung der darin enthaltenen Informationen ermöglicht. Der zugrundeliegende Gedanke lässt sich durch die folgende Fragestellung formulieren: Wie können Services ganzheitlich beschrieben und die modellierten Informationen wiederverwendet werden, wenn die zugrundeliegenden Modelle auf unterschiedlichen Metamodellen beruhen? Daraus leitet sich das Forschungsziel dieser Arbeit ab:

Definition 1-1: Forschungsziel

Ziel der Arbeit ist die Konstruktion eines Rahmenwerkes zur modellgetriebenen Servicebeschreibung, mittels dessen Informationen zwischen Modellen transparent für die Modellnutzer ausgetauscht werden können. Dies beinhaltet auch die Entwicklung von Modellen und Transformationsansätzen.

Mit diesem Forschungsziel sind drei Basishypothesen verbunden:

Definition 1-2: Basishypothesen

1. Eine vollständige Repräsentation eines Service kann nicht durch ein einzelnes Modell abgedeckt werden.
2. Wenn a priori nicht alle möglichen Modelle zur Beschreibung eines Service bekannt sind, ist ein auf der Integration von Metamodellen basierender Ansatz besser geeignet als die direkte Verwendung dieser Modelle.
3. Die Definition einer Abbildung von Modellelementen verschiedener Modelle auf Basis fachlicher Angaben ermöglicht die Verknüpfung von Informationen dieser Modelle und damit eine durchgängige Modellierung.

Diese Hypothesen bedingen die Nutzung serviceorientierter Prinzipien sowie die Verwendung und Einhaltung von Modellierungsansätzen zur Beschreibung von angebotenen Services. Sie können darüber hinaus nur dann gelten, falls eine Darstellung der eingesetzten Modellierungssprachen mittels eines einheitlichen Modellierungsansatzes möglich ist und falls für die Beschreibung von Services ein endlicher und hinreichend spezifischer Katalog von Eigenschaften angegeben werden kann. Die Operationalisierung dieser Hypothesen erfolgt anhand spezifischer Fragestellungen für diese Arbeit:

Definition 1-3:Forschungsfrage 1

Woraus setzt sich eine adäquate und ausreichende Servicebeschreibung zusammen?

Mit der Forschungsfrage 1 wird zunächst das Verständnis von Services in verschiedenen Disziplinen hinterfragt. Neben der reinen Definition eines Service sind darüber hinaus auch vorhandene Ansätze zur Umsetzung der Definitionen, d.h. zur Beschreibung von Services zu diskutieren und zu bewerten. Relevant sind hierbei insbesondere Eigenschaften von Services, die übergreifend Verwendung finden.

Definition 1-4:Forschungsfrage 2

Sind Konzepte des MDE auf eine konzeptionelle Modellierung von Services übertragbar?

Analog zum Bereich der Services thematisiert Forschungsfrage 2 den Bereich der Modellierung und der modellgetriebenen Entwicklung. Der Schwerpunkt dieser Frage liegt dabei weniger in der Auseinandersetzung mit dem Modellbegriff selbst. Vielmehr werden Ansätze der MDE im Allgemeinen sowie Einsatzgebiete im Bereich der Servicemodellierung untersucht.

Definition 1-5:Forschungsfrage 3

Wie können (Teil-)Modelle ineinander überführt werden, ohne dass im Voraus alle zu einer Transformation notwendigen Informationen vorhanden sind?

Forschungsfrage 3 ergibt sich aus den Überlegungen zu alternativen Transformationsansätzen und zur zugrundeliegenden Problemstellung, dass das Antizipieren aller möglichen Modellierungssprachen zur Beschreibung von Services weder möglich noch zielführend für einen breiten Einsatz ist. Diese Fragestellung ist zusammen mit der folgenden Forschungsfrage 4 wesentlich für den Kern dieser Arbeit.

Definition 1-6: Forschungsfrage 4

Wie müssen einzelne Komponenten eines Rahmenwerks ausgestaltet werden, um damit eine hinreichend präzise Servicebeschreibung spezifizieren zu können?

Schließlich thematisiert Forschungsfrage 4 den Rahmen für den Gesamtansatz in dieser Arbeit. Für versierte Nutzer sollen Artefakte und Konzepte geschaffen werden, die einen transparenten Umgang mit unterschiedlichen Modellen ermöglichen. Forschungsfrage 4 geht explizit auf die zu entwickelnden Artefakte bzgl. des Umfangs als auch der Nutzbarkeit ein.

1.3 Forschungsdesign

Die vorliegende Arbeit widmet sich Fragestellungen der Servicemodellierung, bzw. der Konzeption von Servicemodellen und auch deren Operationalisierung zum Zwecke der Informationsmodellierung und -wiederverwendung. Sie ist stark gestaltungsorientiert und weniger theoriebildend geprägt und lässt sich nach (Becker, Krcmar et al. 2009, S. 1) dem Forschungsgebiet der Wirtschaftsinformatik (WI) eindeutig zuordnen. Sie ist epistemologisch der konstruktivistischen Forschung zuzuschreiben und verfolgt insbesondere die Ideen des radikalen Konstruktivismus (von Glasersfeld 1997; von Foerster und Glasersfeld 2007), nach dem Realität nicht durch eine objektive Wahrheit intersubjektiv beschrieben werden kann. Realität ist vielmehr Ergebnis einer subjektiven Wahrnehmung, die insbesondere durch die persönlichen Erfahrungen (das Gedächtnis) eines Einzelnen beeinflusst ist. Dies deckt sich mit dem teils doch sehr unterschiedlichen Verständnis von Services, repräsentiert durch die Vielzahl an Servicemodellen mit unterschiedlichem Abstraktionsgrad und Detailniveau. Wenngleich Modelle aus unterschiedlichen Disziplinen unterschiedliche Fragestellungen beantworten und damit objektiv gesehen unterschiedlich sein müssen, zeigt sich doch auch innerhalb einer Disziplin, dass es häufig nur einen kleinen Konsens bezüglich der wesentlichen Eigenschaften und Attribute von Services gibt (s. 3.1 bzw. 3.2)

Die WI als gestaltungs- bzw. konstruktionsorientierte Forschungsdisziplin will Artefakte entwickeln (Zelewski 2006, S. 3) und strebt durch Schaffen von IT-Lösungen nach Erkenntnisgewinn (Wilde und Hess 2006, S. 3). Darüber hinaus versteht sich die WI „als Wissenschaft mit einer methodenpluralistischen Erkenntnisstrategie“ (Wilde und Hess 2007) bzw. (Becker und Pfeiffer 2006, S. 15), die den wissenschaftlichen Erkenntnisgewinn durch den Einsatz, auch mehrerer, unterschiedlicher wissenschaftlicher Methoden vorantreibt. Das Memorandum zur gestaltungsorientierten WI (Österle, Winter et al. 2010)

unterstreicht diese Merkmale und stellt zugleich die Grundlage für den in dieser Arbeit verwendeten Erkenntnisprozess dar. Demnach ist ein solcher Prozess iterativ durchzuführen und besteht aus den folgenden Phasen:

1. **Analyse:** Identifizierung der Problemstellung und Formulierung der Forschungsziele in Form von Forschungsfragen und Aufzeigen von „Lücken“. Für die Arbeit wurden in der Analysephase eine Literaturrecherche in relevanten Fachdomänen (z. B. Service Engineering, Service Science oder Modellierung) durchgeführt sowie im praktischen Umfeld einschlägige Modellierungssuiten untersucht und schließlich aus dem Bereich der Logistik relevante Prozesse betrachtet. Letzteres diente dem Zweck den Umfang und die Variabilität der dort vorkommenden Services näher zu untersuchen und damit eine Ausgangslage für den Entwurf zu schaffen.
2. **Entwurf:** Herleitung von Artefakten nach anerkannten Methoden, die für die Umsetzung der Gesamtlösung notwendig sind. Für die Arbeit wurde ein Prototyping-Ansatz gewählt, um mittels schnell verfügbarer und lauffähiger Versionen von Editoren und Hilfswerkzeugen das Konzept eines generischen Servicemodells zu entwickeln. Außerdem wurde der entstandene Prototyp zu Evaluationszwecken angewendet (vgl. (March und Smith 1995)) und somit ein Proof-of-Concept für die gesamte Arbeit durchgeführt.
3. **Evaluation:** Überprüfung der entwickelten Artefakte, ob sie den definierten Zielen genügen und ob deren Entwicklung den verwendeten wissenschaftlichen Methoden genügt. Die Evaluation fand auf verschiedenen Ebenen statt. Zum einen durch wissenschaftliche Begutachtung in einschlägigen Konferenzen und Journalen zum anderen durch praktische Erprobung mittels des entwickelten Prototyps für verschiedene Beispielszenarien (Laborexperiment).
4. **Diffusion:** Verbreitung der gewonnenen Ergebnisse in Form von Artefakten sowie in Form von wissenschaftlichen Beiträgen. Die Verbreitung der gewonnenen Erkenntnisse und Konzepte fand in einschlägigen wissenschaftlichen Konferenzen und Journalen, sowie in Form von Vorträgen und praxisnahen Versuchen statt.

Der gesamte Prozess wurde insgesamt zweimal durchlaufen, um eine Verbesserung der im ersten Schritt entstandenen Konzepte zu erreichen. Die Entwicklung des Prototyps fand ebenfalls in einem iterativen Prozess innerhalb der Entwurfsphase statt, um sukzessive notwendige Eigenschaften ergänzen und die Konzepte der Servicemodellierung erproben zu können. Dazu wurden Hypothesen aus der Literatur abgeleitet, mit deren Hilfe Anforderungen an den Prototyp gestellt wurden. Im Anschluss daran dient der Prototyp wiederum zur Überprüfung der Basishypothesen (s. Definition 1-2) und somit zur Überprüfung der

gewonnenen Erkenntnisse und Ergebnisse. Tabelle 1-1 gibt einen Überblick über die gestalterischen und methodischen Ziele der Arbeit im Überblick.

	Erkenntnisziel	Gestaltungsziel
Methodischer Auftrag	Generischer Servicemodellierungsansatz, Ableiten eines Rahmenwerks zur Realisierung einer Servicemodellierung mittels modellgetriebener Verfahren	
Inhaltlich-funktionaler Auftrag	Umsetzung des Rahmenwerks mittels intuitiv nutzbarer, modellbasierter Prototypen	Anwendung des Rahmenwerks in der Logistik in KMU Netzwerken

Tabelle 1-1: Forschungsziele im Überblick

Die Tabelle verdeutlicht, dass mit dieser Arbeit vor allem ein inhaltlich-funktionaler Auftrag verbunden ist. Ein methodischer Auftrag ist mit dieser Arbeit nur insofern verknüpft, als dass sich die Gewinnung der Erkenntnisziele der Arbeit an wissenschaftlichen Methoden der WI orientiert. Unter Verwendung des Forschungsrahmens der gestaltungsorientierten Wirtschaftsinformatik (Österle, Winter et al. 2010) ergeben sich für die Arbeit sowohl Modelle als auch Prototypen als Ergebnistypen, die in einem Rahmenwerk (Erscheinungsform) gebündelt werden. Dabei werden die Artefakte und Erscheinungsformen an den folgenden Prinzipien ausgerichtet:

1. Abstraktion (Anwendung auf eine Klasse von Problemen):

Die modellgetriebene Servicebeschreibung wird auf Servicemodellen angewandt. Es existieren jedoch zunächst keine Gründe für eine Einschränkung auf eine spezifische Domäne oder für die Anwendung der modellgetriebenen Servicebeschreibung auf Nicht-Servicemodellen, solange die grundlegenden Anforderungen und Einschränkungen unangetastet bleiben.

2. Originalität (innovativer Beitrag zum Wissensstand):

Nach aktuellem Kenntnisstand ist der mit dieser Arbeit verfolgte Ansatz einzigartig in der Manipulation von Servicemodellen und der Verwaltung von Services.

3. Begründung (Nachvollziehbarkeit und Validierbarkeit der Lösung):

Die Herausarbeitung der Konzepte und Artefakte erfolgt entlang eines stringenten, wissenschaftlichen Prozesses mit definierten Ergebnissen. Dadurch wird sowohl die Nachvollziehbarkeit als auch die Validierbarkeit ermöglicht.

4. Nutzen (Nutzen stiften für Anspruchsgruppen):

Anspruchsgruppen werden für die Arbeit definiert. Anwendern der modellgetriebenen Servicebeschreibung ermöglichen die Artefakte der Arbeit einen einfacheren und transparenten Umgang mit Servicemodellen und darin enthaltenen Informationen.

1.4 Aufbau der Arbeit

Der Aufbau der Arbeit gliedert sich in drei Teile. Teil I dient der Erarbeitung notwendiger Grundlagen und Begriffe aus den angrenzenden Gebieten der Modellierung und der Services. Darüber hinaus werden vergleichbare, bereits bestehende Ansätze, die in ihrer Komplexität und in ihrem Umfang der modellgetriebenen Servicebeschreibung entsprechen, vorgestellt und bewertet. Dabei werden in jedem Grundlagenkapitel entweder am Ende eines Kapitels oder am Ende eines Abschnitts noch einmal diejenigen Teile hervorgehoben, die für die vorliegende Arbeit notwendig sind. Teil II beschreibt das hier entwickelte Rahmenwerk, die notwendigen Konzepte sowie Artefakte und setzt diese zueinander in Beziehung.

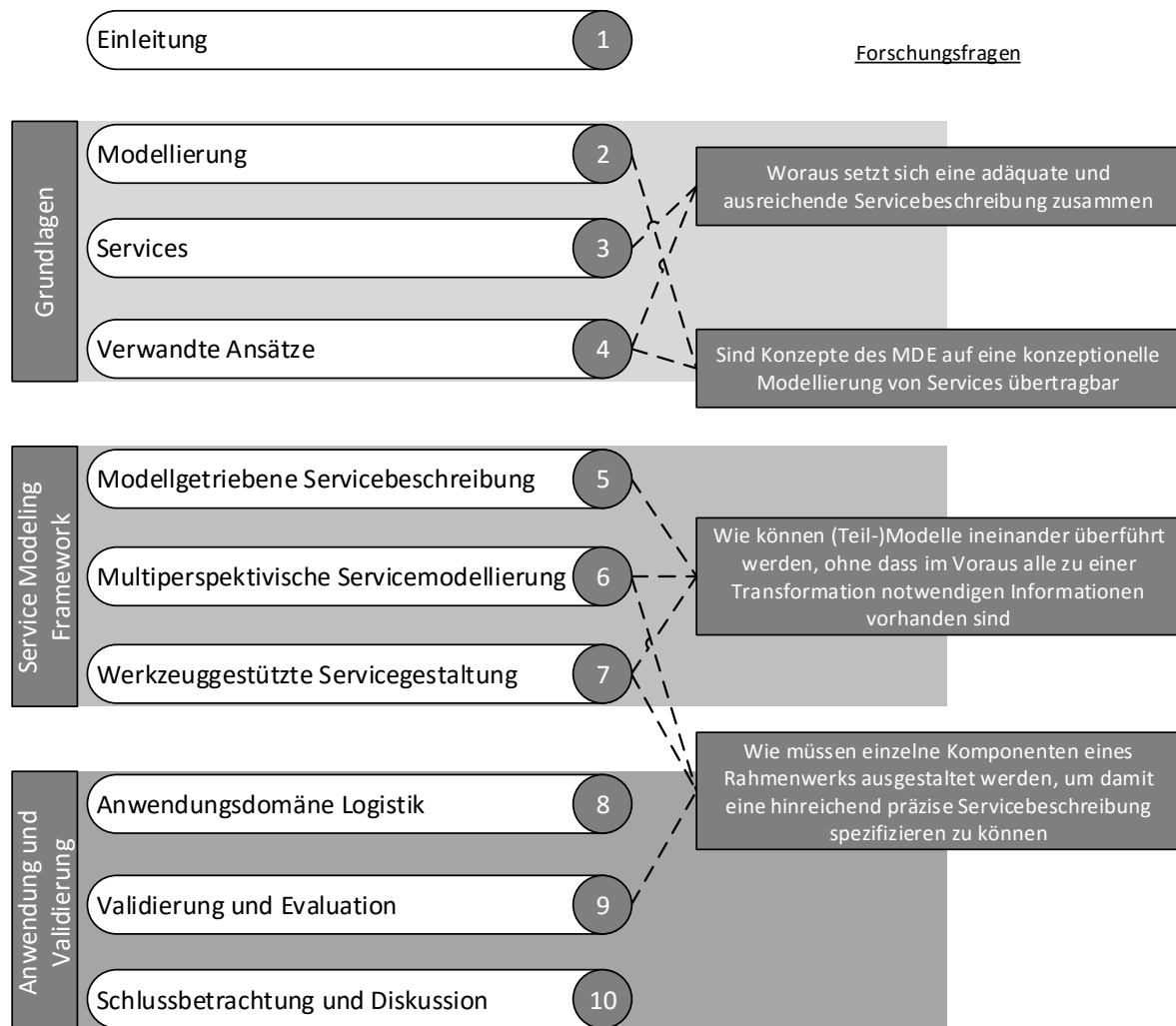


Abbildung 1-2: Schematische Kapitelübersicht

Teil III beginnt mit einer Beschreibung der Anwendungsdomäne, mittels derer die modellgetriebene Servicebeschreibung exemplarisch umgesetzt werden soll. Außerdem enthält dieser Teil die eigentliche Validierung sowie abschließende Bemerkungen und die Ge-

samtwürdigung der Arbeit. Abbildung 1-2 bietet zu den einzelnen Kapiteln einen Überblick und zeigt den Zusammenhang zwischen einzelnen Bestandteilen der Arbeit und den zugrundeliegenden Forschungsfragen auf. Es wird ersichtlich, dass der Aufbau der Arbeit insofern den Forschungsfragen entspricht, als dass Erkenntnisse bereits durch die Ausarbeitung der Grundlagen gewonnen werden. Die Forschungsfragen 3 (Definition 1-5) und 4 (Definition 1-6) werden in den folgenden Teilen behandelt und unterstützen explizit die Validierung der Artefakte.

Teil I

Grundlagen

2 Modellierung

Modellierung als Oberbegriff bildet den Kontext für die Arbeit. Vor der eigentlichen Auseinandersetzung mit Modellen bzw. mit dem Modellbegriff soll zunächst die Frage stehen, wozu Modelle verwendet werden. Unter dem Begriff der Modellierung wird dabei die Aufnahme oder Konstruktion eines realen Problems mit Hilfe konzeptueller Modellierungssprachen verstanden, also deren Nutzung zur Problemlösung (vgl. (Ferstl und Sinz 2001, S. 85), (Schütte 1998, S. 60f), (Herrmann 1992, S. 196ff)). Dabei besteht bereits zu Anfang oder im Zuge der Modellierung ein gedankliches, implizites Modell, welches durch die Modellierung in ein Explizites umgewandelt wird (vgl. (Gehlert 2007, S. 47), (Pfeiffer 2008, S. 50)). Ein (konzeptuelles) Modell stellt in diesem Kontext eine Abstraktion der Realität dar (Becker, Rosemann et al. 1995). Mit der Hilfe von Modellen ist es möglich, komplexe Zusammenhänge, unsere Umwelt oder Systeme zu verstehen, zu verwalten oder auch zu steuern. Modelle sind, wie auch Ontologien, zur Konzeptbildung geeignet ((Fensel 2003), (Ushold und Gruninger 2004)), allerdings mit der Einschränkung, dass Modelle für einen spezifischeren Zweck eingesetzt und in der Regel nicht allgemeingültig oder durch eine breite Basis definiert werden (Hesse und Krzensk 2004). Somit tragen sie zur Entwicklung eines Problem- oder Lösungsraums bei. Häufig werden Modelle auch zur Dokumentation von Wissen eingesetzt, bspw. in Form eines EPK- oder BPMN-Prozessmodells. Dieses Wissen veraltet jedoch in vielen Fällen, da die Aufnahme der Modelle am Anfang eines Projekts erfolgt, jedoch spätere Änderungen nicht mehr in das Modell eingepflegt werden, so dass ggf. der modellierte Wissensstand nicht mehr dem Aktuellen entspricht. Ebenfalls problematisch ist, dass häufig inadäquate Modelle zur Erschließung verwendet werden, z. B. weil geeignetere Modelle nicht bekannt oder nicht verfügbar sind. Dies führt häufig zu einem fehlerhaften Modellierungsprozess und zu inkonsistenten Modellen. Dementsprechend sind zahlreiche Herausforderungen in der Arbeit mit Modellen zu meistern. Modelle müssen konzeptuell verstanden werden, d.h. deren Zweck muss bekannt und die im Modell abgebildeten Zusammenhänge erschließbar sein. (Becker, Rosemann et al. 1995; Schütte 1997; Schütte 1998) haben zu diesem Zweck Eigenschaften und Kriterien unter dem Begriff „Grundsätze ordnungsmäßiger Modellierung“ (GoM) zur Modellbewertung eingeführt, die eine Analogie zu den „Grundsätzen ordnungsmäßiger Buchführung“, einer Reihe von kodifizierten und nicht-kodifizierten Regeln zur Fehlervermeidung bei der Buchführung und Bilanzierung, darstellen. Das Ziel der GoM besteht primär in einer hohen Qualität der zu erstellenden Modelle, so dass die mit Modellen zu erreichende, komplexitätsreduzierende Abbildung der Realität auf einem hohen Niveau

ermöglicht wird. Die GoM sind dabei als allgemeine Regeln zu verstehen, die unabhängig vom Modellzweck, der zugrundeliegenden Modellierungssprache oder des abzubildenden Szenarios sind. Sie beinhalten:

- *Konstruktionsadäquanz*: Nachvollziehbarkeit der Modellkonstruktion. Untersucht wird die Geeignetheit eines verwendeten Modells bzgl. des zugrundeliegenden Problems und der zugehörigen Entwurfsentscheidungen bei der Modellierung.
- *Sprachadäquanz*: Angemessenheit einer Sprache zur Beschreibung einer Problemsituation, sowie deren korrekte Verwendung der Syntax, ableitbar aus dem zugehörigen Metamodell.
- *Wirtschaftlichkeit*: Sparsamer Ressourceneinsatz auch im Sinne einer Informationsmodellierung, d.h. der Aufwand zur Modellierung eines Sachverhalts muss in Relation zum erwarteten Nutzen sein.
- *Systematischer Aufbau*: Notwendigkeit zur Einführung unterschiedlicher Sichten zur Komplexitätsreduktion. Über ein Metamodell muss der Zusammenhang zwischen den Sichten abbildbar bzw. umsetzbar sein.
- *Klarheit*: Anschaulichkeit eines Modells. Einfache Darstellung zur Steigerung des (gemeinsamen) Verständnisses z. B. bei unterschiedlichen Nutzergruppen.
- *Vergleichbarkeit*: Nebeneinander existierende Modelle, die denselben Sachverhalt repräsentieren, machen eine Vergleichbarkeit notwendig, um bspw. Modellinhalte zusammenfassen oder integrieren zu können und so eine vollständigere Sicht zu erhalten.

Die Allgemeingültigkeit der Forderungen macht eine universelle Einsetzbarkeit möglich. Sie sollen im Folgenden auch für die hier zu entwickelnden Modelle sowie als Grundvoraussetzung für die Modellierung mit Hilfe der modellgetriebenen Servicebeschreibung gelten, d.h. die Anwendung der Servicebeschreibung in einer bestimmten Domäne bedingt die Einhaltung der GoM. Zusammenfassend lässt sich außerdem noch ein Prinzip der Sparsamkeit ableiten:

Nach den GoM sollte ein Modell so viele Informationen wie notwendig, aber so wenige wie möglich beinhalten.

Dementsprechend müssen komplexe Szenarien zwangsläufig mit Hilfe mehrerer, spezifische Sichten repräsentierende Modelle abgebildet werden. Jedes Modell, d.h. jede Sicht, ist so zu wählen, dass das zugehörige Metamodell aufgrund der Syntax und Semantik mit möglichst wenigen Konstrukten eine adäquate Abbildung bzw. Komplexitätsreduktion erlaubt. Der nachfolgende Abschnitt setzt sich nun detailliert mit dem Modellbegriff ausei-

inander, um wesentliche Eigenschaften eines Modells zu definieren und damit den Grundstein für die Konzeption des Rahmenwerks und die Auswahl geeigneter Modelle zu legen.

2.1 Modellbegriff

„Ein Modell ist ein immaterielles Abbild der Realwelt für Zwecke eines Subjekts“ (Becker, Rosemann et al. 1995, S. 435).

Die Aussage des Satzes fasst die wesentlichen Eigenschaften von Modellen sehr prägnant zusammen. Ein Modell ist nicht real, nicht fassbar, sondern spiegelt die Realität in einem bestimmten Sinne, vollständig oder teilweise wider (vgl. (Stachowiak 1973, S. 129ff), (Domschke, Scholl et al. 2003, S. 30), (Fischer, Dangelmaier et al. 2012, S. 281)). (Stachowiak 1973, S. 129ff) benennt darüber hinaus Verkürzung und Pragmatismus als wesentlichen Eigenschaften von Modellen. Erstere führt dazu, dass Modelle nicht alle Attribute der Realität beinhalten und somit eine Vereinfachung darstellen und wegen Letzterer ersetzen Modelle ganz oder teilweise den abgebildeten Realitätsausschnitt zur Komplexitätsreduktion (Ersetzungsfunktion). Ein Modell ist zunächst an ein Subjekt (Perspektive) gebunden, das mit dem Modell seine Sicht auf die Realität abbildet und dies zu einem bestimmten Zweck (s. (Kent 2002, S. 289f), (Becker, Probandt et al. 2012, S. 1)). Modelle sind somit primär Hilfsmittel zur Erklärung und Gestaltung und dienen zur Gewinnung von Zusammenhängen oder Sachverhalten (Adam 1997, S. 44). Außerdem sind sie wichtige Instrumente zur Manipulation komplexer Systeme, indem sie Strukturen, Verhalten und / oder Funktionen vereinfacht nachbilden (Fischer, Dangelmaier et al. 2012, S. 281). Es sind unterschiedliche Ausprägungen für Modelle möglich. Sie können bspw. als Skizze, als UML-Diagramm oder in textueller Form vorliegen. Unterschieden werden kann auch in Menschen- oder maschinenlesbare Modelle (Petrasch und Meimberg 2006).

Im Bereich der WI kommt Modellen einen besondere Bedeutung zu (Ferstl und Sinz 2001, S. 119). Sie sind zentraler Gegenstand und Werkzeug zur Entwicklung von Informationssystemen, dem eigentlichen Ziel der WI und damit auch wesentlicher Erkenntnisgegenstand (vgl. (Schütte 1998, S. 63), (Becker, Rosemann et al. 1995, S. 435), (Gehlert 2007, S. 56)). Für die Arbeit mit Modellen ist es erforderlich, die Bestimmung oder Klassifikation eines Modells zu erarbeiten, also den Zweck für den es zu entwickeln ist. Sind alle Gründe und Einflussfaktoren für ein Modell bekannt, so konkretisieren sich Methoden und Vorgehensweisen zur Erstellung (Fischer, Dangelmaier et al. 2012, S. 284ff). Den zugehörigen Modellraum (Kent 2002, S. 289f), also die Gesamtheit der zu erfassenden Merkmale der Realität oder Systems, erhält man dann dadurch, dass man unterschiedliche Perspektiven Kategorien zuordnet, die einen n-dimensionalen Raum aufspannen und so die abzubilden-

de Realität multiperspektivisch erfasst. Daraus ergibt sich eine spezifische Definition eines Modells durch die Nennung der betroffenen Dimension(en). Allerdings ist die Auswahl der Kategorien und Perspektiven abhängig vom Modellierer. Der Grund dafür liegt in dem oben beschriebenen Zusammenhang zwischen Realität und deren subjektiver Wahrnehmung durch den Modellierer und damit in dessen erkenntnistheoretischen Position (vgl. (Becker, Holten et al. 2003, S. 8), (Becker, Niehaves et al. 2004, S. 4), (Hesse und von Braun 2001, S. 277)). Daher wird für diese Arbeit aufgrund des konstruktivistischen Modellbegriffs der WI eine subjektgebundene, erfahrbare Realität nochmals betont. Somit ergibt sich die folgende Definition des Modellbegriffs:

Definition 2-1: Modellbegriff

Ein Modell ist ein Subjekt- und zweckbezogenes, konstruiertes Abbild der wahrgenommenen Realität oder eines Systems. Mittels Vereinfachung und Analogiebildung werden existierende Zusammenhänge prägnant abgebildet.

2.1.1 Modellierungsparadigma

Die Modellierung eines bestimmten Sachverhalts muss einer bestimmten Denkweise, einem Paradigma, folgen. Damit ist gemeint, dass resultierende Modelle einer spezifischen Anschauungsweise unterliegen und dass bei ihrer Konstruktion spezifische Annahmen und Voraussetzungen gelten müssen. Nur dann können Modelle ihrem Zweck optimal entsprechen. Das verwendete Modellierungsparadigma engt zugleich die Zahl der möglichen Modellarten ein. Beispiele für Paradigmen und entsprechende Modelle sind:

- statistische Modelle
- systemdynamische Modelle
- zeitdiskrete Modelle
- agentenbasierte Modelle
- deskriptive Modelle
- entscheidungsunterstützende Modelle
- präskriptive Modelle

Für diese Arbeit sind insbesondere die letztgenannten Paradigmen und zugehörige Modelle relevant. Letztlich werden in der Servicemodellierung eine Reihe unterschiedlicher Paradigmen und Modellarten eingesetzt, da je nach Zweck beschreibende (Zustandsbeschreibung), dynamische (Simulationen) oder auch statistische (Einsatzanalysen) Modelle verwendet werden.

2.1.2 Modellarten

Eng verknüpft mit einem Paradigma sind auch Modellarten. Die Art des verwendeten Modells ist in erster Linie mit dem Zweck der Erstellung oder mit der Ausgestaltung verbunden. Letztere Interpretation greifen (Fischer, Dangelmaier et al. 2012, S. 284ff) auf, die Modelle in gedankliche Modelle, verbale Systembeschreibungen, formale Modelle oder operable Modelle untergliedern. (Domschke, Scholl et al. 2003, S. 30ff) unterscheiden zunächst in Beschreibungsmodelle (Darstellung), Erklärungs- /Kausalmodelle (Ursache-Wirkung), Prognosemodelle (Vorhersage von Entwicklung oder Wirkung) und in Entscheidungs- / Optimierungsmodelle (Bewertung und Auswahl von Handlungsmöglichkeiten bzw. Simulationsmodelle (spezielle Prognosemodelle für komplexe Systeme). Unabhängig davon lassen sich Modelle unterscheiden in quantitative und qualitative Modelle. Erstere versetzen einen Anwender in die Lage mittels kardinal messbarer Informationen Probleme zu analysieren und Lösungen zu ermitteln, wie bspw. bei vielen Modellen des Operations Research. Letztere Modelle werden auf Basis ordinaler oder nominaler Informationen erstellt und bestehen größtenteils aus verbalen Beschreibungen, subjektiven Einschätzungen und stellen grundlegende Tendenzen oder allgemeine Zusammenhänge dar. Eine weitere, wichtige Unterscheidung der Modellarten, insbesondere für die später beschriebenen Modelltransformationen (s. 0) betrifft die Frage der Abstraktion. Normalerweise wird zwischen deterministischen (alle zugrunde liegenden Informationen sind mit Sicherheit bekannt), statistischen (Abstraktion von der zeitlichen Entwicklung eines Systems), dynamischen (wie statistisch, aber mit Einbezug der zeitlichen Entwicklung), Total- (abzubildendes reales System wird in seiner Gesamtheit modelliert) bzw. Partialmodellen (Betrachtung eines bestimmten Aspekts) unterschieden. Für die Arbeit mit Servicemodellen spielen dabei vor allem die Partialmodelle eine sehr wichtige Rolle, da mit spezifischen Modellen einzelne Perspektiven auf einen Service beschrieben werden sollen. Totalmodelle dagegen, werden aufgrund der Eigenschaften von Services (s. Kapitel 3) nicht eingesetzt.

2.2 Modellierungsmethode

Wesentlich für die systematische Entwicklung eines Modells ist die Nutzung einer expliziten Modellierungsmethode, mit deren Hilfe ein Subjekt einen Aufgabentyp über eine Analogiebildung mittels Modell löst (Fischer, Dangelmaier et al. 2012, S. 287). Unter einer Methode ist dabei eine Kombination aus Aufgabentyp und Regeln zur deren Bewältigung zu verstehen (Becker, Knackstedt et al. 2001, S. 5). Eine Modellierungsmethode unterstützt die Konstruktion und Pflege einer Klasse von Modellen, die für einen bestimmten Aufga-

bentypus geeignet sind und beinhaltet in der Regel mindestens eine Modellierungssprache, mit deren Hilfe die Modelle erstellt werden. Genauer, eine Modellierungsmethode wird durch die drei Elemente Modellierungssprache, Modellierungstechnik und eine Handlungsanleitung spezifiziert (Karagiannis und Kühn 2002), wobei die Modellierungstechnik Sprache und Handlungsanleitung subsumiert (vgl. auch (Strahinger 1996, S. 91)).

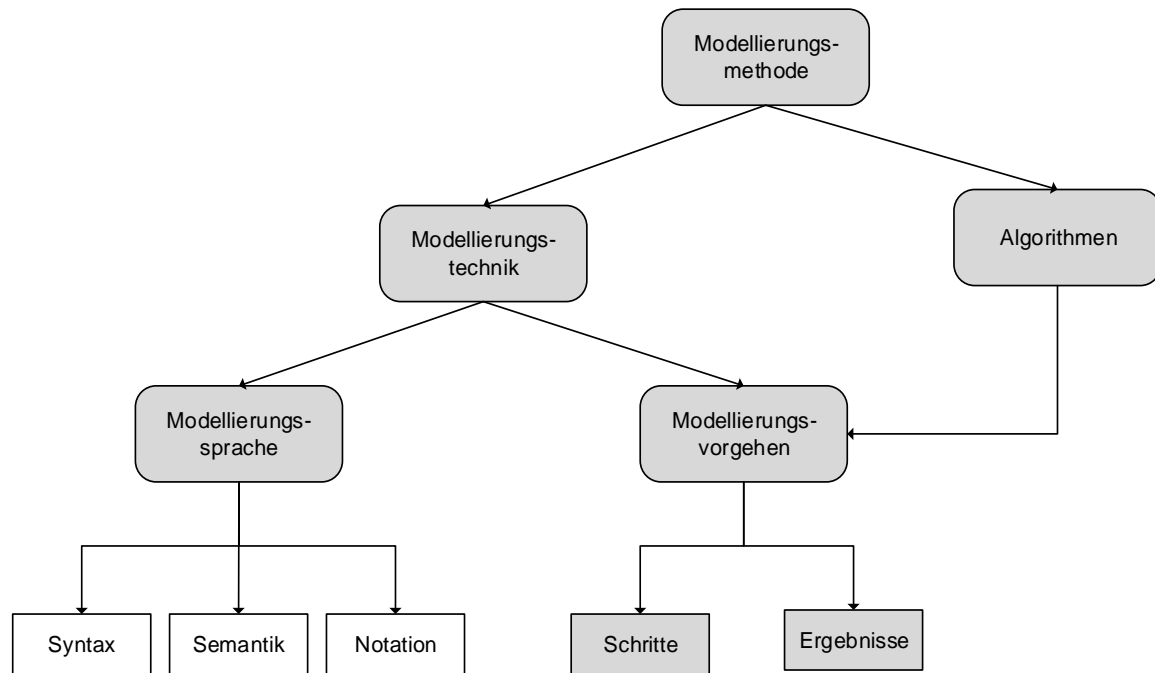


Abbildung 2-1: Modellierungskonzepte und -konstrukte einer Modellierungsmethode nach (Karagiannis und Kühn 2002)

Wesentlich für die Beschreibung, von (Verhoef, ter Hofstede et al. 1991, S. 502ff) auch „way of modelling“ bezeichnet, ist die Modellierungssprache, da sie die Elemente bereitstellt, mittels derer ein Modell erstellt wird. Eine Handlungsanleitung baut lediglich darauf auf und beschreibt wie die Modellelemente verwendet werden sollen (way of working). Die sprachlichen Elemente werden dabei auch häufig als Konstrukte zur Modellierung (Fischer, Dangelmaier et al. 2012, S. 288) und die Gesamtheit der Konstrukte als Vokabular der Sprache bezeichnet (Patig 2007, S. 26). Häufig sind Modellierungssprachen künstliche Sprachen. Im Kern besteht eine Sprache aus einer Syntax (Regeln zum formalen Aufbau der Sprache), einer Semantik (definierte, inhaltliche Bedeutung der Sätze und Wörter der Sprache) sowie mindestens einer zugehörigen, sprachlichen oder grafischen Notation, die den Regeln der Syntax und Semantik gehorcht und die die eigentliche Repräsentation darstellt (vgl. Abbildung 2-1). Präziser noch lässt sich die Syntax in abstrakte Syntax, der Menge von Symbolen und den zugehörigen Anordnungsregeln, und konkrete Syntax, die Darstellung der Symbole oder Notation, unterscheiden (Frank und van Laak 2003, S. 20).

Die Zuordnung einer Bedeutung zu den Symbolen einer Sprache sowie die Definition zulässiger, zusammengesetzter Konstrukte aus den Symbolen wird als Semantik bezeichnet (Frank und van Laak 2003, S. 20).

Die Handlungsanleitung, in der Abbildung 2-1 Modellierungsvorgehen genannt, definiert einen Prozess, wie die Sprachbausteine genutzt werden, um ein konkretes Modell zu erstellen. Demnach spezifiziert sie elementare Aktivitäten und Aufgaben und eine Reihenfolge dieser (vgl. (Verhoef, ter Hofstede et al. 1991, S. 502ff)). Das Ziel der Handlungsanleitung besteht in einer syntaktisch und semantisch korrekten Beschreibung des zugrundeliegenden Sachverhalts in einem Modell mit den verfügbaren Mitteln der Modellierungssprache (Becker, Knackstedt et al. 2001, S. 9). Auf Basis dieser Vorgaben wird in Kapitel 6.4 eine konkrete Modellierungsmethode für die modellgetriebene Servicebeschreibung definiert. Für diese Arbeit gilt nachstehende Definition:

Definition 2-2: Modellierungsmethode

Eine Modellierungsmethode ist eine Methode, die die Erstellung und Pflege einer Klasse von Modellen unterstützt. Sie umfasst i. d. R. eine oder mehrere Modellierungssprachen, ein Verzeichnis projektspezifischer Rollen und Ressourcen, ein Vorgehensmodell, ggf. Kriterien oder Metriken zur Evaluierung von Modellen sowie Anwendungsbeispiele.

2.3 Metamodellierung

Neben der Modellierungsmethode ist die Metamodellierung das zweite, elementare Konzept zur Realisierung des Rahmenwerks. Metamodellierung als Konzept ist essentiell, da die beschriebenen Elemente einer Methode, vornehmlich die Sprache und die Handlungsanleitung gegenüber dem zu erstellenden Modell auf einer Meta-Ebene existieren (vgl. (Greiffenberg 2004, S. 35), (Strahinger 1996, S. 24ff)). In Bezug zur Modellierungsmethode ist ein Metamodell das konzeptuelle Datenmodell, da es alle verwendeten Konstrukte zueinander in Beziehung setzt (Blessing und Fleisch 2000, S. 22), (Ferstl und Sinz 2001, S. 130f). Das Metamodell dient somit als formalisierte Beschreibung der Modelle der Modellierungsmethode und beschreibt deren Struktur bzw. ist selbst als Modell zu verstehen, das Aussagen über die Modellierung trifft (Strahinger 1996, S. 23). Ein bekanntes Beispiel für ein Metamodell ist die Unified Modeling Language (UML), deren Metamodell die Meta Object Facility (MOF) ist, womit MOF das Metametamodell der auf UML basierenden Modelle ist. Ein weiteres Beispiel für ein Metametamodell ist das von MOF abgeleitete Metametamodell Ecore. Ein Metametamodell erfüllt dabei denselben Zweck für ein Meta-

modell wie selbiges für ein Modell. Die Zusammenhänge sind in Abbildung 2-2 für den sprachbasierten Metamodellbegriff abgebildet.

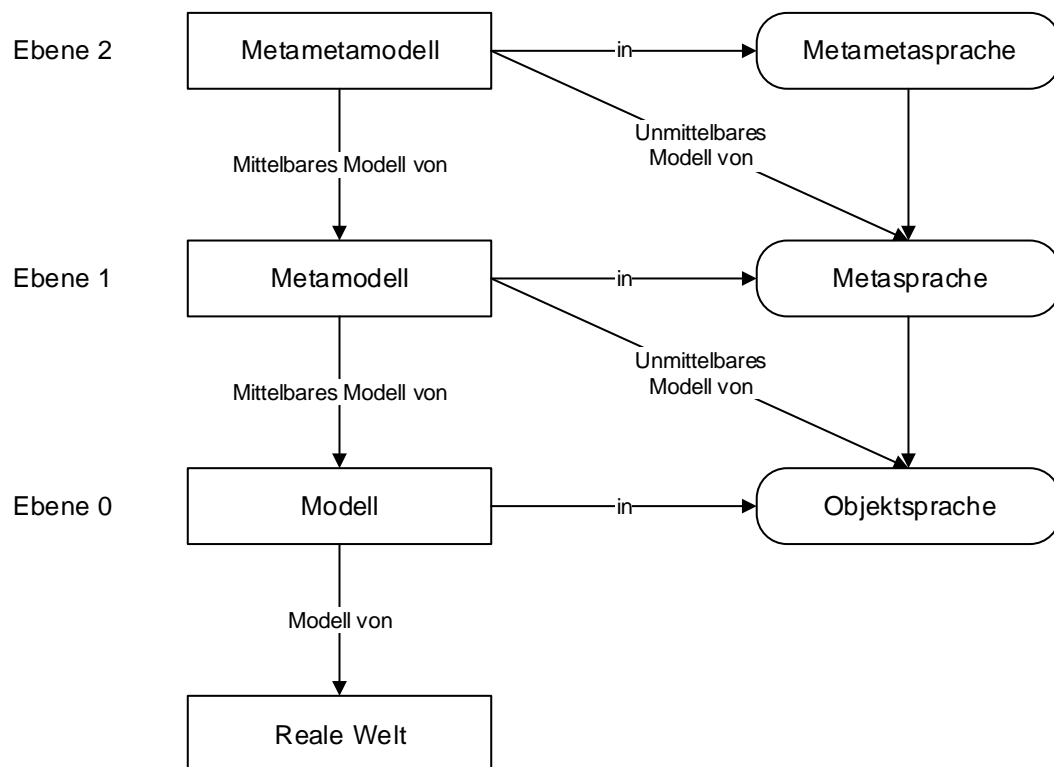


Abbildung 2-2: Sprachbasiertes Metaisierungsprinzip nach (Strahinger 1998)

Der sprachbasierte Metamodellbegriff und das dahinterliegende Metaisierungsprinzip (Strahinger 1998) beschreiben die Abhängigkeiten zwischen Modell, Metamodell und Modellierungssprache, wobei ein Modell der Ebene i immer nur ein mittelbares Modell der Ebene $i-1$, aber unmittelbares Modell der Sprache auf Ebene $i-1$ ist. Das bedeutet, in Übereinstimmung mit den oben beschriebenen Zusammenhängen, dass mit Hilfe eines Modells eine Sprache definiert wird, die zur Beschreibung eines realweltlichen Sachverhalts geeignet ist. Ein Metamodell erfüllt denselben Zweck in Bezug zu diesem Modell, usw. Das Metaisierungsprinzip beantwortet hierzu die Frage, welche Art von Modell eines Modells ein Metamodell ist (Mansfeld 2012, S. 18). Es wird per Konvention auf der Metametaebene beendet, indem das Metametamodell zusätzlich die Eigenschaft erhält, dass es selbstdefinierend ist, d.h. mit Sprachelementen der Metametasprache kann das Metametamodell selbst definiert werden (vgl. auch (Strahinger 1998), (Karagiannis und Kühn 2002), (Mansfeld 2012, S. 19)). Neben dem Sprachbasierten existieren darüber hinaus auch das prozessbasierte, das linguistische und das ontologische Metaisierungsprinzip. Für die Modellierungsmethode ist dabei neben dem sprachbasierten vor allem das prozessbasierte Prinzip relevant, da bei letzterem die Herausbildung der Handlungsanleitung mit Hilfe von

Modellen im Vordergrund steht. Die vorliegende Arbeit verwendet nur das sprachbasierte Prinzip, außer es wird explizit auf ein Anderes verwiesen.

2.4 Modelltransformationen

Modelltransformationen dienen dem Zweck Informationen zwischen unterschiedlichen Modellen zu transferieren. Transformationsansätze spielen für die Entwicklung der modellgetriebenen Servicebeschreibung eine entscheidende Rolle. Durch den Verzicht der Erstellung von Totalmodellen, repräsentieren unterschiedliche Modelle einzelne Perspektiven. Das ist vergleichbar mit der Sicht auf Modelle von (Czarnecki und Helsen 2006). Diese beschreiben ein Modell als „Antwort“ auf eine bestimmte Klasse von Fragen, die Bezug zur realen Welt besitzen. Für unterschiedliche Fragen sind demnach unterschiedliche Antworten (sprich Modelle) geeignet, wodurch Informationen u.U. durch verschiedene Modelle repräsentiert und daher transformiert werden müssen. Ein wesentliches Element der modellgetriebenen Servicebeschreibung besteht darin, Informationen unterschiedlicher Modelle in einen gemeinsamen Kontext zu bringen und Informationen zwischen Modellen zu übertragen und so die Entwicklung zu vereinfachen und zu beschleunigen. Grundsätzlich sind Transformationsansätze zu unterscheiden in Modell-zu-Text- (M2T) und Modell-zu-Modell-Ansätze (M2M). Erste sind insbesondere relevant bei der Generierung von Quellcode, bspw. aus UML-Klassendiagrammen. Diese Art der Modelltransformation wird im nachfolgenden nicht mehr weiter beschrieben oder berücksichtigt. M2M-Transformationen sind in ihrer Art und Beschaffenheit komplexer, vielschichtiger und allgemeiner ausgelegt, da die M2T-Transformation auch als Spezialfall der M2M-Transformation gesehen werden kann, bei der das zu generierende Modell ein textuelles und nicht ein grafisches Modell darstellt. Abbildung 2-3 stellt schematisch die Zusammenhänge bei einer Modelltransformation dar.

Soll ein Modell M_I in ein Modell M_O ganz oder teilweise transformiert werden, so ist zunächst die Definition eines dritten Modells M_T notwendig, das beschreibt, wie sich Elemente des Input-Modells zu Elementen des Output-Modells verhalten, d.h. das Transformations-Modell besteht aus Regeln der Art:

- $M_I.\text{Element} \rightarrow M_O.\text{Element}$
- $M_I.\text{Beziehung} \rightarrow M_O.\text{Beziehung}$

Jedes der Modelle M_I , M_O und M_T gehorcht dabei einem spezifischen Metamodell MM_I , MM_O bzw. MM_T , wobei insbesondere MM_I und MM_O nicht zwangsläufig identisch sein müssen. Daher muss ein Metametamodell die Brücke zwischen allen betroffenen Modellen

schlagen und es ermöglichen, dass die Konstrukte der jeweiligen Modelle auf der Metaebene kompatibel gemacht werden.

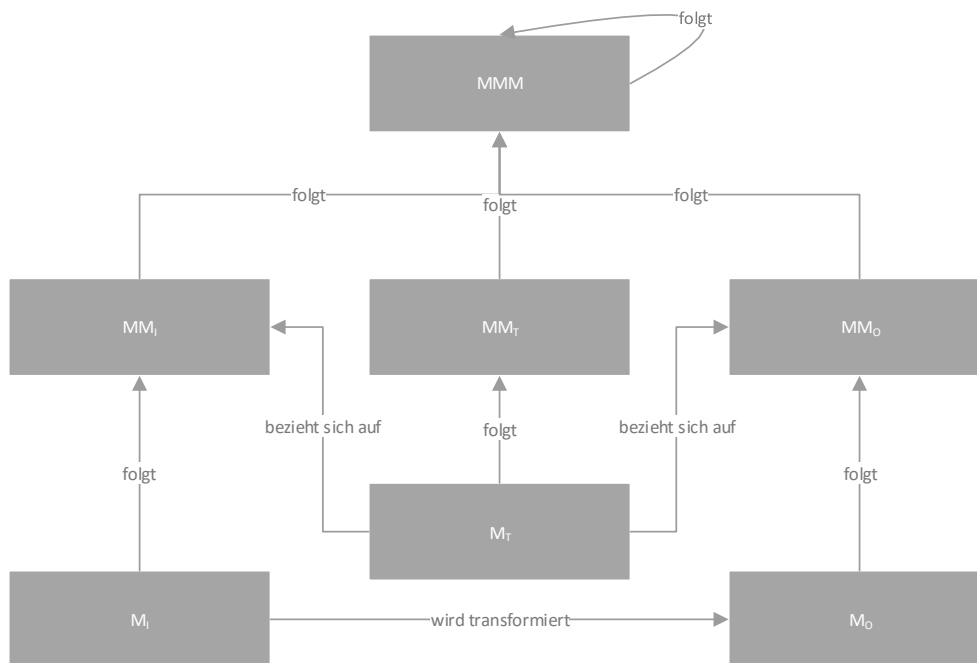


Abbildung 2-3: Modelltransformation - Elemente und Beziehungen

Dieses Vorgehen wird von (Kent 2002, S. 291) auch „language translation“ genannt, da letztenendes auf der Metamodellebene und damit auf der Sprache, mittels derer das Modell beschrieben ist, operiert wird. Das Gegenstück dazu, bei dem direkt auf der Modellebene Regeln zur Überführung angegeben werden, wird „model translation“ genannt. Dementsprechend gilt für diese Arbeit die von (Kleppe, Warmer et al. 2003, S. 24) weitestgehend übernommene und leicht erweiterte Definition für eine Modelltransformation:

Definition 2-3: Modelltransformation

Eine Modelltransformation, oder einfach nur Transformation, ist die (teil-) automatische Generierung eines Zielmodells aus einem oder mehreren Quellmodellen. Ihr zugrunde liegen Regeln, die beschreiben, wie Elemente aus einem Quellmodell in Elemente des Zielmodells überführt werden. Das Ziel einer Transformation ist primär die Übertragung von Informationen zwischen den Modellen.

Die Regeln, auch Mapping genannt (Gruhn, Pieper et al. 2006, S. 153), sind der elementare Bestandteil einer Transformation. Mit Hilfe des Vokabulars einer geeigneten Modellierungssprache (vgl. 2.2) beschreibt das Mapping, wie sich Input- und Outputelemente zueinander verhalten. Dazu müssen mit der Sprache des Mappings die jeweiligen Elemente

abbildbar sein, was die Sprachauswahl wesentlich beeinflusst. Darüber hinaus existiert für die Umsetzung von Transformationen eine Reihe von Eigenschaften, die je nach Ausprägung die Umsetzung entscheidend in der Ausgestaltung beeinflussen. Für eine gewählte Sprache muss insbesondere gelten, dass sie sich für die konkrete Problemstellung eignet, mindestens ein Werkzeug zur Automatisierung verfügbar ist und dass sie möglichst auf standardisierten Werkzeugen aufbaut, bzw. nutzt (Jouault und Kurtev 2007, S. 116). Der morphologische Kasten in Tabelle 2-1 fasst eine Vielzahl von Eigenschaften, die (Gruhn, Pieper et al. 2006; Mens und Van Gorp 2006; Jouault und Kurtev 2007; Amrani, Dingel et al. 2012; Calegari und Szasz 2013) entnommen sind, zusammen. Je nach Kombination der Eigenschaften ergeben sich somit völlig unterschiedliche Herangehensweisen, Anwendungsgebiete und Lösungsräume, die eine Anpassung an eine spezifische Problemstellung ermöglichen.

Einbezogene Modelle	1:1	1:n	n:1	n:m		
Einbezogene Metamodelle	1 (endogen)		n (exogen)			
Endogene Transformation	Optimierung	Refactoring	Vereinfachung	Normalisierung	Anpassung	Verfeinerung
Exogene Transformation	Synthese		Reverse Engineering		Migration	
Abstraktionsstufe	Horizontal			Vertikal		
Anwendungsbe- reich	In-place			Out-place		
Transformations- gegenstand	Syntax			Semantik		
Automatisie- rungsgrad	Manuell		Teil-automatisiert		Voll-automatisiert	
Komplexitäts- grad	Einfach			komplex		
Invariante	Struktur	Verhalten	Integrität	Korrektheit	Vollständig- keit	
Paradigma	deklarativ (was)		imparativ (wie)		funktional (was)	
Anforderungen	vollständig	wohlgeformt	determiniert	korrekt	endlich	
Richtungsgebun- denheit	Unidirektional			bidirektional		
Prägnanz	wenige Konstrukte			viele Konstrukte		
Regelausführung	interpretiert			kompiliert		
Flexibilität	eingeschränkt		erweiterbar		konfigurierbar	

Tabelle 2-1: Morphologischer Kasten zu Transformationseigenschaften

Für das SMF wird in Kapitel 7.4 explizit auf die Ausgestaltung der hier verwendeten Transformationen eingegangen und eine Parametrisierung der Eigenschaften vorgenommen. Ein Spezialfall der Modelltransformationen ist das so genannte Model Weaving. Anstatt von Regeln werden Korrespondenzen zwischen Modellen definiert und diese Korrespondenzen in einem so genannten Weaving-Modell festgehalten (Del Fabro, Bezivin et al. 2005). Das resultierende Weaving-Modell kann dann als Ausgangsbasis für eine nachfolgende, automatisierte Modelltransformation herangezogen werden. Die Korrespondenzen können dabei als direkte, typisierte Verbindung zwischen Elementen unterschiedlicher Modelle betrachtet werden (Jouault und Kurtev 2007), was eine Automatisierung weiter begünstigt, da mit einer Typisierung der Verbindung wichtige Eigenschaften der Transformation bereits beschrieben sind. Insbesondere kann dadurch die Integrität und Korrektheit der Transformation sichergestellt werden, insofern die Typisierung korrekt durchgeführt wurde. Voraussetzung für die Anwendung des Model Weaving ist die Definition eines Metamodells, mit dessen Hilfe die erlaubten Typen und Verbindungen definiert werden (Jouault und Kurtev 2007). Der Nachteil dieses Vorgehens liegt in der Nutzer-spezifischen Definition des zugrundeliegenden Metamodells und der darin zugrundeliegenden Semantik. Während existierende Transformationsansätze zumeist standardisiert ablaufen und eine strikte Menge an Operationen und Parametern besitzen, ist dies naturgemäß beim Model Weaving nicht der Fall, wodurch es zu einer höheren Fehleranfälligkeit bzw. zu einer durch falsche Definitionen hervorgerufenen fehlerhaften Transformation kommen kann. Der Vorteil besteht in der Flexibilität im Sinne der Anwendbarkeit in verschiedenen Domänen, da in diesem Fall nur das Metamodell ausgetauscht werden muss.

2.5 Modellgetriebene Entwicklung

Die MDE bietet einen Rahmen für die bislang vorgestellten Konzepte und ermöglicht deren Anwendung. Unter ihr sind Konzepte zusammengefasst, die den Umgang mit Modellen erleichtern und es ermöglichen Modelle als Artefakte der Informationsbereitstellung zu nutzen. Modelle sind dabei Informationsträger und können mit Methoden und Werkzeugen des MDE erzeugt, manipuliert oder gelöscht werden. Im Vordergrund stehen dabei insbesondere solche Methoden, die die Umwandlung von Modellen betreffen, d.h. die Modelltransformationen beschreiben. Ihren Ursprung findet MDE in der Softwareentwicklung. Dort wurden in erster Linie M2T-Transformationen genutzt, um bspw. aus UML-Klassendiagrammen direkt Quellcode zu erzeugen. Dementsprechend sind Arbeiten und Veröffentlichungen geprägt von Beispielen und Analogien zur Softwareentwicklung. Allerdings hat sich die MDE stark weiterentwickelt und nutzt dieselben Methoden und Werk-

zeuge, um mittlerweile beinahe beliebige Modelle zu manipulieren. Dennoch sind die meisten MDE-basierten Ansätze noch immer geprägt von einem Fokus auf Architekturen und Automatisierung sowie die Anwendung von Modelltransformationen, im Sinne einer M2M-Transformation (Kent 2002). Dementsprechend ist mit einem MDE-Ansatz verbunden, dass verwendete Modellierungssprachen, angewandte Transformationen sowie notwendige Prozesse zur Erstellung und Weiterentwicklung von Modellen definiert werden (Kent 2002, S. 294f). (Bézivin 2004) beschreibt MDE mit einer Analogie zur Objektorientierung in der Programmierung. Während dort das Konzept „Alles ist ein Objekt“ vorherrschend ist, soll dies auf MDE zu einem „Alles ist ein Modell“ umgewandelt werden. Dementsprechend repräsentieren Modelle tatsächliche Systeme oder reale Zusammenhänge analog zu Klassen, die durch Objekte instanziiert werden und zugleich gehorchen Modelle ihren Metamodellen analog zu der Vererbungsbeziehung zwischen Klassen und ihren Oberklassen. Modelle werden daher auch als „Bürger erster Klasse“ im Rahmen von MDE beschrieben (Bézivin 2006), was bedeutet, dass sie der Dreh- und Angelpunkt bei MDE sind. Eingesetzte Technologien und Werkzeuge, Anwendungsgebiete und Verfahren werden auf die Verwendung von Modellen als Trägermedium für Informationen ausgerichtet. Dabei können innerhalb eines Projekts auch unterschiedliche Modelle, Technologien und Werkzeuge zum Einsatz kommen, die spezifisch auf die konkrete Aufgabe ausgelegt sind (Favre und Nguyen 2005). Wesentlich für MDE-Ansätze ist dabei der Aspekt des „Separation of Concerns“, also der Aufteilung in spezifische für die jeweilige Problemstellung geeignete Sichten bzw. Modelle, einem der Hauptprinzipien der Softwarearchitektur (vgl. auch (Parnas 1972; Dijkstra 1976)). Ein Beispiel für einen weit verbreiteten MDE-Ansatz ist die MDA. Mit ihrer 3-Ebenen Architektur bestehend aus den Modelltypen CIM, PIM und PSM wird aus anfangs abstrakten Modellen und Beschreibungen sukzessiv lauffähige Software erzeugt. Während mit den CIM-Modellen Anforderungen, Zusammenhänge auf der Ebene der Geschäftsprozesse und Randbedingungen der zu entwickelnden Systeme modelliert werden, beschreiben PIM-Modelle Struktur und Funktionalität dieser Systeme, jedoch ohne konkreten Bezug zu eingesetzten Sprachen oder Technologien. Diese kommen auf der untersten Ebene bei den PSM-Modellen zum Tragen. MDA ermöglichte damit einen Wechsel weg von einer Code- hin zu einer Modell-zentrierten Sicht bei der Softwareentwicklung. Der MDA-Ansatz ist daher ein Spezialfall des MDE und ist der modellgetriebenen Softwareentwicklung (Model-driven Software Development, MDSD) zuzuordnen, die als ein Oberbegriff Techniken, die aus formalen Modellen automatisiert lauffähige Software erzeugen, bündelt (Stahl, Völter et al. 2007, S. 11). Wichtig dabei ist, dass insbesondere auf den Software-nahen Ebenen das Formalitätskriterium für die Modelle gilt, d.h.

dass die verwendeten Modelle mathematisch exakt, widerspruchsfrei und vollständig definiert sind (Petrasch und Meimberg 2006, S. 42). Ein weiteres, konkretes Beispiel für die Anwendung von MDE ist der Ansatz von (Bitsaki, Danylevych et al. 2009), die vergleichbar zu dieser Arbeit MDE mit Konzepten der Serviceorientierung verbinden. Aufgrund des Fokus der Arbeit auf BPMN und BPEL wird dieser Ansatz jedoch nicht in Kapitel 4 erwähnt. (Bitsaki, Danylevych et al. 2009) modellieren Prozesse eines so genannten Service-Netzwerks mit BPMN-Mitteln, realisieren Prozesse mit Web-Services und einer BPEL-Beschreibung und vergleichen beides mit Hilfe von Transformationen, um den Realisierungsgrad zu überwachen und Optimierungen durchzuführen. Der Ansatz zeigt sehr schön die entstehenden Potenziale aus der Verbindung von MDE und SO. MDE ermöglicht eine automatisierte Verarbeitung unterschiedlicher Serviceaspekte, ausgedrückt durch Modelle, und kann so helfen ein vollständigeres Bild einer Servicelandschaft zu entwickeln. Dass dies notwendig ist, zeigt das nächste Kapitel 3, das sich intensiv mit dem Servicebegriff und Servicemodellierungsansätzen befasst. Zunächst werden in einer Zusammenfassung alle für diese Arbeit relevanten Punkte der letzten Abschnitte zusammengeführt.

2.6 Zusammenfassung

Modellierungskonzepte sind für die Umsetzung der modellgetriebenen Servicebeschreibung essentieller Bestandteil: Sämtliche der vorausgesetzten und entstehenden Artefakte sind Modelle oder zumindest modellbasiert. Daher sind die GoM als Qualitätsmerkmal für Modelle entscheidend. Die GoM repräsentieren Richtlinien sowohl bei der Erstellung, als auch bei der Analyse von Modellen und ermöglichen eine effiziente Verarbeitung der Modelle. Sie stellen somit auch wichtige Anforderungen an die verwendeten Modelle dar (vgl. auch 5.2). Dazu zählt auch die Auseinandersetzung mit dem Modellbegriff selbst. Erst wenn eine hinreichend präzise Definition für diesen gegeben ist, können Modelle dementsprechend entworfen werden. Dabei spielen für diese Arbeit insbesondere die Zweck- und Subjektbezogenheit eine wichtige Rolle, erklären sie doch die Vielfalt an Modellen, oder das Vorhandensein von redundanten, teils widersprüchlichen Modellen, Modellarten oder Begriffen in Modellen. Unter diesem Gesichtspunkt sind auch die unterschiedlichen Modellierungsparadigmen zu nennen, die als Grundlage der Modellierung dienen. Wichtig ist demzufolge eine Einschränkung dahingehend, dass für diese Arbeit nur bestimmte Paradigmen (deskriptiv, entscheidungsunterstützend und präskriptive) und Arten (Partialmodelle) eingesetzt werden können. Erstere Einschränkung führt dazu, dass die Modelle bzgl. der Beschreibung ihres Inhalts vergleichbar sind, d.h. dass eine Informationsübertragung zwischen ihnen sinnvoll ist. Letztere Einschränkung führt schließlich zur Notwendigkeit

eines integrativen Ansatzes, ist im Zuge der Fokussierung auf Services jedoch selbst notwendig, um der Unschärfe in der Servicedefinition (vgl. 3) zu begegnen.

Das sprachbasierte Metaisierungsprinzip wird in dieser Arbeit verwendet, um auf Basis der Metamodellierung eine Modellhierarchie abzuleiten (vgl. 6.1). Außerdem unterstützt das Prinzip die Ableitung einer geeigneten Modellierungsmethode (vgl. 6.4), da die Spezifikation einer Handlungsanleitung, als Bestandteil einer Modellierungsmethode, beim sprachbasierten Prinzip im Vordergrund steht. Konzepte der modellgetriebenen Entwicklung bilden schließlich die Klammer, verbinden wichtige Modellkonzepte und umfassen Lösungen zur Arbeit mit Modellen, die der Programmierung von Softwareartefakten ähnlich sind. Modelle werden dadurch zu einem Gebrauchsgegenstand der modellgetriebenen Servicebeschreibung und zur wichtigsten Klassen von Elementen. Sie sind Träger von Informationen, können manipuliert, ausgetauscht oder erweitert werden, ähnlich zu den Objekten des objektorientierten Programmierparadigmas. Abschließend wird im unmittelbar nachfolgenden Exkurs der Zusammenhang zwischen der Referenzmodellierung und dem hier entwickelten Ansatz beschrieben.

Exkurs I: Referenzmodellierung

Der Begriff der Referenzmodellierung kommt in der Wirtschaftsinformatik zunächst im Zusammenhang mit Arbeiten vor, in denen Informationsmodelle (insbesondere Daten- und Prozessmodelle) für spezifische Domänen entwickelt werden. Die Referenzmodellierung oder auch Referenzinformationsmodellierung ist ein Teilgebiet der Modellierung, die sich generell mit Modellen für Informationssysteme beschäftigt und dabei Modellklassen für Systeme beschreibt, d.h. eine gewisse Allgemeingültigkeit der Modelle fordert (vgl. (Vom Brocke 2003, S. 31), (Fettke und Loos 2002, S. 9), (Becker und Delfmann 2004)). Beispielhaft seien als Anwendungsdomänen genannt:

- Method Engineering (Becker 2007),
- Management von Prozessvarianten (Hallerbach 2010),
- eGovernment (Karow 2008),
- IT Service Management (Brocke 2010),
- Energiemanagement (Schlieter 2010),
- Standardisierungsinitiativen (Hofreiter 2012),
- Change Management (vom Brocke 2006b),
- Maschinensimulation (Esswein 2010),
- Hochschulmanagement (vom Brocke 2006a),
- eHealth (Baacke 2009),

Ansätze finden sich darüber hinaus im Bereich des eLearnings, in der Gestaltung von Smart Grids, sowie in der Gestaltungsorientierten Forschung generell. Bekannteste Beispiele für Referenzmodelle sind das Handels-H-Modell für Handelsunternehmen (Becker und Schütte 2004, S. 42ff,72f) sowie das ARIS-Haus für Industriebetriebe (Scheer 1997). Eine Modellklasse stellvertretend für eine Reihe von Informationsmodellen, die eine spezifische Domäne oder Systeme beschreiben, wobei die Modellklasse selbst die Konstruktion dieser Modelle beschreibt (Vom Brocke 2003, S. 37). Ein Referenzmodell enthält dazu Bausteine, die für die Konstruktion notwendig sind, lässt zugleich aber genügend Spielraum zur Anpassung an konkrete Systeme (Vom Brocke 2003, S. 36f). Der Zweck von Referenzmodellen ist daher eng verbunden mit der Entwicklung und dem Customizing von Standardsoftware für spezifische Branchen oder Wirtschaftssektoren. Darüber hinaus dienen sie zur „Konstruktion unternehmensspezifischer Modelle auf Basis vorgefertigter Modelle bzw. Modellbausteine“ (Fettke und Loos 2002, S. 9). In beiden Fällen spielt die möglichst effiziente und wirtschaftliche Wiederverwendung von Modellinhalten eine wesentliche Rolle (Vom Brocke 2003, S. 37). Eine mögliche Zielsetzung ergibt sich daher in einer schnelleren System Einführung bei geringeren Risiken des Scheiterns (Goeken 2003, S. 1). Für die vorliegende Arbeit sind die Ansätze von Hars (Hars 1994), Lang (Lang 1997), Remme (Remme 1997) und insbesondere Delfmann (Becker, Delfmann et al. 2002; Delfmann 2006) relevant.

Der Ansatz von Hars zeigt die Entwicklung von Datenmodellen mit Hilfe von Softwarewerkzeugen mit starkem Fokus auf der Wiederverwendung bestehender Unternehmensmodelle und einzelnen Modellelementen. Dazu führt Hars eine eigene, auf Graphen-basierende Beschreibungssprache sowie eine Klassifikation von Modellelementen ein und erweitert bestehende Modelle und Elemente um zusätzliche Merkmale. Dabei sind sowohl der Fokus auf Datenmodelle wie auch die Wiederverwendung bereits bestehender Modelle ein wichtiger Aspekt für diese Arbeit.

Der Ansatz von Lang hebt die bausteinorientierte Modellierung hervor, um insbesondere eine branchenübergreifende Nutzung von Referenzmodellen zu erreichen. Dies entspricht sehr stark dem Gedanken des Separation of Concerns der modellgetriebenen Servicebeschreibung. Wesentliche Zielstellung in beiden Arbeiten ist eine konzeptionelle Trennung der unterschiedlichen Beschreibungsmöglichkeiten (Modelle) zum Zwecke einer eindeutigeren und stärker zweckgebundenen Beschreibung. So führt Lang bspw. aus, dass Prozessabläufe eine zeitlich-logische Anordnung von Prozessbausteinen darstellen und damit anders gehandhabt werden müssen als Prozesslösungen, die durch Ressourcen, Aktivitäten und Input-Output-Vorgängen beschrieben sind (vgl. auch (Fettke und Loos 2002, S. 20)).

Der Schwerpunkt bei Remme liegt im Versagen bei der systematischen Dokumentation von getroffenen Entscheidungen im Organisationsprozess. Auch diese Arbeit fokussiert zunächst die Beschreibung mittels Prozessmodellen. Dabei entsteht ein vollständiges Prozessmodell aus einzelnen Bausteinen, analog zur Herangehensweise bei der modellgetriebenen Servicebeschreibung, die ein vollständiges Servicemodell durch Verbindung mehrerer, zweckgebundener Modelle zu einem Gesamtbild zu entwickeln versucht. Eine Stärke in der Arbeit von Remme besteht dabei in einer fortschreitenden Dokumentation bei der Konstruktion (vgl. auch (Fettke und Loos 2002, S. 99-110).

Die adaptive Referenzmodellierung von Delfmann wird im Vergleich zu den drei erst genannten nicht aufgrund ihrer inhaltlichen Nähe, aber aufgrund ihres Konstruktionsprozesses als relevant eingestuft. Sie stellt einen Prozess für die Erstellung bzw. Anpassung von Referenzmodellen zur Entwicklung daraus abgeleiteter Informationsmodelle vor. Außerdem wird die Entwicklung einer Sprache zur Modellierung von Referenzmodellen beschrieben. Eines der wesentlichen Strukturierungselemente stellt der Ordnungsrahmen dar, welcher die Strukturierung der zu modellierenden Domäne und die Navigation durch und zwischen den Domänenbestandteilen ermöglicht. Üblicherweise wird dabei von einem Top-Down-Ansatz ausgegangen, bei dem sowohl der Ordnungsrahmen wie auch das Gesamtmodell als wichtigste Artefakte zuerst vollständig beschrieben sein müssen. Im Falle, dass Verfeinerungsmodelle eine Umarbeitung der Artefakte notwendig werden lassen kann jedoch auch auf andere Strategien wie bspw. Bottom-up zurückgegriffen werden (Becker, Delfmann et al. 2002, S. 51). Dadurch werden im Zuge des Entwicklungsprozesses vermehrt Schnittstellen zwischen dem Gesamt- und den Verfeinerungsmodellen notwendig. Den Kern der Arbeit bilden so genannte Projektionen. Sie repräsentieren einzelne Verfeinerungsmodelle, die Bestandteile des Gesamtmodells in einer höheren Detaillierungsstufe abbilden. Dabei können Projektionen auf fünf unterschiedlichen Stufen vorkommen:

- **Modelltyp.** Variation in der verwendeten Modellierungssprache
- **Elementtyp.** Einschränkung von erlaubten Elementen innerhalb einer Modellierungssprache (z. B. keine Ereignisse bei Verwendung von BPMN)
- **Elemente.** Einschränkungen bzgl. des Wertebereichs von Elementen, so dass bspw. nur aggregierte Werte erlaubt sind.
- **Namenskonventionen.** Einschränkung und Vereinheitlichung des genutzten Vokabulars bspw. einiger weniger Synonyme innerhalb eines spezifischen Kontexts.

- **Repräsentation.** Gestaltung und Verwendung von geeigneten Symbolen für eine Modellierungssprache zur Unterstützung der Nutzer innerhalb einer bestimmten Domäne.

Die Projektionen ermöglichen einem Modellierer dabei eine spezifische Sicht auf das Gesamtmodell, mit einem reduzierten, nur für ihn relevanten Ausschnitt (Delfmann 2006, S. 65) und repräsentieren elementare Konfigurationsmechanismen der Arbeit, deren Ziel es ist, eine adaptive Modellierungssprache zu entwickeln, mittels derer kontextabhängig bestehende Modellierungssprachen an die konkreten Erfordernisse im Modellierungsprozess angepasst werden können (Delfmann 2006, S. 15) und so der Aufwand bei der Erstellung eines Referenzmodells reduziert werden kann. Auf die Anwendung der Kernkonzepte im Zuge der modellgetriebenen Servicebeschreibung wird in Kapitel 5 nochmals eingegangen.

3 Services

Die Modellierung von Services ist der zentrale Gegenstand der Arbeit. Das Ziel der Arbeit, die Konstruktion eines Rahmenwerks zur Servicebeschreibung kann aber nur gelingen, falls Services mit Hilfe von Partialmodellen hinreichend präzise beschrieben werden können. Allerdings fällt bereits eine inhaltliche Abgrenzung für den Service- oder auch Dienstleistungsbereich komplex aus. Der Grund dafür liegt in der schwer fassbaren Natur von Services (s. auch 3.1.1 oder 3.1.2). Dementsprechend wird die Domäne der Services in der Forschung teils kontrovers, aber vor allem sehr umfangreich seit vielen Jahren diskutiert. In der Forschung haben sich dazu mehrere Richtungen etabliert, wovon drei Wesentliche hier kurz vorgestellt werden sollen. Sie repräsentieren jeweils wichtige Strömungen in ihrem länderspezifischen Kontext und beinhalten wichtige Konzepte für diese Arbeit bzw. haben insgesamt einen starken Einfluss auf die Entwicklung serviceorientierter Konzepte. Im Deutschsprachigen Raum sind das vor allem die Dienstleistungsmodellierung (Thomas und Nüttgens 2010) und das Service Engineering (SE, (Bullinger und Scheer 2006)), im Anglo-amerikanischen Raum insbesondere die Service Science bzw. Science of Services (Alter 2008; Dorne, Lesaint et al. 2008; Spath und Ganz 2008). In jedem Zweig sind die Schwerpunkte bei der Konzeption von Services anders gewichtet, was auch durch die leicht unterschiedliche Namensgebung zum Ausdruck kommt. Die Dienstleistungsmodellierung steht insbesondere für den Modellaspekt bei der Entwicklung und Anwendung von Services, das SE umfasst Ansätze zum systematischen und ingenieurmäßigen Entwickeln und die Service Science ist ein multidisziplinärer Ansatz zur ganzheitlichen Forschung und Gestaltung von Services. Für diese Arbeit ist das SE dabei von besonderem Interesse. Zwar erscheint zunächst die Dienstleistungsmodellierung aufgrund ihres Fokus auf die Modellie-

rung relevant, die Verbindung dieses Ansatzes mit der MDE und die daraus resultierende, auch ingenieurmäßige Entwicklung geeigneter Werkzeuge führt zur einer starken Korrespondenz mit Konzepten des SE (vgl. auch 1). Darüber hinaus nutzt auch das SE Modelle zur Beschreibung von Services. Das zentrale Anliegen des SE ist es, in einem interdisziplinären Ansatz ingenieurwissenschaftliche und softwaretechnische Verfahren auf die Dienstleistungsentwicklung zu übertragen (Scheer, Griebler et al. 2006, S. 20). Dem SE, aber auch den anderen Strömungen der Serviceentwicklung liegt dabei der Gedanke der SO zu Grunde. Auch wenn mit dem Begriff häufig eine SOA und damit eine eher technische Sichtweise assoziiert werden, ist die Serviceorientierung auch ein wichtiges Konzept in betriebswirtschaftlichen Ansätzen. In den nachfolgenden Abschnitten werden Services daher aus diesen beiden Blickwinkeln untersucht, um daraus ein Gesamtbild zu entwickeln.

Im Gegensatz zum betriebswirtschaftlich geprägten Blickwinkel wird die SO aus informationstechnischer Sicht in einem unternehmensweiten Architekturansatz (SOA) verwendet, mit dessen Hilfe fachliche Prozessarchitekturen flexibel umgesetzt werden können (Vogel 2009). Der Hauptvorteil einer solchen SOA liegt dabei in einer zusätzlichen Service-schicht, mittels derer technische Systeme von der Benutzeroberfläche entkoppelt werden. Die darin befindlichen Services umfassen klar abgegrenzte Geschäftslogikbausteine, die im Idealfall mehrfach und in unterschiedlichen Kontexten wiederverwendbar sind. Darüber hinaus können Services sowohl horizontal wie vertikal (Domänenbildung) segmentiert werden, was zu einer klaren Zuweisung von Verantwortung führt. Wichtige Zielstellungen einer SOA im Unternehmensumfeld sind die Erhöhung der Flexibilität durch Adaptierbarkeit von Geschäftsprozessen, kürzere Entwicklungszeiten, reduzierte Kosten sowie ein verbessertes Zusammenspiel von Fachabteilung und IT (Alignment) (Buchwald, Tiedeken et al. 2010).

Viele der genannten Eigenschaften oder Zielstellungen sind jedoch in höchstem Maße von den vorhandenen Services bzw. von deren Qualität und Zuschnitt, d.h. von der konkreten Ausgestaltung und Zusammensetzung der Servicebausteine, abhängig. So ist bspw. der Grad der Wiederverwendbarkeit stark von den implementierten Funktionen abhängig. Je spezialisierter die einzelnen Funktionen, umso geringer die potenzielle Wiederverwendbarkeit, aber auch: je allgemeiner oder je generischer eine solcher Service entwickelt wird, umso trivialer sind die implementierten Funktionen wodurch die Verwendbarkeit im Allgemeinen ebenfalls sinkt. Somit kommt der Identifizierung geeigneter Services im Unternehmen eine zentrale und wichtige Rolle zu. Die Identifizierung wird aber erst dadurch möglich, wenn Services mit Hilfe unterschiedlicher Perspektiven beschrieben und damit vergleichbar bzw. analysierbar gemacht werden (Demirkan, Kauffman et al. 2008). In die-

sem Punkt, der als Voraussetzung für eine erfolgreiche SO genannt wird, setzt diese Arbeit mit der modellgetriebenen Servicebeschreibung an und unterstützt die Serviceidentifizierung und -entwicklung mittels eines multiperspektivischen Modellierungsansatzes. Servicemodelle dienen dabei als Bezugspunkt zur Beschreibung relevanter Merkmale, mit deren Hilfe gezielt Services identifiziert werden können. Die modellgetriebene Servicebeschreibung verknüpft darüber hinaus bestehende Modelle zu einem größeren Ganzen und ermöglicht damit ebenfalls eine effiziente Serviceentwicklung durch Bereitstellung von Kontextinformationen.

3.1 Servicebegriff

“... many different and often incompatible approaches to describing, managing and providing services have been developed, and there is still a lack of consensus and sometimes clear understanding about what constitutes a service” (Dumas, O’Sullivan et al. 2003, S. 278).

Im englischen Sprachraum existiert im Gegensatz zum Deutschen kein Sprachwirrwarr bzgl. eines Services. Dort wird einheitlich der Begriff des „Service“ sowohl in der Literatur als auch in praktischen Anwendungsfeldern gebraucht. Im deutschen Sprachgebrauch dagegen ist es nicht immer eindeutig, ob „Service“ synonym zu „Dienst“ oder zu „Dienstleistung“ zu verstehen ist. Bereiche wie die Dienstleistungsforschung benutzen bspw. das Wort „Dienstleistung“ zur Abgrenzung zu anderen Forschungsbereichen. Innerhalb dieser Arbeit wird jedoch keine strikte Unterscheidung zwischen diesen Begriffen getroffen. Insbesondere sind die Begriffe „Service“ und „Dienst“ synonym zu verstehen. Der Begriff der „Dienstleistung“ wird verwendet, um die Dienstleistungserbringung als Bestandteil eines Geschäftsmodells zu beschreiben. Gegenstand dieser Erbringung sind die im Folgenden näher beschriebenen und definierten Services.

Die Erbringung im Rahmen einer Leistung eines Unternehmens für seine Kunden macht es notwendig, dass angebotene Services sorgfältig entwickelt und in das Leistungsprogramm des Unternehmens integriert werden. Dies umfasst vor allem auch die Ressourcennutzung für die Bereitstellung. Die Forderung nach einer systematischen Entwicklung (Bullinger und Scheer 2006, S. 4) sorgt in diesem Zusammenhang für eine effiziente Ressourcenauslastung bzw. dafür, dass die Planbarkeit der Auslastung und damit eine Vermeidung von Engpässen, gewährleistet werden kann. Unbeantwortet bleibt damit jedoch weiter die Frage was ein Service ausmacht oder besser: welche Bestandteile sind für eine Servicebeschreibung notwendig, damit dieser im Unternehmen effektiv genutzt werden kann und mit welchen Mitteln kann eine solche Beschreibung erfolgen?

3.1.1 Einordnung

Vor dem Versuch einer Definition, die für die vorliegende Arbeit ein ausreichendes Maß an Präzision beinhaltet, wird in diesem Abschnitt zunächst eine Einordnung von Services aus der Literatur wiedergegeben. Wie bereits dargelegt existieren mehrere, aktive Forschungsdisziplinen, die die Entwicklung und Beschreibung von Services umfassen, so dass je nach konkreter Ausrichtung der Wissenschaftler unterschiedliche Auffassungen existieren. Aus dem Bereich des Marketings wurden bereits früh Eigenschaften von Services definiert. So stellt (Lovelock 1983) fünf Eigenschaften vor: die Natur eines Services (tangibel oder intangibel), die Beziehung zwischen anbietendem Unternehmen und Kunden (formal / informal bzw. kontinuierlich / diskret), die angebotene Variabilität eines Service (customizing), die Nachfrageschwankung und die Bereitstellung des Service im Sinne der unterschiedlichen Interaktionsmuster zwischen Unternehmen und Kunde.

Der Fokus bei (Hill 1977) liegt dagegen im unterschiedlichen Wert, den ein Service für Unternehmen und Kunde annehmen kann. Dementsprechend sind die Eigenschaften, ob eine Person oder eine Sache durch den Service betroffen ist, ob die Wirkung des Service permanent oder temporär ist, ob die Auswirkungen reversibel oder nicht sind und ob Auswirkungen physischer Natur sind oder nicht, relevant für die Beurteilung.

Eher Technologie-orientiert sind die Ansätze von (Bitner, Brown et al. 2000), (Grönroos 2000) und (Lindgreen und Sempels 2001). Diesen Ansätzen ist gemein, dass sie im Wesentlichen eine Unterscheidung zwischen Services treffen, die auf Technologie basieren und auf solchen, die nur eine technologische Unterstützung erfahren (low / high tech bzw. low / high touch). Im Fokus dieser Ansätze steht dabei die Schnittstelle zwischen Unternehmen und Kunden bei der Serviceerbringung und damit verbunden der Anteil manueller Aktivitäten am Gesamtprozess der Leistungserbringung, im Gegensatz zu einer automatisierten Erbringung mit Hilfe von Informationstechnologie. (Grönroos 2000) stellt zusätzlich noch die Abrechenbarkeit eines Service als wichtige Eigenschaft dar, mittels der festgestellt werden kann, ob ein Service als wertschöpfend betrachtet werden kann oder nicht. Eine ähnliche Unterteilung nutzen (Lovelock und Gummesson 2004). Dort wird allerdings eine Unterscheidung zwischen Services mit hoher und niedriger Eigenleistung des Kunden getroffen und als Merkmal genutzt. Schließlich setzen (Cunningham, Young et al. 2004) auf den Grad der Standardisierbarkeit bzw. auf die Materialität des Service.

Die steigende Durchdringung serviceorientierter Architekturen in Unternehmen hat dazu geführt, dass Eigenschaften von Services mit stärkerem IT-Bezug untersucht wurden. In erster Linie stehen dabei eine differenziertere Betrachtung möglicher Servicetypen sowie Eigenschaften einer SOA im Vordergrund. (Erradi, Anand et al. 2006a) unterscheiden vier

Arten von Services, abhängig vom Einsatzzweck oder Komplexität: process services (Workflows), application services (Kapselung übergreifender Prozessaktivitäten), shared data services (Integrationsunterstützung für multiple Datenquellen) und infrastructure services (Hilfs- und Unterstützungsdienste, die grundlegende Funktionalitäten bereitstellen). Im zugehörigen Rahmenwerk SOAF (SOA Framework) werden diese Servicetypen verwendet, um Services ausgehend sowohl von der Ebene der Geschäftsprozesse als auch von der Ebene der Anwendungslandschaft aus bestehenden Funktionalitäten zu kapseln und zur Verfügung zu stellen. Services werden darüber hinaus auch in geschäftsübergreifend, kanalübergreifend oder kanalspezifisch eingeteilt, wodurch der Grad der Wiederverwendung innerhalb eines Unternehmens als wesentliche Eigenschaft Berücksichtigung findet.

Ähnlich verfahren auch (Legner und Vogel 2007). Sie legen aber nicht nur Servicetypen fest (business und technical services, wobei letztere noch in integration und infrastructure services unterschieden werden), sondern verwenden zusätzlich Granularitätsabstufungen (utility, entity und process services) zur Untergliederung einer Servicelandschaft und verwenden strukturelle Aspekte, wie Kompositionen (atomic und composite services), Kommunikationsstil (synchron / asynchron), den Zustand (zustandslos oder -behaftet) und Zugangsbeschränkungen (innerhalb / außerhalb eines Unternehmens) als zusätzliche Merkmale zur Kategorisierung.

Bereits dieser kurze Überblick zeigt, dass in unterschiedlichen Forschungsdisziplinen jeweils ein anderes Verständnis für Services vorherrscht. Zwar gibt es teils gleiche oder gleichartige Merkmale in mehreren Disziplinen, ein gemeinsames Bild lässt sich jedoch schwer zeichnen. Der nächste Abschnitt dient einer Schärfung des Servicebegriffs und mündet in einer für die vorliegende Arbeit geeigneten Servicedefinition.

3.1.2 Definition

Die Notwendigkeit einer möglichst präzisen Definition² erschließt sich, wenn man ein umfassendes Bild einer Servicelandschaft mit Hilfe von Modellen zu erstellen sucht. Ein solches Bild kann nur dann gelingen, wenn die Intention hinter den modellierten Services verstanden ist und damit nur dann, wenn diese Intention durch eine Reihe von Eigenschaften hinreichend beschrieben werden kann. Mittels der hier vorgestellten Definitionen wird versucht eine gemeinsame Menge an Eigenschaften herauszufiltern, die es erlauben einen Service hinreichend zu spezifizieren und daraus wichtige Perspektiven zur umfänglichen

² Im Duden wird „Definition“ als genaue Bestimmung eines Begriffes durch Auseinanderlegung oder als Erklärung seines Inhalts näher bestimmt.

Beschreibung abzuleiten. Wie bereits im vorhergehenden Abschnitt werden zunächst Ansätze aus unterschiedlichen Forschungsdisziplinen vorgestellt und abschließend zu einer einheitlichen Definition zusammengeführt.

In der Literatur werden unterschiedliche Begriffe teils synonym teils mit unterschiedlicher Bedeutung verwendet. Am gebräuchlichsten sind dabei die Begriffe „Service“, „e-Service“ oder „Web Service“ (Cardoso, Voigt et al. 2009). Diese und ähnliche werden allerdings in verschiedenen Forschungsdisziplinen und Anwendungsdomänen übergreifend verwendet (Baida, Gordijn et al. 2004b). Ausgehend von der obigen Aufteilung in eher wirtschaftlich bzw. informationstechnisch orientierte Definitionen werden zunächst Begriffsbestimmungen aus wirtschaftlich geprägten Anwendungsbereichen vorgestellt.

Typisch für diesen Bereich ist in der Regel die Hervorhebung der Interaktivität und des Beitrags zur Wertschöpfung, so zum Beispiel (Spohrer, Maglio et al. 2007) in (Demirkan, Kauffman et al. 2008, S. 358): „Ein Service ist die Anwendung von Kompetenzen und Wissen zur Wertschöpfung zwischen Anbieter und Empfänger. Der Wert wird dabei aus der Interaktion generiert, die in einem Service System miteinander verbunden sind.“

Wirtschafts-orientierte Definitionen beinhalten eher konzeptuelle Beschreibungen und greifen zur Herleitung auf empirische Untersuchungen zurück (Edvardsson, Gustafsson et al. 2005). Allerdings hat der technologische Fortschritt und insbesondere der Hype um SOA dazu geführt, dass durch Web Services und das Internet als Plattform auch in diesen Bereichen Services vermehrt als formalisierbare, kodifizierbare und ausführbare Prozessbausteine verstanden werden (Kushida und Zysman 2009, S. 177). Es entwickelten sich neue Begriffe wie „e-Service“ oder „information based service“ (Lovelock und Gummesson 2004), (Baida, Gordijn et al. 2004b), (Rowley 2006), (Alter 2010). Diese neue Begriffsvielfalt trug jedoch nicht zu einem besseren Verständnis oder zu einem Konsens bezüglich der in Betracht kommenden Konzepte bei. Stattdessen wurden die Begriffe verwendet, um Subkategorien zu entwickeln wie beispielsweise die Betonung des „e-Service“ als Internet-basierter Service. Andere Beispiele zeigen eine Verlagerung auf architekturelle Themen über SOA und Web Services bis hin zu soziotechnischen Themen, bei denen Services zwar über das Internet realisiert, aber in erster Linie zum Nutzen Dritter erbracht werden.

Betrachtet man im Vergleich technologisch motivierte Begriffsbestimmungen, so zeigt sich eine Verschiebung der Prioritäten hin zu einer Verfügbarkeit bzw. Ansprechbarkeit mittels geeigneter Schnittstellen. So zum Beispiel bei (López-Sanz, Acuña et al. 2008, S. 28), die einen Service als „... Mechanismus zum Zugang zu einer Menge von Fähigkeiten sehen, wobei der Zugang über bestimmte Schnittstellen erfolgt und dessen Ausführung konsistent

und nur im Rahmen seiner Beschreibung möglich ist“. Ein ähnlich, technisch motivierter Servicebegriff nach (Vogel 2009) beschreibt Services als Softwarekomponenten mit wohl definierten, stabilen Schnittstellen, die anderen Applikationen unter Nutzung von verbreiteten Standards Zugriff auf wiederverwendbare Softwarefunktionalität bieten. Häufig werden Services auch als elektronische Repräsentation von einzelnen Aktivitäten oder ganzen Prozessen gesehen (Piccinelli, W. et al. 2003), wobei die Tatsache, ob ein Service z.T. durch menschliche Intervention unterstützt wird, nebensächlich ist. Die letzte Auffassung wird u.a. durch (Berardi, Calvanese et al. 2003) in (Alter 2010) vertreten. In deren Definition ist nur relevant, dass die leistende Entität ein Softwareartefakt ist. Konsument und Anbieter können entweder manuell oder auch automatisiert mit diesem Artefakt interagieren. Für eine technologisch motivierte Definition werden auch Eigenschaften aus dem Bereich der Softwaretechnik herangezogen. Hier sind insbesondere die Eigenschaft der Komponierbarkeit (Erl 2005), (Overhage und Turowski 2007), (Nayak, Nigam et al. 2006) und der Wiederverwendbarkeit (Krafzig, Banke et al. 2005b), (Legner und Heutschi 2007), (Legner und Vogel 2007) wesentlich für Servicedefinitionen. In der folgenden Tabelle 3-1 werden die gewonnen Erkenntnisse aus der Literatur noch einmal zusammengefasst. Ein Service:

Eigenschaft	Quelle
Ist eine logische Arbeitseinheit / repräsentiert einzelne Aktivitäten oder Prozesse	(Erl 2005), (Grönroos 2000), (Gustafsson und Johnson 2003), (Vargo und Lusch 2004), (Chesbrough und Spohrer 2006), (Alter 2010), (Krafzig, Banke et al. 2005b), (Baida, Gordijn et al. 2004b), (Bartsch 2010), (Chituc 2010b)
Ist ein Paket von Funktionalitäten mit messbaren Service Levels und kann als in sich geschlossene Einheit betrachtet werden	(Alarcon, Wilde et al. 2010), (Barakat, Miles et al. 2012), (Faravelon, Chollet et al. 2012), (Löffler 2011), (Kleiner 2013), (Klatt, Brosch et al. 2012), (Wagner, Ishikawa et al. 2012)
Beinhaltet unterschiedlich große Mengen an Logik / Komplexität	(Fernández, Tedeschi et al. 2012), (Klatt, Brosch et al. 2012), (Löffler 2011), (Meiländer, Bucchiarone et al. 2012), (Melzer 2010), (Wagner, Ishikawa et al. 2012), (Neto 2012)
Wird mittels Interaktion zwischen unterschiedlichen Entitäten erbracht	(Gustafsson und Johnson 2003), (Berardi, Calvanese et al. 2003), (Vargo und Lusch 2004), (Lovelock und Gummesson 2004), (Chesbrough und Spohrer 2006), (Fitzsimmons und Fitzsimmons 2001), (Sampson 2001), (Alter 2010), (Nayak, Nigam et al. 2006), (Baida, Gordijn et al. 2004b), (Becker, Niehaves et al. 2015), (Zirpins, Feuerlicht et al. 2010), (Löffler 2011), (Smit und Stroulia 2010), (Roy, Suleiman et al. 2010), (Alarcon, Wilde et al. 2010), (Hull 2013), (Kattepur, Georgantas et al. 2013), (Dasgupta, Shrinivasan et al. 2013), (Shrinivasan, Dasgupta et al. 2012), (Hastings und Saperstein 2014), (Jongmans, Santini et al. 2012), (Preda, Gabbrielli et al. 2012), (Yu, Han et al. 2013), (Juszczuk, Schall et al. 2010), (Zolnowski und Böhmann 2013), (Cardoso, Pedrinaci et al. 2012), (Neto 2012)

Bringt einen Mehrwert für eine Entität	(Grönroos 2000), (Vargo und Lusch 2004), (Lovelock und Gummesson 2004), (Alter 2010), (Baida, Gordijn et al. 2004b), (Cardoso, Pedrinaci et al. 2012), (Wang, Wang et al. 2013), (Hastings und Saperstein 2014), (Zolnowski und Böhmann 2013)
Nutzt unterschiedliche Ressourcen wie Wissen, Fähigkeiten, Informationen, aber auch physische Güter, Technologien oder Infrastrukturen	(Grönroos 2000), (Gustafsson und Johnson 2003), (Lovelock und Gummesson 2004), (Tapscott und Ticoll 2003) in (Chesbrough und Spohrer 2006), (Chesbrough und Spohrer 2006), (Alter 2010), (Cardoso, Pedrinaci et al. 2012), (Löffler 2011), (Hastings und Saperstein 2014), (Hradilak 2011), (Meyer, Thieme et al. 2012)
Wird via elektronischer Netzwerke angeboten und erbracht	(Rust und Kannan 2002), (Piccinelli, Emmerich et al. 2001), (Berardi, Calvanese et al. 2003), (Alter 2010), (Baida, Gordijn et al. 2004b), (Bianchini, De Antonellis et al. 2006), (Zirpins, Feuerlicht et al. 2010)
Unterstützt und basiert auf Informationsflüssen bzw. auf Technologie	(Lovelock und Gummesson 2004), (Rowley 2006), (Bitner, Brown et al. 2000), (Rust und Kannan 2003), (Hradilak 2011), (Roy, Suleiman et al. 2010), (Saini, Nanchen et al. 2013), (Chituc 2010b)
Besteht aus autonomen und plattformunabhängigen Entitäten, die innerhalb einer komplexen IT-Landschaft interagieren	(Zirpins, Feuerlicht et al. 2010), (Moha, Palma et al. 2012), (Ivanović, Carro et al. 2012), (Barakat, Miles et al. 2012), (Fernández, Tedeschi et al. 2012), (Preda, Gabbrielli et al. 2012), (Yu, Han et al. 2013), (Chituc 2010b), (Chituc 2010a)
Ist flexibel, wiederverwendbar, prozessgesteuert, skalierbar, lose gekoppelt und besteht aus einer Komposition verschiedenster Dienste	(Juszczyk, Schall et al. 2010), (Alarcon, Wilde et al. 2010), (Ye, Bouguettaya et al. 2013), (Ghezzi, Pezzé et al. 2013), (Björkqvist, Spicuglia et al. 2013), (Moha, Palma et al. 2012), (Kattepur, Georgantas et al. 2013), (Ivanović, Carro et al. 2012), (Saini, Nanchen et al. 2013), (Jongmans, Santini et al. 2012), (Barakat, Miles et al. 2012), (Demchenko, Ngo et al. 2012), (Preda, Gabbrielli et al. 2012), (Fernández, Tedeschi et al. 2012), (Wagner, Ishikawa et al. 2012), (Melzer 2010), (Yu, Han et al. 2013), (Upadhyaya, Zou et al. 2013), (Moustafa und Zhang 2013), (Shrinivasan, Dasgupta et al. 2012), (Oster, Ali et al. 2012), (Faravelon, Chollet et al. 2012), (Klatt, Brosch et al. 2012), (Lago und Razavian 2012), (Neto 2012)

Tabelle 3-1: Serviceeigenschaften

Die Tabelle beinhaltet dabei nur solche Eigenschaften aus Definitionen, die von mehreren Autoren geteilt werden. Darüber hinaus lassen sich je nach Forschungsrichtung noch weitere Eigenschaften aufzeigen. Die letzte Eigenschaft der Tabelle kann als eine Art Zusammenfassung aller Eigenschaften aufgefasst werden, da darin einige Bestandteile der oberen wiederverwendet werden. Allerdings liefert sie auch keine besondere Trennschärfe zur Unterscheidung.

Dementsprechend schwer fällt eine allgemeingültige, zugleich aber präzise Definition. Services können demnach entweder durch Menschen oder Maschinen erbracht und konsumiert werden, repräsentieren abgeschlossene Arbeitseinheiten in Form von einzelnen Aktivitäten oder Prozessen und greifen für die Ausführung mehr oder weniger stark auf technologische Komponenten (Netzwerke, Softwareartefakte, Architekturen) zurück. Dabei sind sowohl die Extrem- also auch Zwischenpositionen bei der Erbringung möglich. Relevant

ist jedoch die umfassende Betrachtung aus verschiedenen Blickwinkeln, um alle notwendigen Eigenschaften zu erfassen. Im Rahmen der Arbeit gilt die nachfolgende Definition:

Definition 3-1: Service

Unter einem Service wird der Gegenstand einer Dienstleistung analog zu einem Sachgut verstanden. Hierbei handelt es sich um eine immaterielle, u.U. jedoch materielle Gegenstände beeinflussende Handlung einer oder mehrerer Beteiligter. Ein Service ermöglicht den Zugriff auf Fähigkeiten und Ressourcen des Anbieters über eine spezifizierte Schnittstelle und wird reproduzierbar, gemäß seiner (öffentlichen) Beschreibung, in einer einheitlichen Art und Weise erbracht.

Diese Definition eines Service ist die Quintessenz aus dem Versuch der Spezifikation und Einordnung von Servicemerkmalen aus der Literatur. Sie zeigt aber zugleich die Schwierigkeiten einer umfassenden Darstellung auf. Es kann keine trennscharfe Definition angegeben werden, mittels derer eindeutig eine „Handlung“ als Service ausgewiesen werden kann oder nicht. Für diese Arbeit ist die Auseinandersetzung mit unterschiedlichen Definitionen umso wichtiger, als dass dadurch die Vielschichtigkeit von Services belegt werden kann. Je nach Anwendungsdomäne werden unterschiedliche Aspekte eines Service stärker in den Vordergrund gerückt und modelliert. Dies zeigt sich auch in dem nachfolgenden Abschnitt über Modellierungsansätze.

3.2 Spezifische Servicemodellierungsansätze

Servicemodelle dienen dem Zweck der Einteilung und Strukturierung (elementarer) Bestandteile von Services und besitzen je nach Domäne, für die sie entwickelt werden, einen unterschiedlichen Fokus (z. B. ökonomisch, physisch, informationstechnisch, architektonisch). Analog zur Bildung von Ontologien (Fensel 2003; Gómez-Pérez, Fernández-López et al. 2004; Staab und Studer 2009) werden spezifische Modelle zur Konzeptualisierung der jeweiligen Domäne verwendet, d.h. wesentliche Elemente werden zueinander mit Hilfe spezialisierter Verbindungen in Beziehung gesetzt.

Es zeigt sich, dass dieselbe Problematik des vorhergehenden Abschnitts auch auf die Modellierungsansätze selbst zutreffend ist. Alle betrachteten Ansätze sind entweder unstrukturiert, semi-strukturiert oder strukturiert und besitzen entweder eine textuelle oder grafische Notation. Die zugrundeliegenden Formalismen (Metamodelle bzw. Ontologien) basieren häufig auf Standards (z. B. formale Beschreibungssprachen), einige Modellierungsansätze repräsentieren jedoch vollständige Eigenentwicklungen der Autoren. Da in diesen Fällen

der zugrundeliegende Formalismus nicht vollständig spezifiziert ist, sind solche Ansätze für eine systematische Weiterverarbeitung schwer zu integrieren.

Die existierenden Arbeiten beziehen sich aufgrund der Komplexität zumeist auf einzelne Aspekte von Services und können als komplementär bzw. substituierend angesehen werden. Beispielhaft seien die Projekte „Rebeca“ (Thomas und Scheer 2003) und „Service Concepts“ (Karni und Kaner 2006) genannt. Gezielt wurden auch Modellierungsansätze mit einem Anwendungsbereich in der Logistik gesucht. Jedoch hat sich gezeigt, dass solche Ansätze insbesondere für den Bereich des Supply Chain Managements (SCM) entwickelt wurden, wodurch aufgrund einer vollkommen anderen Zielstellung (Arns, Fischer et al. 2002; Savaskan, Bhattacharya et al. 2004) die Ansätze für die Analyse nicht geeignet sind. Im Gegensatz zu hier vorgestellten Ansätzen der Servicemodellierung besteht der Kern von SCM-Ansätzen darin, aus der Perspektive des Produzenten (Verlader) oder des Handels Möglichkeiten aufzuzeigen, wie Waren-, Informations- und Geldströme möglichst optimal über ein Netzwerk verteilt werden (Min und Zhou 2002). Ansätze, die SCM und Service Management verbinden, besitzen ebenfalls keine ausgeprägtes Servicemodell, sondern wenden Erkenntnisse des Supply Chain Managements auf die Nutzung von Services in Unternehmensnetzwerken an (Bossert und Willems 2007; Dorne, Lesaint et al. 2008).

Die untersuchten Ansätze zeigen zunächst ein breites Bild an zu modellierenden Eigenschaften, wobei deren Relevanz teils stark unterschiedlich ausfällt. In einer ersten groben Übersicht lassen sich die Modellierungsansätze durch eines der drei Gegensatzpaare grob beschreiben:

1. Services, die physische Entitäten manipulieren (z. B. Transport oder Verpackung von Gütern) bzw. in Form von Softwareprodukten.
2. Services als Konsumgegenstand und Verkaufsobjekt bzw. Fokus auf Erbringung.
3. Fokus auf Ablauf (z. B. als Geschäftsprozess) bzw. Fokus auf Aufbau und Struktur (z. B. Softwarearchitektur / SOA).

Darüber hinaus kann die Zielstellung bei der Entwicklung der Modellansätze weiter präzisiert werden und folgende Ziele abgeleitet werden:

1. Interaktionsmodellierung: Meist im Sinne einer Kunde- und Lieferanteninteraktion.
2. Domänenmodellierung und Konzeptualisierung: Darstellung wichtiger Elemente im Kontext sowie deren Beziehungen untereinander. Eine spezielle Ausprägung ist die Angabe eines detaillierten Informationsmodells.
3. Architektur- und Softwaremodelle: Umsetzung einer spezifischen Servicedefinition mit Hilfe von Architekturen oder Standards. Außerdem Beschreibung wichtiger Softwarekomponenten zur Umsetzung und Bereitstellung von Services.

Tabelle 3-2 enthält eine Liste der untersuchten Modelle mit ihrem Autor, Titel und dem Zweck ihrer Erstellung.

Servicemodell	Titel	Zweck
(Garschhammer, Hauck et al. 2001)	Towards generic service management concepts - A service model based approach	Geschäftsprozesse, Geschäftswert
(Akkermans, Baida et al. 2004)	Value Webs: Using Ontologies to Bundle Real-World Services	Realwelt-Services
(Baida, Gordijn et al. 2004a)	Ontology-based analysis of e-service bundles for networked enterprises	Geschäftsprozesse, Geschäftswert
(Jones 2005)	Toward an Acceptable Definition of Service	Standards für Services
(Simon, Bansal et al. 2005)	A Universal Service Semantics Description Language	Semantische Beschreibung elektronischer Services
(Erradi, Anand et al. 2006b)	SOAF: An Architectural Framework, for Service Definition and Realization	SOA Framework
(Nitin, Anil et al. 2006) (Nayak und Nigam 2007)	Concepts for Service-Oriented Business Thinking	Geschäftsprozesse, SOA
(Uribarren, Parra et al. 2006)	Service Oriented Pervasive Applications Based On Interoperable Middleware	Realisierung einer Middleware für Services
(Walkerdine, Hutchinson et al. 2007)	A Faceted Approach to Service Specification	Ableiten von Kategorien zur Beschreibung von Services
(López-Sanz, Acuña et al. 2008)	Modelling of Service-Oriented Architectures with UML	MDA-basierter Ansatz zur Spezifikation einer SOA mittels UML-Profilen
(Weigand, Johannesson et al. 2009)	Value-Based Service Modeling and Design: Toward a Unified View of Services	Geschäftswert, Typisierung von Services
(Böttcher 2009)	Architektur integrierter Dienstleistungssysteme - Konzeption, Metamodell und technikraumspezifische Konkretisierung	Systematische, 3-dimensionale Serviceentwicklung
(Cardoso, Barros et al. 2010)	Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments	Beschreibungssprache für Services in Marktplätzen; Geschäftsprozesse
(Ferrario, Guarino et al. 2010)	Towards an Ontological Foundation of Services Science: The General Service Model	Geschäftsprozesse, Geschäftswert

Tabelle 3-2: Ansätze zur Servicemodellierung

Vor einer detaillierteren Auseinandersetzung mit der Analyse obiger Modellierungsansätze wird zunächst eine auf den Beobachtungen gestützte Definition eines Servicemodells genannt:

Definition 3-2: Servicemodell

Ein Servicemodell ist ein domänenspezifisches Modell mit begrenztem Umfang zum Zwecke der Modellierung einer spezifischen Perspektive auf einen Service. Ein zugrundeliegendes Metamodell liefert den notwendigen Formalisierungsgrad.

Aus den oben genannten Ansätzen lassen sich verschiedene Schlussfolgerungen für die Servicemodellierung ziehen. So zeigt sich, dass je spezifischer der Einsatzzweck eines Modells ist, umso schärfer ist auch der jeweilige Anwendungsbereich definiert. D.h. je konkreter ein Modell auf eine bestimmte Domäne zugeschnitten ist, umso spezifischer und umfangreicher ist das verwendete Vokabular und umso detaillierter sind die modellierten Beziehungen zwischen den Elementen. Enthalten beispielsweise die Modelle von (Nitin, Anil et al. 2006; Nayak und Nigam 2007) nur wenige eher abstrakte Servicebegriffe und setzen diese zueinander in Beziehung, so ist der Ansatz von (Cardoso, Barros et al. 2010) wesentlich umfangreicher in der Zahl verwendeter Elemente und deren Beziehungen. Der Ansatz von Cardoso et al. zeichnet sich im Vergleich auch durch eine stärkere Segmentierung der Elemente aus. Beiden Ansätzen gemein ist dennoch das Konzept des Separation of Concerns. Im Falle des kleineren, abstrakteren Modells wurde aus Sicht der Autoren nur eine Perspektive behandelt, während im umfassenderen Modell mehrere Perspektiven durch eine Strukturierung in mehr oder minder unabhängige Teile enthalten sind.

Es zeigt sich, dass durch das Hinzufügen mehrerer darzustellender Perspektiven die Komplexität der Modelle sehr schnell ansteigt. Dies lässt sich dadurch begründen, dass Eigenschaften in verschiedenen Perspektiven relevant sind, so dass sich in einem solchen Modell Abhängigkeiten zwischen voneinander abgegrenzten Perspektiven ergeben. Die naheliegende Lösung der Aufteilung und Separierung der Beschreibung in voneinander getrennte Teilmodelle verschiebt dabei das Problem der existierenden Abhängigkeiten jedoch lediglich. Die Problematik wird verschärft, insofern beschriebene Eigenschaften zwischen Modellen konsistent gehalten werden müssen. Ein einzelnes Modell, das alle relevanten Perspektiven beinhaltet ist dennoch nicht praktikabel. Zum einen ist aufgrund der unklaren Servicedefinition die Zahl der Perspektiven variabel. Müssen zu einem späteren Zeitpunkt neue Perspektiven in ein bereits umfangreiches Servicemodell integriert werden, steigt die Komplexität aufgrund der Abhängigkeiten enorm. Hinzukommt, dass bedingt durch die unterschiedlichen Perspektiven der eigentliche Modellzweck aufgeweicht wird, bspw. durch Vermischung struktureller, prozessualer oder beschreibender Aspekte und Eigenschaften. Somit nimmt letztlich die Qualität des Gesamtmodells ab. Stattdessen wird eine strikte Orientierung am Modellzweck empfohlen. Konsequenterweise kann ein Service

dann nur durch eine Menge, sich überlappender Modelle vollständig beschrieben werden. Die Herausforderung besteht dann in der Konsistenzwahrung sämtlicher Teilmodelle bezogen sowohl auf das Einzelmodell wie auch auf das gesamte Konstrukt der zusammenhängenden Modelle. Gelingt dies entsteht dadurch ein im Sinne des Modellierers vollständiges Bild eines Service. Damit wird auch der Kern dieser Arbeit deutlich hervorgehoben. Ziel der Arbeit ist es, ein konsistentes Gesamtbild aus unterschiedlichen Teilmodellen eines Service zusammenzusetzen, wobei sich die Einzelteile je nach Anwendungskontext unterscheiden können, d.h. unterschiedliche Perspektiven auf die zu modellierenden Services relevant werden.

4 Verwandte Ansätze

Dieser Abschnitt behandelt Ansätze, die ähnlich zur modellgetriebenen Servicebeschreibung einen konzeptuellen Rahmen für die ganzheitliche Servicemodellierung darstellen. Im Gegensatz zu spezifischen Servicemodellierungsansätzen (s. 3.2) liegt der Fokus hier auf einer multiperspektivischen Betrachtungsweise, d.h. es werden mehrere Modelle mit unterschiedlichen Betrachtungswinkeln miteinander zu einem ganzheitlichen Bild verbunden. Über die einfache Servicemodellierung, also die Konzeptualisierung elementarer Servicebegriffe, hinaus existieren teils umfassende, sehr aktuelle Arbeiten, die sich mit der mehrdimensionalen Modellierung auseinandersetzen, wodurch die Relevanz des Themas noch einmal hervorgehoben wird. Die Vielzahl und die unterschiedlichen Herangehensweisen an diese Problemstellung ergeben sich dabei aus den in 3.1 dargestellten, heterogenen Servicedefinitionen und damit aus der noch immer unpräzisen, weil nicht eindeutigen, Begriffsbestimmung für Services, Diensten oder Dienstleistungen. Dementsprechend kann trotz multiperspektivischer Betrachtung eine Vereinheitlichung der Bemühungen nur schwer gelingen, da die Wahl der Perspektiven sehr von der ursprünglichen Intention des Modellierers und damit vom Wissenschaftsgebiet abhängt.

4.1 Vergleichsbasis

Für den Vergleich der nachfolgenden Ansätze wird zunächst ein Katalog an spezifischen Eigenschaften vorgestellt, mit dessen Hilfe die Ansätze mit der später vorgestellten modellgetriebenen Servicebeschreibung vergleichbar gemacht werden können. Sämtliche Eigenschaften lassen sich in die Kategorien *Modellierung*, *Rollenmodell* und *Anwendung* einteilen. Die Kategorien leiten sich aus der Motivation für die Entwicklung der modellgetriebenen Servicebeschreibung ab, mit dem Ziel ein Rahmenwerk für die im Sinne der Nutzer transparente Informationsmodellierung zu erstellen. Um andere Ansätze mit diesem

hier vergleichen zu können, ist es daher notwendig, dass sich solche Ansätze grundsätzlich mit der Servicemodellierung auseinandersetzen.

Die Modellierungskategorie umfasst solche Eigenschaften, die die Konzeption und den internen Aufbau des Ansatzes betreffen. Ansätze können dabei unterschiedliche Arten der Modellierung nutzen, um Services zu modellieren. Relevant ist ebenfalls, ob mehrere Perspektiven zum Einsatz kommen, da nur solche Ansätze mit der modellgetriebenen Servicebeschreibung vergleichbar sind, die mehrere Perspektiven nutzen. Ist dies nicht der Fall, werden sie den Servicemodellierungsansätzen zugeordnet (vgl. 3.2). Die Frage nach dem Einsatz standardisierter Modelle ist notwendig, da viele Ansätze eigene Sprachen zur Beschreibung entwickeln, wodurch einerseits ein erhöhter Lernbedarf bei der Nutzung des Ansatzes und andererseits ein erhöhter Arbeitsaufwand bei der Übertragung in die jeweilige Sprache entstehen. Ähnlich gelagert ist die Erweiterbarkeit. Sind die Ansätze in der Lage bereits bestehende Modelle zu integrieren, sinkt der Aufwand für die Nachbearbeitung durch den Nutzer. Die Erweiterbarkeit zielt dabei explizit auf die Flexibilität des Ansatzes ab, weitere (standardisierte) Modelle in das Gesamtbild einzufügen. Die Informati-onstransparenz als Eigenschaft eines Ansatzes gibt Auskunft darüber, ob unterschiedliche Perspektiven so miteinander verbunden sind, dass Informationen zwischen Modellen übertragen werden können und ob die so verbundenen Modelle Querbeziehungen untereinander aufweisen. Die letzte Eigenschaft der Kategorie Modellierung kennzeichnet schließlich die Anwendungsnähe des Ansatzes. Maßgebend hierfür ist die Beurteilung der konkret eingesetzten Modelle und Konzepte. Sind diese eher generisch bzw. reflektieren die Modelle nur grundsätzliche Tatsachen, so ist von einem hohen Abstraktionsniveau und damit von einer geringeren Anwendbarkeit auszugehen.

Eigenschaften der Kategorie Rollenmodell beziehen sich sowohl auf das im Ansatz enthaltene Rollenmodell als auch auf die unterschiedlichen Anwender eines Ansatzes etwa im Unternehmenskontext. Aus interner Sicht beschreibt die Eigenschaft Nutzergruppen die Zahl involvierter Akteure, die im Zuge der Modellierung nötig sind, um den Ansatz vollständig umsetzen zu können. Beispiele hierfür wären ein Domänenarchitekt oder fachliche Modellierer, die einzelne Modelle im Zuge ihrer Tätigkeiten kreieren. Im Gegensatz dazu wird mit der Eigenschaft Rollenmodell spezifiziert, ob ein Ansatz ein spezifisches Rollenmodell beinhaltet, d.h. ob für dessen Umsetzung dezidierte Rollen mit spezifischen Aufgaben notwendig sind. Schließlich wird über die Zielgruppen der Kreis von Personen beschrieben, die direkte Nutznießer der resultierenden Artefakte sind. Beispiele für Zielgruppen sind etwa Entscheidungsträger oder operatives Personal, wobei letztere ihre Tätigkeiten auf die erstellten Artefakte stützen können.

Die Kategorie Anwendung beinhaltet schließlich solche Eigenschaften, die die grundsätzliche Anwendbarkeit eines Ansatzes betreffen und beschreiben unter welchen Voraussetzungen bzw. mit welchen Einschränkungen ein Ansatz angewendet werden kann. So beinhaltet das Anwendungsgebiet den konkreten Einsatzzweck, für den der Ansatz entwickelt wurde. Das Anwendungsgebiet kann dazu herangezogen werden, die Zahl der Perspektiven einzugrenzen, indem per Ausschlussprinzip Perspektiven als nicht relevant unberücksichtigt bleiben können. Das Anwendungsgebiet ist jedoch in keiner Weise eingeschränkt, so dass es dem Entwickler des Ansatzes überlassen ist, wie weit dieser Anwendung finden kann. Anwendbarkeit und Anpassbarkeit sind Eigenschaften, die mit dem Abstraktionsniveau bzw. mit der Erweiterbarkeit der Modellierungskategorie verbunden sind. Je abstrakter ein Ansatz, umso schlechter kann die grundlegende Anwendbarkeit sein, da zunächst die allgemeinen Konzepte und Sachverhalte auf ein konkretes Anwendungsszenario angewendet werden müssen. Allerdings ist die Beziehung zwischen Abstraktionsniveau und Anwendbarkeit nicht hinreichend, da die generelle Anwendbarkeit nicht per se durch höhere Abstraktion eingeschränkt wird. Durch ein höheres Abstraktionsniveau kann auch eine breitere Anwendbarkeit gegeben sein, da durch die allgemeinere Modellierung mehrere Anwendungsszenarien denkbar sind. Die Anwendbarkeit stellt jedoch ein Maß dafür da, inwieweit ein Ansatz praxistauglich ist. Die Anpassbarkeit kennzeichnet die Flexibilität eines Ansatzes, ähnlich wie die Erweiterbarkeit. Anpassbarkeit bezieht sich jedoch zum einen darauf, ob bestehende Modelle erweitert bzw. ergänzt werden können und nicht nur, ob weitere Modelle als Ganzes dem Ansatz hinzugefügt werden können. Zum anderen ist die Anpassbarkeit auch ein Maß dafür, wie robust ein Ansatz z. B. bei Änderung des zugrundeliegenden Anwendungsgebietes ist, d.h. wie stark ein Ansatz an das ursprüngliche Anwendungsgebiet gekoppelt ist. Die nachstehende Tabelle 4-1 fasst die Eigenschaften zusammen:

Kategorie / Eigenschaft	Beschreibung
Modellierung	
Modellierungsart	Metamodell oder Ontologie-basiert
Zahl vorhandener Perspektiven	Betrachtete Serviceaspekte
Einsatz standardisierter Modelle	Verwendung standardisierter Modellierungssprachen
Erweiterbarkeit (Modelle)	Möglichkeit der Integration neuer Modelle in den bestehenden Ansatz
Informationstransparenz	Konsistenzwahrung unter der Wiederverwendung von Informationen in unterschiedlichen Perspektiven
Abstraktionsniveau	Anwendungsnähe des Ansatzes

Rollenmodell	
Nutzergruppen	Zahl involvierter Akteure bzw. internes Rollenmodell des Ansatzes
Rollenmodell	Rollenmodell für den Anwendungsfall des Ansatzes vorhanden
Zielgruppen	Zahl der Zielgruppen des Ansatzes
Anwendung	
Anwendungsgebiet	Anwendungsdomäne des Ansatzes, Einsatzzwecke
Anwendbarkeit	Grundsätzliche Anwendbarkeit oder reines Konzept
Anpassbarkeit	Flexibilität im Umgang mit Änderungen bzw. Erweiterungen des Ansatzes

Tabelle 4-1: Vergleichsmerkmale zur Abgrenzung

Die zu überprüfenden Eigenschaften sind so definiert, dass sie mit Ausnahme des Anwendungsgebietes einer *Kardinal-* (Perspektivenzahl, Nutzergruppen, Zielgruppen), *Ordinal-* (Abstraktion (Hoch / Mittel / Gering)) oder *Nominalskala* ((Ja/Nein): Einsatz standardisierter Modelle, Informationstransparenz, Erweiterbarkeit, Rollenmodell, Anwendbarkeit, Anpassbarkeit; (M/O): Modellierungsart) folgen.

4.2 Ansätze

Nachfolgend werden sieben Ansätze in der Reihenfolge ihrer Veröffentlichung ausgehend von der Neuesten bis hin zur Ältesten vorgestellt. Die Identifizierung geeigneter Kandidaten fand im Rahmen einer erweiterten Recherche zu den in 3.2 vorgestellten, rein konzeptuellen Servicemodellen statt. Im Folgenden werden für jeden Ansatz die wesentlichen Grundzüge kurz herausgearbeitet und im Anschluss eine Bewertung anhand des in 4.1 erarbeiteten Kriterienkatalogs durchgeführt.

(Nardi, de Almeida Falbo et al. 2013): Towards a Commitment-based Reference Ontology for Services

Nardi et al. verwenden mehrere Ontologien zur Beschreibung von Services entlang eines dreiteiligen Lebenszyklus, bestehend aus den Phasen „Service Offer“, „Service Negotiation“ und „Service Delivery“. Dabei wird für jede Phase eine spezifische Ontologie vorgestellt, mit deren Hilfe die wesentlichen Elemente zueinander in Beziehung gesetzt werden. Ausgangsbasis für alle erstellten Ontologien ist eine so genannte Unified Foundation Ontology (UFO). Darüber hinaus sind jedoch keine weiteren Modelle, etwa Rollenmodelle o.ä. vorhanden. Eine Anwendung der entwickelten Ontologien findet mit Hilfe von vier Perspektiven statt: Service als Co-creation, Service als Beschreibung von Fähigkeiten und Anwendung von Kompetenzen, Service als Interaktion und technische Services. Allerdings bleiben die Autoren eine detaillierte Anwendung schuldig und beschränken sich auf die Beschreibung der jeweiligen Perspektive. Außerdem legen die Autoren den Fokus auf die

konzeptuelle Erfassung der Servicedomäne, ohne jedoch eine Anwendungsmöglichkeit ihrer Arbeit in einem konkreten Kontext aufzuzeigen. Da die verwendeten Elemente in den Ontologien Services auf einem recht hohen, abstrakten Niveau beschreiben ist die Anwendbarkeit nicht gegeben. Eine Erweiterung einer oder mehrerer der vorgestellten Ontologien sollte jedoch problemlos möglich sein.

(Berre, Lew et al. 2013): Service Innovation and Service Realisation with VDML and ServiceML

Ähnlich zu der hier vorliegenden Arbeit kombinieren Berre et al. serviceorientierte Konzepte mit Modellen und entwickeln darauf aufbauend ein modellbasiertes Architekturbild für Unternehmen. Das Design der elementaren Merkmale zeigt eine Matrix-Struktur, vergleichbar mit dem Zachman-Framework (Zachman 1987). Unterschieden werden Sichten (Interaktion, Struktur, Funktion, Koordination, Information, nicht-funktionale Anforderungen (EFA)) und Abstraktionslevel (Anforderungen, Design, Implementierung, Infrastruktur). Jeder Matrixzelle werden spezifische Modellierungssprachen zugeordnet, mit deren Hilfe eine Sicht auf einer bestimmten Abstraktionsstufe modelliert werden kann. Kern der Arbeit sind eine Plattform für die Serviceentwicklung mit Hilfe domänenspezifischer Sprachen und die AT-ONE (Actors, Touchpoints, Offerings, Needs, Experiences) genannte Methode sowie die zur Servicebeschreibung verwendeten Sprachen VDML (Value Delivery Modeling Language) und ServiceML. ServiceML ist eine eigens kreierte Sprache, jedoch keine vollständige Eigenentwicklung, sondern verbindet bereits existierende Sprachen, um ein vollständiges Bild entlang der AT-ONE relevanten Merkmale zu erhalten. Der Ansatz besitzt inhärent eine hohe Modellierungsvielfalt aufgrund der vielen zum Einsatz kommenden Sprachen. Nachteilhaft im Sinne dieser Arbeit ist jedoch der zu starke Fokus auf den Kunden. Dadurch sind die im Ansatz verwendeten Konzepte und Beziehungen zu stark geprägt vom konkreten Einsatzgebiet. Der Ansatz beinhaltet ausschließlich Perspektiven, die die Nutzersichtweise von Services beschreiben. Auch sind die z. B. durch AT-ONE vorgegeben Sichten zu stark auf den Kunden (Wünsche, Erfahrungen, Interaktionspunkte) ausgerichtet.

(Steen, Iacop et al. 2012): Agile Service Development - Combining Adaptive methods and flexible solutions

Das Buch beschreibt verschiedene Sichtweisen rund um die agile Entwicklung und Nutzung von Services. Für die vorliegende Arbeit ist insbesondere das Kapitel 4 (Seiten 59-93) relevant, da dort der Fokus auf die Modellierung von Services gelegt wird. Für die Autoren spielen Modelle bei der Serviceentwicklung eine gewichtige Rolle, da mit ihnen der Prozess der Serviceentwicklung beschleunigt werden kann. In erster Linie dienen Modelle

zur Kommunikation mit allen beteiligten Stakeholdern. Der Kern der Arbeit besteht aus dem ASD (Agile Service Development) genannten Rahmenwerk und einer zugehörigen Modellierungsmethode. Wie schon in dem zuvor vorgestellten Ansatz ist auch hier die Nähe z. B. zum Zachman Framework durch die Verwendung einer 2-dimensionalen Struktur gegeben. Analog werden hier ebenfalls Sichten bzw. Aspekte (Interaktion, Struktur, Funktionalität, Koordination, Entscheidungen) und Abstraktionslevel (Anforderungen (Warum wird der Service benötigt), Design (Was macht den Service aus), Implementierung (Wie wird ein Service umgesetzt), Infrastruktur (Wer verwendet den Service)) eingesetzt. Daraus resultiert eine Matrix-hafte Darstellung, bei der in jeder Zelle zu modellierende Eigenschaften eines Services enthalten sind und diesen Eigenschaften geeignete Modellierungssprachen zugeordnet werden. Außerdem werden für die einzelnen Eigenschaften über Metamodelle notwendige Konzepte eingeführt, die im Laufe des Modellierungsprozesses einbezogen werden müssen. Dies mündet in einem integrativen Metamodell, welches die einzelnen Teilmodelle miteinander verbindet und strikte Vorgaben an die Modellierer enthält. Das erklärte Ziel ist die Verringerung des Aufwands bei der Serviceentwicklung von der Anforderungsaufnahme bis hin zur Ausführung und die Schaffung eines Business-IT-Alignments durch das Zusammenbringen der unterschiedlichen Stakeholder mit Hilfe von Modellen. Mit diesen soll ein gemeinsames Verständnis zwischen allen Beteiligten geschaffen werden. Obwohl der Ansatz von Steen et al. damit sehr dem hier vorgestellten Ansatz ähnelt, gibt es Kritikpunkte. Zum einen wird ein integratives Metamodell entwickelt, mit dessen Hilfe einzelne Sichten zueinander in Beziehung gesetzt werden können. Diese Herangehensweise ist vergleichbar mit einigen Ansätzen der Referenzmodellierung (s. Exkurs I), bei denen ein Ordnungsrahmen durch ein zentrales Modell geschaffen werden soll. Allerdings hat diese Vorgehensweise gerade im Zusammenhang mit der Servicemodellierung den Nachteil, dass dadurch zu stark auf den Charakter der Servicebeschreibung eingewirkt wird und dass dadurch evtl. zusätzliche Sichten nicht in Betracht gezogen werden. Zum anderen ist die strikte Integration eines Domänenmodells als nachteilig zu bewerten. Die Eindeutigkeit der Domänenbeschreibung ist in Anbetracht der Vielzahl an Stakeholdern nicht mehr zwingend eindeutig, da unterschiedliche Sichtweisen teils unterschiedliche Interpretationen auf denselben Sachverhalt erlauben.

(Kett, Voigt et al. 2008; Scheithauer, Voigt et al. 2009): Integrated service engineering workbench: service engineering for digital ecosystems

Der Ansatz von Kett et al. bzw. Scheithauer et al. (ISE) ist auf die Serviceentwicklung in einer dynamischen Umgebung mit hohen Unsicherheiten und gemeinschaftlich mit anderen Unternehmen ausgelegt. Sie verwenden einen modellgetriebenen Ansatz und beziehen

verschiedene Stakeholder mit ein. Die Motivation für die Arbeit liegt in der Tertiärisierung bzw. im Übergang von der Industrie- in eine Servicegesellschaft, in einer Globalisierung und in einem stärkeren Konsumverhalten, wodurch existierende Softwareentwicklungsmethodiken aufgrund der hohen Dynamik als unzureichend beschrieben werden. Insbesondere wird die mangelnde Interdisziplinarität vieler Ansätze kritisiert. Eine Grundlage bildet wie bei den vorhergehenden Ansätzen die Matrix-hafte Darstellung zweier Dimensionen analog zum Zachman Framework zur ganzheitlichen Gestaltung. Ebenfalls werden wie bei den vorherigen Ansätzen spezifische Serviceaspekte (Service-, Prozess-, Akteur-, Regel- und Datenbeschreibung) mit Hilfe unterschiedliche Abstraktionsschichten (auch Serviceperspektiven genannt: strategische, konzeptuelle, logische, technische Perspektive) dargestellt. Jedes Matrixelement steht stellvertretend für ein Metamodell, Aktivitäten, Werkzeuge und einer Notation. Alle Modelle einer Perspektive können außerdem zu einem Modell zusammengefasst werden. Ziel ist es, mit Hilfe von Werkzeugen auf Eclipse-Basis Modelltransformationen sowohl in vertikaler wie horizontaler Richtung anzugeben und so zwischen den Matrixelementen mit Hilfe technischer Unterstützung wechseln zu können. Es existiert jedoch keine Vorgabe hinsichtlich der Reihenfolge, sehr wohl aber hinsichtlich der zu nutzenden Modellierungssprache. Daraus resultiert eine enorme Vielzahl notwendiger Editoren und zu definierender Modelltransformationen (derzeit 20 Editoren). In Bezug zur modellgetriebenen Servicebeschreibung besteht eine große Nähe bei der Nutzung Eclipse-basierter Werkzeuge und in der Anwendung von Modelltransformationen, um Informationen zwischen vorgegebenen Modellen austauschen zu können. Allerdings ist auch hier der Kritikpunkt, dass die Vorgaben zu strikt sind. Dies wird umso mehr klar, wenn man die Zahl der notwendigen Modelltransformationen betrachtet, die notwendigerweise vorgegeben werden müssen. Zum einen ist die Zahl der zu definierenden Transformationen recht hoch, um den Gesamtansatz vollständig zu unterstützen und zum anderen sind Transformationen zwischen allen Sprachen nicht zwingend durchführbar, so dass entweder Einschränkungen in der Wahl der Sprache oder in der Anwendbarkeit des Ansatzes vorliegen. (Quartel, Steen et al. 2007): *COSMO: A conceptual framework for service modeling and refinement*

Wie bereits bei den zuvor dargestellten Ansätzen, verwendet auch der Ansatz von Quartel et al. eine Matrix, und damit zwei Dimensionen zur Darstellung von Services. Es werden Aspekte (Struktur, Verhalten, Informationen, Ziele, Qualität) und Abstraktionsstufen (Single Interaction, Choreography, Orchestration) als Dimensionen verwendet. Darüber hinaus besitzt der Ansatz Gemeinsamkeiten mit dem Ansatz von Steen et al., da für die einzelnen Serviceaspekte Metamodelle angegeben werden, mit deren Hilfe die wesentlichen Konzep-

te genannt und in Beziehung zueinander gesetzt werden. Außerdem entwickelten Quartel et al. eine spezifische Notation, um Services entsprechend ihres Rahmes zu modellieren. Ziel ist es, mit Hilfe von zwei elementaren Rollen (Nutzer, Dienstleister) die Interaktion und damit die Serviceerbringung zu modellieren. Die Autoren verstehen ihren Ansatz dabei als semantisches (konzeptuelles) Metamodell und als Bindeglied zwischen Design- (z. B. UML, BPMN), Analyse- (z. B. Petri Netze, OWL-DL) und Implementierungssprachen (z. B. BPEL, WSDL). Ihre zentrale Hypothese zum Ansatz lautet, dass die verwendeten Konzepte in den Metamodellen generisch genug sind, um eine Abbildung von den Modellierungsprimitiven der einzelnen Sprachen auf diese grundlegenden Servicekonzepte zu ermöglichen. Die Stärke des Ansatzes soll auf der Modellierung des Verhaltensaspekts liegen, der die Interaktion zwischen den beiden elementaren Rollen beinhaltet. Damit besitzt der Ansatz eine sehr große Nähe zur modellgetriebenen Servicebeschreibung in dieser Arbeit. Mit Hilfe von generischen Konzepten wird versucht eine Brücke zu schlagen, die es ermöglicht bei der Modellierung Sprachen eigener Wahl zu verwenden. Der Erfolg oder die Anwendbarkeit ist dadurch entscheidend von der korrekten Auswahl und Granularität der verwendeten Servicekonzepte abhängig bzw. vom jeweiligen Verständnis der Anwender. Die Auswahl geeigneter Konzepte ist aufgrund der Problematik hinsichtlich der Servicedefinitionen (s. Kapitel 3.1) bei vielen Ansätzen zur Servicebeschreibung jedoch umstritten bzw. widersprüchlich. Insbesondere bietet der Ansatz von Quartel et al. jedoch keine Möglichkeit der automatisierten Transformation zwischen den möglichen Sprachen. Außerdem ist nicht gewährleistet, dass die Angabe semantischer Mappings für alle im Ansatz angeführten Sprachen möglich ist. Dies entspricht demselben Kritikpunkt wie im Ansatz von Kett et al. und Scheithauer et al. Transformationen auf Basis der Sprachkonzepte können nicht für alle Sprachen untereinander durchgeführt werden. Da der Ansatz rein konzeptueller Natur ist, wird dieses Problem jedoch abgeschwächt. Auch eine Erweiterung der Modelle erscheint prinzipiell möglich. Allerdings dürfte die Konsistenzwahrung u.U. schwierig werden, da bspw. keine Versionierung der Modelle einbezogen ist und damit frühere Versionen evtl. nicht mehr kompatibel zu einer Erweiterung sein können. Im Vergleich zu anderen hier vorgestellten Ansätzen berücksichtigen Quartel et al. allerdings auch die Suche bzw. das Finden von bereits modellierten Services in ihrem Ansatz und unterscheiden dabei zwischen Konzept- und Prozess-basierter Suche.

(Störrle und Glock 2007, S. 77-94): Geschäftsprozessmodellierung für Service-Orientierte Architekturen

Der Ansatz von Störrle und Glock ist ein Beispiel für die Entwicklungen im Bereich der SOA. Ausgehend von einer SOA werden Services nicht mehr nur als Web Services oder

als Softwarebausteine verstanden, sondern zunehmend als zentrales Element im Unternehmen (Business-driven SOA). Mit Fokus auf den zentralen Problemen bei Softwareprojekten in Unternehmen (z. B. Kostendruck, komplexe Anwendungslandschaft) wird ein Vorgehen präsentiert, welches eine durchgängige Entwicklung ausgehend von einer fachlichen Beschreibung hin zu einer technischen Umsetzung und zum Betrieb von Services ermöglicht. Auf Basis eines Wasserfallmodells werden Spezifikationen und Modelle entwickelt. Dabei kommen auch Techniken des MDE und insbesondere Modelltransformationen zum Einsatz, die einen automatisierten Übergang von fachlichen zu technischen Modellen erlauben. Ziel ist es, ohne einen zu hohen Programmieraufwand lauffähige Software zu erzeugen, indem bspw. BPMN- in BPEL-Modelle umgewandelt werden. Im Vergleich zu den anderen Ansätzen betrachtet der Ansatz von Störrle und Glock nur den Serviceaspekt des Ablaufs im Unternehmen. Es werden aber bereits hier Modelle und Transformationen zur Zielerreichung verwendet. Im Gegensatz zu den anderen Ansätzen ist der Kontext sehr spezifisch mit der Projektentwicklung in Unternehmen verbunden. Ebenso ist die Integration eines konkreten Vorgehens bislang eine Besonderheit des Ansatzes. Somit kann der Ansatz von Störrle und Glock zur Entwicklung der modellgetriebenen Serviceentwicklung beitragen.

(Kunau, Junginger et al. 2005, S. 187-216): Ein Referenzmodell für das Service Engineering mit multiperspektivischem Ansatz

Der Ansatz von Kunau et al. stellt ein branchenneutrales Referenzmodell für die Dienstleistungsentwicklung vor. Insbesondere wird dabei auf die Beschreibung notwendiger Aktivitäten Wert gelegt. Wesentlich für das Referenzmodell ist die Darstellung branchenübergreifender Gemeinsamkeiten, die Integration unterschiedlicher Fachdisziplinen, die Multiperspektivität im Sinne von Rollen, als auch von Phasen, die Einbeziehung deskriptiver wie auch präskriptiver Elemente sowie eine breite Palette von Perspektiven zur Beschreibung von Services. Explizit wird bei diesem Ansatz auch die Nähe zum SE hervorgehoben. Entlang eines spezifischen Lebenszyklus sollen systematisch Services entwickelt und zu diesem Zweck mittels mehrerer Sichten (Lebenszyklus, Basis der Entwicklung, Interventionsebenen, Zieldimensionen, Rollen, Marketing) beschrieben werden. Die Entwicklung selbst untergliedert sich in zwei zentrale Prozesse, das Service Engineering und das Service Management, sowie in einen Metaprozess, der steuernd einwirkt. Außerdem werden die einzelnen Aktivitäten durch den Einsatz von Werkzeugen unterstützt. Die Branchenneutralität wird bei Kunau et al. als besonderes Merkmal hervorgehoben. Der im Vergleich älteste Ansatz legt jedoch bereits Wert auf eine multiperspektivische Betrachtung von Services. Der Nachteil daran besteht jedoch in der reinen Beschreibung eines

Vorgehensmodells ohne konkrete Empfehlungen auszusprechen. Ebenfalls relevant ist die Modell- und werkzeuggestützte Entwicklung. Für alle beschriebenen Ansätze zeigt Tabelle 4-2 deren Bewertung anhand der zuvor vorgestellten Eigenschaften.

Kategorie / Eigenschaft	(Nardi, de Almeida Falbo et al. 2013)	(Berre, Lew et al. 2013)	(Steen, Iacop et al. 2012)	(Kett, Voigt et al. 2008; Scheithauer, Voigt et al. 2009)	(Quartel, Steen et al. 2007)	(Störrie und Glock 2007)	(Kunau, Junginger et al. 2005)
Modellierung							
Modellierungsart	O	M	M	M	M	M	M
Zahl vorhandener Perspektiven	4	2	6	4	5	3	3
Einsatz standardisierter Modelle	ja	ja	Archi-Mate	ja	ja	ja	SeeMe
Erweiterbarkeit (Modelle)	k.a.	ja	nein	nein	nein	ja	ja
Informationstransparenz	ja	ja	ja	ja	ja	ja	nein
Abstraktionsniveau	hoch	gering	gering	gering	mittel	gering	mittel
Rollenmodell							
Nutzergruppen	2	5	3	6	2	0	3
Rollenmodell	nein	ja	ja	ja	ja	nein	ja
Zielgruppen	2	3	4	2	2	3	4
Anwendung							
Anwendungsgebiet	1)	2)	3)	4)	5)	6)	7)
Anwendbarkeit	nein	ja	ja	ja	ja	ja	ja
Anpassbarkeit	ja	ja	ja	ja	ja	ja	ja

1) Service Science and Services Computing, 2) Service Innovation and Realisation, 3) Agile Serviceentwicklung, 4) Service Marktplätze / Internet of Services, 5) Webdienstleistungen, 6) Business-driven SOA, 7) Service Engineering

Tabelle 4-2: Eigenschaften verwandter Ansätze

In eine zeitliche Reihenfolge gesetzt, zeigen die hier vorgestellten Ansätze die Entwicklung einer umfassenden Servicegestaltung. Dabei ist festzustellen, dass die konkrete Umsetzung zu anfangs noch den Anwendern von Referenzmodellen überlassen ist. Beginnend mit der prozessualen Ausgestaltung wurden im Laufe der Zeit Ansätze komplexer was die Zahl der betrachteten Perspektiven angeht und zugleich konkreter bei den Empfehlungen zur Umsetzung. Insbesondere die jüngeren Ansätze zeigen die Relevanz des Themas einer umfassenden Servicegestaltung mit Hilfe von Modellen zur Darstellung unterschiedlicher Perspektiven. Der Grund hierfür liegt im zunehmenden Reifegrad und damit einhergehen-

der Nutzung modellgetriebener Verfahren und Automatisierungen im Bereich der Modelltransformationen. Bis auf einen verfolgen alle einen modellbasierten Ansatz und verbinden Perspektiven auf einer Metamodellebene. Die direkte Kopplung unterschiedlicher Modelle kann jedoch u.U. schwierig oder sogar unmöglich werden. Das ist von den konkreten Modellierungssprachen abhängig. Allerdings ist anzunehmen, dass bei steigender Zahl an verwendeten Modellierungssprachen zwangsläufig Inkompatibilitäten auftreten.

4.3 Zusammenfassung

In Bezug zu den Forschungsfragen 1 & 2 (vgl. Definition 1-3 und Definition 1-4) lassen sich aus den hier dargestellten Grundlagen bereits erste Erkenntnisse ableiten. Hinsichtlich der ersten Forschungsfrage zu einer allgemeingültigen Definition bzw. Beschreibung eines Service oder einer angebotenen Dienstleistung kann eine abgeschlossene Menge an Attributen oder Modellen nicht spezifiziert werden. Zwar kann im Sinne einer bestimmten Fachdisziplin bzw. einer konkreten Anwendungsdomäne ein vollständiges Modell im Sinne der Modellierer angegeben werden, für eine übergreifende Abbildung ist es jedoch nicht zweckdienlich gewissermaßen die Vereinigungsmenge aller involvierten Disziplinen oder Domänen zu bilden. Dagegen spricht, dass viele Eigenschaften mehrdeutig in unterschiedlichen Kontexten verwendet werden. Ebenso ist es nicht zweckdienlich die Schnittmenge zu wählen, da dann je nach Kontext sehr viele Eigenschaften nicht berücksichtigt werden können bzw. die Schnittmenge nur sehr wenige, allgemeingültige Aussagen zulässt. Die Zahl der vorhandenen Ansätze zur Definition oder zur Modellierung von Services vermag diese Einschätzung ebenso zu belegen. Daher erscheint es sinnvoll, hinsichtlich der Beschreibung von Services Anwendungs- bzw. kontextgetrieben zu agieren und einen Ansatz zur Beschreibung so zu gestalten, dass eine hohe Flexibilität und Variabilität in der Beschreibung ermöglicht wird. Dementsprechend kann eine adäquate Servicebeschreibung nur dann gegeben sein, wenn sie aus Sicht eines konkreten Modellierers vollständig ist, nicht jedoch aufgrund einer allgemeingültigen Definition. Unterstützt wird diese Aussage ebenfalls durch die als verwandte Ansätze beschriebenen Lösungen. Dort zeigt sich, dass eine multiperspektivische Lösung, bei der unterschiedliche Perspektiven und Abstraktionsgrade miteinander verbunden werden zum aktuellen Stand der Forschung gehören und viele Ansätze bzgl. des Vorgehens große Übereinstimmungen besitzen.

Hinsichtlich der zweiten Forschungsfrage kann auf Basis der verwandten Ansätze ebenfalls eine Aussage getroffen werden. Die zeitliche Entwicklung zeigt, dass modellgetriebene Verfahren eine stärkere Verbreitung und Unterstützung erfahren. Wie bereits beschrieben dürfte der Grund dafür in einem zunehmenden Reifegrad der verwendeten Konzepte und

vor allem auch der Technologien liegen. Gerade Fortschritte im technologischen Bereich, also die Entwicklung neuer Frameworks und Technologien zum Umgang mit Modellen und Modelltransformationen, reduzieren Einstiegsbarrieren. Die zugrundeliegenden Konzepte der Modellierung erfahren dadurch verstärkt einen über den akademischen Verwendungszweck hinausgehenden Nutzen. Die Vielfalt und Variabilität bestehender Modellierungsarten in Verbindung mit einfach umzusetzenden Modelltransformationen erleichtert die Entwicklung multiperspektivischer Modellierungsansätze enorm. Wichtig ist in diesem Zusammenhang auch die Weiterentwicklung von Transformationsansätzen weg von einem rein syntaktischen Vergleich hin zu einer deskriptiven Beschreibung von Korrespondenzen. Solche Korrespondenzen können mit Hilfe eines zusätzlichen Modells festgehalten werden, mit dessen Hilfe Transformationen schlussendlich auch bei komplexeren Sachverhalten eingesetzt werden können, wodurch die Konzipierung der modellgetriebenen Servicebeschreibung unterstützt wird.

Teil II

Service Modeling Framework

Das hier vorgestellte Service Modeling Framework (SMF) bildet den konzeptuellen Rahmen für eine modellgetriebene Servicebeschreibung. Das SMF ist nicht losgelöst zu betrachten, d.h. die Anwendung des Rahmenwerks erfolgt innerhalb eines größeren Rahmens und besitzt integrativen Charakter für die weiteren Komponenten dieses Rahmens. Insbesondere ist das SMF durch das im Exkurs II: Servicemodellierung in LSEM beschriebene LSEM (Logistik Service Engineering & Management) beeinflusst. Ziel ist es, für bestehende Modelle einen höheren Wiederverwendungsgrad zu erreichen und deren Informationen in verschiedenen Kontexten wiederzuverwenden. Das SMF unterstützt zu diesem Zweck die Generierung von Artefakten aus bestehenden Modellen und ermöglicht damit einen über den reinen Dokumentationszweck hinausgehenden Verwendungszweck der Modelle.

Als Rahmenwerk bildet das SMF den Bezugsrahmen für die verschiedenen Bausteine, die in der Arbeit entwickelt werden. Integraler Bestandteil eines Rahmenwerks sind Methoden und Techniken, mit deren Hilfe Artefakte entwickelt und in den festgelegten Grenzen des Rahmenwerks Anwendung finden. Ein Rahmenwerk dient dabei auch dem Zweck ein Grundgerüst für wichtige Konzepte bereitzustellen und die bereits genannten Grenzen, in Sinne von Anforderungen, Restriktionen oder Voraussetzungen zu definieren. Auf das SMF angewendet bildet das Kapitel 5 die Grenzen des Rahmenwerks ab. Es enthält zunächst eine Einführung in die modellgetriebene Servicebeschreibung und motiviert deren Entwicklung. Außerdem werden notwendige Anforderungen und Restriktionen definiert, die den Rahmen für die Beantwortung der Forschungsfragen 3 & 4 (vgl. Definition 1-5 und Definition 1-6) bilden. Vor allem die Beantwortung der Frage nach der Überführung von Modellen bedarf spezifizierter Grenzen innerhalb deren die zu entwickelnden Konzepte ihre Gültigkeit besitzen. Kapitel 6 definiert wichtige Konzepte und die zentrale Methode des Rahmenwerks als wesentliche Artefakte, zum einen ein Basismodell als Bindeglied zwischen Modellen und zum anderen eine Modellierungsmethode, mit deren Hilfe die erforderlichen Schritte zur Modellverknüpfung beschrieben sind. Kapitel 7 schließt das Rahmenwerk mit Hilfe geeigneter Techniken ab, konkretisiert die Umsetzung der Überführung und definiert eine auf dem Basismodell zugrundeliegende Sprache und spezifiziert die Details der Modellintegration als wesentlichem Schritt bei der Überführung.

Hinsichtlich der Ausgestaltung des Rahmenwerks sind neben den Inhalten der bereits erwähnten Kapitel die in Kapitel 5 dargelegten Anforderungen zu berücksichtigen und bei der prototypischen Umsetzung einzuhalten. Die Spezifikation und Implementierung eines entsprechenden Prototyps ist Teil der Validierung und wird in Kapitel 9 beschrieben.

5 Modellgetriebene Servicebeschreibung

5.1 Einführung

Die modellgetriebene Servicebeschreibung als zentralem Paradigma des SMF stellt einen Ansatz dar, mit dessen Hilfe konzeptuelle wie technische Modellierung eingesetzt wird, um ein

- einheitliches,
- vollständiges und
- für alle Nutzer transparentes

Abbild der Servicelandschaft zu erstellen. In dieser Arbeit wird dazu ein Ansatz vorgestellt, der es ermöglicht Services im Allgemeinen und logistische Services im Speziellen (s. Kapitel 8 und 9) auf Basis von konkreten Servicemodellen zu entwickeln, ohne ein umfassendes Servicemetamodell bereitstellen zu müssen. Analog zur Referenzmodellierung (s. Exkurs I: Referenzmodellierung) entsteht im SMF ein komplexes Servicemodell durch Verknüpfung der Teilmodelle miteinander. Allerdings wird für das Gesamtmodell nicht die Entwicklung eines Ordnungsrahmens empfohlen, da sich gezeigt hat, dass die Entwicklung eines umfänglichen Gesamtmodells bereits mehrfach (s. 3.2) vorangetrieben wurde, ohne jedoch einen Konsens bzgl. des Inhalts zu finden. In (Augenstein, Ludwig et al. 2012) wurde dazu auch detailliert die Problematik eines Gesamtmodells für Services diskutiert. Bei der Entwicklung eines multiperspektivischen Modells basierend auf Teilmodellen sind drei unterschiedliche Vorgehensweisen möglich:

- Top-down,
- Middle-out und
- Bottom-up

Unter Top-down wird ein Vorgehen definiert, bei dem ein Referenzmodell zunächst vorgegeben wird, das als integratives Element fungiert (z. B. (Steen, Iacop et al. 2012)). Sukzessive werden verschiedene Sichten hinzugefügt, die mit Hilfe des integrativen Modells konzeptuell miteinander verbunden sind und deren Fokus auf einem spezifischen Aspekt liegt. Dieses Vorgehen wird insbesondere bei der Entwicklung von Referenzmodellen angewandt. Konträr dazu steht ein Bottom-up Vorgehen, bei dem zunächst die Teilmodelle spezifiziert und daraus ein integratives Modell abgeleitet wird, welches entweder aus dem kleinsten gemeinsamen Nenner oder aus der Obermenge der vorhandenen Konzepte der Teilmodelle gebildet wird. Ein letztes, Middle-out genanntes, Vorgehen kombiniert beide vorherige Ansätze, indem entweder partiell Vorgaben definiert, jedoch nicht auf alle Bereiche angewendet werden oder durch die Definition von Schnittstellen, die einer späteren

Erweiterung des Gesamtmodells dienen. In dieser Arbeit wird die letztgenannte Variante favorisiert, wobei ein zentraler Aspekt in der Wiederverwendung bereits vorhandener Modellierungssprachen liegt und somit die Entwicklung eines umfassenden Servicemodells als zentralem Artefakt verworfen. Für die Umsetzung wird in 6.3 die Entwicklung eines Basismetamodells gezeigt, mit dessen Hilfe unterschiedliche Servicemodelle, die spezifische Perspektiven repräsentieren, zusammengeführt werden. Gerade aber die Fokussierung auf Services bzw. Servicemodelle als wichtige Bestandteile des Basismetamodells führt zu einer Zusammenführung unterschiedlicher Modellierungsarten und der Fachsprache einer Anwendungsdomäne für die ein Abbild der Servicelandschaft entwickelt werden soll. In (Augenstein, Müller et al. 2010) wurde bereits beschrieben, wie durch Verwendung von Ontologien sowohl unterschiedliche Modellierungskonzepte als auch fachliche Konzepte auf dieselben Servicekonzepte abgebildet werden können und somit auch eine Abbildung von Modellkonzepten unterschiedlicher Modellierungssprachen auf eine Fachsprache erfolgen kann. Verschiedene Modelle, die unterschiedliche Perspektiven eines Service repräsentieren, können demzufolge über die Verwendung servicespezifischer Konzepte und mit Hilfe einer domänenspezifischen Fachsprache aufeinander abgebildet und so ein vollständiges Bild einer Servicelandschaft hergestellt werden. Dies führt dazu, dass die Lücke zwischen rein konzeptueller Modellierung und der technischen Umsetzung bei der Servicegestaltung und -ausführung geschlossen und so eine umfassendere, ganzheitliche Sicht auf Services generiert werden kann. In Anlehnung an (Thomas, Leyking et al. 2007, S. 6) befindet sich das SMF auf der Gestaltungsebene an der Schnittstelle zur Konfigurationsebene, auf der eine modellbasierte Gestaltung der Servicelandschaft entworfen wird. Mit Hilfe von mehreren Modellierungsebenen (Abstraktionsstufen) werden Lösungsräume bereitgestellt, in denen schrittweise verschiedene Modelle zu einem Service miteinander verbunden werden, um einerseits diesen Service aus unterschiedlichen Perspektiven zu beschreiben und andererseits ein Konfigurieren der notwendigen IT-Systeme ermöglichen. Mit dem SMF können so auch Aufgaben des IT-Alignments erfüllt werden: Soll eine über die Dokumentation hinausgehende Verwendung von Modellen erreicht werden, so müssen notwendige IT-Systeme an die Bedürfnisse angepasst werden. Mit Hilfe modellgetriebener Verfahren können dabei sowohl Systemkonfigurationen an die Bedürfnisse angepasst oder auch Teile der Systeme generiert werden. Die Lücke zwischen fachlich-getriebenen Modellen auf der einen Seite und IT-nahen Modellen auf der anderen Seite stellt nach wie vor hohe Herausforderungen an die semantisch korrekte Übersetzung der Anforderungen auf IT-Systeme (Nissen, Petsch et al. 2007, S. 64). Durch die Verwendung und Verknüpfung mehrerer Modelle kann die Schnittstellenkomplexität zwischen Fach- und IT-Abteilungen

reduziert und die Zahl fehlerhafter Entwurfsentscheidungen reduziert werden. Insgesamt betrachtet ergeben sich daraus mehrere Verwendungsmöglichkeiten und Anwendungsgebiete des SMF, die nachfolgend zusammengefasst dargestellt sind:

- Modellmanagement
 - Zentrale Modellverwaltung für alle entwickelten Services aller beteiligten Dienstleister
 - Zentrale Metamodelverwaltung (Modellierungssprachen)
 - Versionierung von Modellen und Metamodellen
 - Erstellen von Templates zur schnellen Generierung neuer Modelle
- Informationsmanagement / Informationsaggregation
 - Bereitstellen von relevanten Informationen bei der Modellierung eines spezifischen Serviceaspekts
 - Verknüpfung von unterschiedlichen Modellen auf Basis der gemeinsam enthaltenen Informationen
- Anwendungsgebiete:
 - Inhaltsvergleich zwischen Modellen
 - Inhaltsvergleich innerhalb eines Modells
 - Verknüpfung von Modellen für eine ganzheitliche Repräsentation von Services

In Bezug zu LSEM (s. Exkurs II) konkretisieren sich die Ziele des SMF folgendermaßen: Der darin enthaltene Lebenszyklus besteht aus vier Phasen, in denen verschiedene Artefakte zur Beschreibung auf technischer oder konzeptioneller Stufe eingesetzt werden. Bei den Artefakten handelt es sich um spezifische Modelle und Softwarekomponenten (Prototypen), die jeweils mit Hilfe eines Metamodells entwickelt wurden. Die Aufgabe des SMF ist es, diese Modelle zu verwalten und „Informationsbrücken“ zwischen den Prototypen zu bilden. Modelle werden sowohl auf Meta- wie auf Instanzebene in einem Repository (s. 7.2 bzw. 9.3.3) vorgehalten und mittels eines Editors (s. 7.3 bzw. 9.3.4) miteinander verknüpft, so dass Informationen verschiedener Modelle und Prototypen innerhalb des LSEM zur Verfügung stehen. Beispiele für zu verwaltende Modelltypen sind Prozess- (BPMN, EPK) und Simulationsmodelle (Prototyp, Arena, ED), Servicekataloge (Service Map) oder auch Provider- bzw. Datenmodelle oder Service Level Agreements. Mittels SMF werden Beziehungen zwischen diesen heterogenen Modellen hergestellt und vorhandene Informationen zwischen Modellen übertragen. Durch die Herstellung von Verknüpfungen zwischen den unterschiedlichen Modelltypen sorgt das SMF für eine konsistente Darstellung und damit

für eine ganzheitliche Betrachtung von Services. Es wird außerdem ermöglicht, dass Modelle auf unterschiedlichen, konzeptuellen Ebenen so kombiniert werden, dass deren Operationalisierung auf einem hohen Level erreicht wird.

5.2 Anforderungen

Anforderungen an SMF ergeben sich aus den Anwendungsgebieten bzw. aus dem Zweck, für den das SMF entwickelt wird. Sie sind zu trennen von den Restriktionen, die im nachfolgenden Abschnitt 5.3 genannt werden und die den Rahmen der Anwendbarkeit näher definieren bzw. einschränken. Die Anforderungen hingegen beschreiben Ansprüche an das SMF und werden zur besseren Verständlichkeit entsprechend den in Abbildung 1-1 genannten Forschungsgebieten aufgeteilt. Insbesondere sollen für die Nutzung bereits bestehender Modelle als auch für die Erstellung neuer Modelle die in Kapitel 2 vorgestellten GoM Anwendung finden. Dabei sind die Grundsätze Sprachadäquanz, systematischer Aufbau, Klarheit und Vergleichbarkeit an oberster Stelle bei der Entwicklung spezifischer Modelle zu nennen, mit deren Hilfe Modelle so erstellt werden können, dass eine Nutzung im Rahmen des SMF erleichtert wird. Im Allgemeinen dienen grundlegende Fragestellungen zu Services zur Definition von Anforderungen. Eine beispielhafte Auswahl sei hier genannt:

- Was ist die Funktionalität eines Service?
- Wie ist ein Service erreichbar (im Sinne eines digitalen, elektronischen Service) bzw. wo wird er angeboten (im Sinne eines physischen Service)?
- Wer bietet einen Service an und wer kann einen Service nutzen?
- Welche Ressourcen benötigt ein Service?
- Welche Dienstgüte wird für einen Service spezifiziert?

5.2.1 Anforderungen aus der modellgetriebenen Entwicklung

Anforderungen hinsichtlich der modellgetriebenen Entwicklung beziehen sich sowohl auf die zu erstellenden Modelle als auch auf die Arbeit mit den unterschiedlichen Modellen. Zu berücksichtigen gilt, dass das Ergebnis der Modellierung, sowohl die einzelnen Teilmodelle als auch deren Zusammenführung mit Hilfe des Basismetamodells in gleichbleibender Qualität und konsistent vorliegt. Daher wird die Forderung nach der *Zweckgebundenheit* eines Modells noch einmal betont. Modelle im Rahmen des SMF sollen

- übersichtlich und klar strukturiert sein und
- nur ein Minimum an Konstrukten beinhalten, um die Sachverhalte vollständig umzusetzen.

Ein Minimum an Konstrukten bedeutet aber auch, dass *Modelle in ihrer Größe insgesamt beschränkt* sein müssen, um noch übersichtlich zu sein. Nach (Mendling, Reijers et al. 2010) betrifft dies insbesondere grafische Modelle, die zum einen für den Modellierer zwar einfacher zu lernen sind, zum anderen jedoch durch die schiere Zahl von Modellelementen ebenso unübersichtlich werden können.

Modellierung im Allgemeinen, im speziellen aber auch innerhalb des SMF, muss *zukunfts-fähig* sein (Petz, Duckwitz et al. 2014), indem Sinne dass Modelle durch künftige Änderungen nicht obsolet oder inkompatibel werden und daher überarbeitet werden müssen. Stattdessen muss im SMF ein *Bestandsschutz* vorhanden sein, der es ermöglicht, dass Modellversionen definiert werden können und somit ein konsistentes Arbeiten über einen längeren Zeitraum ermöglicht wird.

Modellierung muss darüber hinaus auch unterschiedlichen Korrektheitsbedingungen gehorchen. Über die korrekte Erstellung von Teilmodellen hinaus, muss im SMF gewährleistet sein, dass die zu übertragenden Informationen bzgl. des fachlichen Kontexts korrekt transferiert werden (vgl. auch (Hallerbach, Bauer et al. 2008, S. 58)) Die *Korrektheit der Teilmodelle* ist dabei nicht direkter Bestandteil, da durch die jeweils zugrundeliegende Modellierungsmethode der Teilmodelle bereits ein Vorgehen zur korrekten Erstellung definiert ist. Es muss jedoch innerhalb des SMF sichergestellt sein, dass durch den Informationstransfer ein bislang als korrekt geltendes Modell nicht inkorrekt bzw. inkonsistent wird. Hervorzuheben ist auch die jeweilige Perspektive der Modellierer als Ausgangspunkt für die Modellierung, die im Zusammenhang mit der *Zweckgebundenheit* steht, wodurch die Modellbildung auch als Interaktion zwischen allen Akteuren aufzufassen ist (vgl. auch (Klischewski 2000, S. 21)). Daraus resultiert die Forderung nach einer notwendigen *Vereinigung dieser Perspektiven*. Erst dann ist es möglich, das volle Potenzial auszuschöpfen, da diese Perspektiven (z. B. physische, prozessbezogene oder logische Perspektive) gegenseitige Abhängigkeiten aufweisen (vgl. (Petz, Duckwitz et al. 2014), (Böhmman, Winkler et al. 2004)). Zusätzlich wird dadurch das angestrebte IT-Alignment, also die Überführung fachlicher Anforderungen in ausführende IT-Systeme, unterstützt und die Lücke zwischen fachlichen und technischen Modellen geschlossen (vgl. (Thomas, Leyking et al. 2007, S. 6)).

Allerdings birgt die Integration unterschiedlicher Perspektiven in Form verschiedener Modelle auch entsprechende Risiken. Insbesondere die *Abbildung gleichartiger Informationen in unterschiedlichen Modellen* ist für das SMF relevant, da mit Hilfe des SMF Informationen zwischen Modellen transferiert werden sollen. Namenskonventionen sind eine wiederkehrende Form, um der Problematik von Homonymen, Synonymen etc. zu begegnen (vgl.

(Buchwald, Bauer et al. 2010, S. 205f)), deren Einhaltung liegt aber in der Verantwortung der Modellierer. Stattdessen soll es mit dem SMF möglich sein, explizit *Beziehungen zwischen gleichartigen Informationen in unterschiedlichen Modellen zu spezifizieren* und diese Beziehungen so auszuwerten, dass Informationen automatisiert übertragen werden können.

Schließlich führt die Vermeidung von Inkonsistenzen, inkorrekten Modellen oder eine einheitliche Sprachregelung für alle betroffenen Modelle auch zu Anforderungen an die Modelltransformation bzw. an die verwendete Transformationssprache. Nach (Jouault und Kurtev 2007, S. 116) sind mehrere Varianten bei einer Transformation möglich, die die Sprachauswahl für die Regelbeschreibung betreffen. Aufgrund der Vielschichtigkeit der Modelle ist jedoch weder die Ausgangs- noch die Zielsprache eine Variante. Daher muss für das SMF eine *intermediäre Sprache* entwickelt werden, die es ermöglicht, Elemente unterschiedlicher Sprachen zu verknüpfen. Im Gegensatz zu existierenden Transformationsansätzen sollen Modelltransformationen im SMF *nicht über einen syntaktischen Vergleich* der Modellelemente umgesetzt werden, da dies ebenfalls aufgrund der Vielschichtigkeit der Modelle nicht zweckdienlich erscheint.

5.2.2 Anforderungen aus dem Service Engineering

Die Anforderungen des SE setzen bei der Entwicklung eines Service an und beschreiben die notwendigen Eigenschaften und Artefakte, die für eine systematische Herangehensweise erforderlich sind. Die *systematische Entwicklung* ist dabei selbst als Anforderung aufzufassen (vgl. (Scheer, Grieble et al. 2006, S. 20)).

Im Unterschied zur modellgetriebenen Entwicklung sind hier Anforderungen genannt, die sich zwar auch auf die Modellierung auswirken, in erster Linie jedoch die Methodik der Modellierung adressieren. Dazu zählt, dass nach (Kern, Böttcher et al. 2009) die Dienstleistungsmodellierung

- auf einer Modellierungstheorie mit einem in sich geschlossenen Rahmen und
- einem Set an Modellierungsmethoden basieren muss.

Mit Hilfe eines solchen Ordnungsrahmens lassen sich geeignete und benötigte Dimensionen zur Darstellung identifizieren (Bullinger, Fähnrich et al. 2003), die benötigten Ressourcen zur Ausführung in einem Raster erfassen und somit wiederverwendbar machen (Scheer, Grieble et al. 2006, S. 33) sowie adäquate Modellierungskonventionen für die Serviceentwicklung definieren (Meyer und Böttcher 2012, S. 39f).

Zuvor ist es jedoch notwendig *Services als Objekte der Entwicklung zu begreifen* (Scheer, Griebel et al. 2006, S. 31f). Als Rahmenwerk zur modellgetriebenen Servicebeschreibung muss das SMF einen solchen Ordnungsrahmen bereitstellen, indem

- notwendige Perspektiven zumindest vorgeschlagen,
- Ressourcen kategorisiert sowie
- Empfehlungen für ein Modellierungsvorgehen enthalten sind.

Darüber hinaus muss ein Lösungskonzept entwickelt werden, das Modellierungstheorie, Modellierungsframework und -sprachen in einen gemeinsamen Kontext einbettet und miteinander verknüpft (vgl. (Meyer und Böttcher 2012, S. 39)).

Die Modellierung erhält deshalb innerhalb des SE einen hohen Stellenwert, da mit ihr eine *Kommunikation zwischen den beteiligten Akteuren* der Serviceentwicklung realisiert werden soll. Dabei fordert das SE analog zu den Anforderungen der modellgetriebenen Entwicklung einen *ganzheitlichen Ansatz*, der mit Hilfe von Modellierungsmethoden umgesetzt werden muss. Insbesondere muss ein Ansatz des SE es ermöglichen zwischen Objekten bzw. Modellen eines Service *Verknüpfungen zu definieren* (vgl. (Scheer, Griebel et al. 2006, S. 31f)).

Ein solcher Ansatz führt im Weiteren zu Anforderungen an einen Entwicklungsprozess für Services, an die eingesetzten Werkzeuge und zugrundeliegenden Methoden sowie an die daraus resultierenden Ergebnisse. So sollte der Entwicklungsprozess

- spezifische Phasen beinhalten, die die einzelnen Schritten von einander entkoppeln,
- gleichzeitig aber auch eine im Verlauf zunehmende Detaillierung des Serviceartefakts hinsichtlich seiner Beschreibung ermöglichen.

Eingesetzte Werkzeuge sollen Schritten zugeordnet werden können und eine spezifische, für den Schritt angepasste Methode umsetzen und schließlich sollen produzierte Ergebnisse wiederholbar, allgemein und leicht verständlich sein. Eine Visualisierung kann dies unterstützen, wobei analog zur modellgetriebenen Entwicklung auch hier ein Sparsamkeitsprinzip der verwendeten Konstrukte gelten sollte.

5.2.3 Anforderungen aus der Serviceorientierung

Im Unterschied zu den vorangegangenen Abschnitten beinhalten die Anforderungen aus der SO Eigenschaften und Zielvorgaben, die sich mit architekturellen Vorgaben auseinandersetzen, d.h. diese Anforderungen zielen vermehrt auf das Rahmenwerk selbst ab, als nur auf einzelne Bestandteile.

So muss die modellgetriebene Servicebeschreibung das flexible Hinzufügen oder Entfernen von Modellierungsbausteinen bei der Serviceentwicklung unterstützen. Dies kann

bspw. durch eine *einheitliche Modellierung* der Einzelkomponenten (Services) und der Gesamtleistung erreicht werden (vgl. (Schmedes 2007, S. 190f)). Wesentlich ist dabei die Zusammenführung des Wissens aller Beteiligten, so dass für diese das Gesamtbild der modellierten Servicelandschaft erfassbar ist. Das SMF muss daher das *vorhandene Wissen für alle beteiligten Akteure in einer Weise verwalten* können, dass diese in ihrem jeweiligen Kontext (z. B. Marketing, IT, Produktmanagement) mit den vorhandenen Modellen arbeiten können.

In diesem Kontext ist es dann auch notwendig, Entwurfsentscheidungen zu treffen, die das *Einbeziehen von Standards* berücksichtigen. Sowohl für die Realisierung von Services als auch für die Beschreibung mit Hilfe von Modellen müssen einheitliche Terminologien und Modelltypen spezifiziert werden, die eine effiziente Servicelandschaft unterstützen (vgl. (Heutschi 2007, S. 30f), (Krafzig, Banke et al. 2005a, S. 27ff)).

Insbesondere muss eine *effiziente Kommunikation* zwischen den Akteuren ermöglicht werden, die auch unterschiedliche Wissensstände einbezieht und diese zu überwinden versucht (vgl. (Krafzig, Banke et al. 2005a, S. 27ff)). Eine Wiederverwendung von Informationen in anderen Kontexten muss dann so erfolgen, dass eine *Nachvollziehbarkeit* gewährleistet ist. Für die Speicherung, Präsentation und Verarbeitung von Wissen mit Hilfe des SMF können einige Anforderungen an eine SOA, als Umsetzungsvariante der SO, unmittelbar übernommen werden. Dazu zählen

- das Verbergen von Heterogenität, im Sinne von unterschiedlichen Modellierungsparadigmen oder Perspektiven,
- die Nutzung standardisierter Komponenten oder
- eine hohe Flexibilität, bspw. durch Unterstützung einer Vielzahl von Modellierungsstandards.

Ebenso sollte das SMF *Wiederverwendung, Komponierbarkeit* und *Modularisierung* unterstützen, indem Sinne, dass bestehende Modelle als eigenständige, wiederverwendbare Elemente angesehen werden, die mit Hilfe des SMF miteinander kombiniert werden können (vgl. (Lemke 2012, S. 29), (Fiege und Stelzer 2007, S. 911)). Schließlich gelten auch die Anforderungen nach *lokaler Autonomie* und *Auffindbarkeit* für das SMF, da Modelle, aus denen Informationen wiederverwendet werden sollen (Quellmodelle), nicht verändert werden dürfen und Modelle generell über einen zu spezifizierenden Identifikationsmechanismus auffindbar gemacht werden müssen.

5.3 Restriktionen

Die Nutzung des SMF unterliegt nur wenigen Restriktionen, um eine möglichst intuitive Anwendung zu erreichen und möglichst zahlreiche Anwendungsgebiete zu erschließen. Ein umfassendes Rahmenwerk zur modellgetriebenen Beschreibung von Services benötigt jedoch gewisse Grenzen, innerhalb deren das Rahmenwerk gültig bzw. anwendbar ist. Andernfalls besteht die Gefahr, dass ein solches Framework zwar universell im Sinne der Servicemodellierung ist, jedoch keinerlei Mehrwert zu bieten hätte, aufgrund der tatsächlich wenigen, tieferen Zusammenhänge.

Die Restriktionen werden dabei einerseits in zugrundeliegende Annahmen und notwendige Voraussetzungen unterschieden, wobei Annahmen die generelle Anwendung des Frameworks einschränken und Voraussetzungen spezifische Kriterien für eine Instanziierung darstellen. Insgesamt sollen Anforderungen an potenzielle Nutzer möglichst gering gehalten werden, um ein möglichst schnelles Einarbeiten zu gewährleisten.

Hinsichtlich der fachlichen Modellierung bestehen keine Einschränkungen bzgl. der inhaltlichen Überschneidung von Modellen. Vollständig disjunkte Modelle erlauben jedoch keine Wiederverwendung bereits bestehender Informationen. Allerdings wird der Einbezug der fachlichen Domäne für die Modellierung folgendermaßen eingeschränkt: Zu einer vollständigen Modellierung der fachlichen Zusammenhänge ist die möglichst vollständige Erfassung der zugrundeliegenden Anwendungsdomäne erforderlich. Die Domänenmodellierung, wenngleich auch nur exemplarisch, erfordert jedoch einen immensen Aufwand bei der Aufnahme und Modellierung der Zusammenhänge. Daher wird zunächst auf den Einbezug der Domäne, im Sinne kontextueller, ergänzender Informationen bei der Modellierung verzichtet. Dies stellt zwar eine Einschränkung in der Vollständigkeit dar, nicht jedoch in der prototypischen Realisierung, verbunden mit dem Aufzeigen der prinzipiellen Funktionsweise und Korrektheit der angewandten Konzepte.

Für die Anwendung des SMF wird für die Instanziierung der Konzepte (s. 9.2) jedoch zunächst eine Einführung in die Logistikdomäne (s. Kapitel 8) bereitgestellt, mittels derer das Fehlen eines Domänenmodells für den Leser kompensiert werden soll. Ebenfalls eingeschränkt wird auch der Raum möglicher Modellierungssprachen, die zur Modellierung einzelner Serviceaspekte genutzt werden können. Für den korrekten Ablauf der SMF-Komponenten und für die korrekte Umsetzung der Konzepte ist es notwendig, dass Modellierungssprachen über ein zu bestimmendes Metametamodell definiert und beschrieben werden können. Ist dies für eine bestimmte Sprache nicht möglich, so ist sie nicht geeignet. Unwichtig dagegen ist die konkrete Notation einer Sprache, in dem Sinne, dass sowohl grafische, als auch textuelle Modellierungssprachen verwendet werden können. Für grafi-

sche Modellierungssprachen soll o. B. d. A. (ohne Beschränkung der Allgemeinheit) gelten, dass sie ein Knoten-Kanten-Modell repräsentieren, d.h. auf Notationsebene wechseln sich nach bestimmten Gesichtspunkten Knoten (i.d.R. geometrische Figuren wie bspw. Rechtecke oder Kreise) und Kanten (i.d.R. Linien, die die Knoten miteinander verbinden) ab.

5.3.1 Annahmen

Während Anforderungen erwünschte oder zwingende Eigenschaften und Verhaltensweisen beschreiben, die für die Entwicklung des SMF gelten müssen, sind Annahmen mit der grundsätzlichen Anwendbarkeit des Rahmenwerks verknüpft. Annahmen beschreiben sowohl die äußeren Umstände wie auch innere Mechanismen, die einen logischen und konsistenten Rahmen für die Anwendung bilden. Sie sind unmittelbar mit den Basishypothesen (s. Definition 1-2) sowie mit den Forschungsfragen 1 (s. Definition 1-3) und 3 (s. Definition 1-5) verknüpft. Daher werden Annahmen sowohl an die Definition bzw. Beschreibung als auch an die Verknüpfung von Servicemodellen getroffen. Zunächst werden Annahmen dargestellt, die die äußeren Umstände reflektieren:

- Nutzer sind mit Modellierungskonzepten (grafisch oder textuell) im Allgemeinen vertraut und in der Lage konsistente Modelle zu erstellen.
- Unterschiedliche Nutzer sind mit unterschiedlichen Modellierungsmethodiken, -sprachen und -werkzeugen vertraut.
- Nutzer haben ein tiefgreifendes Verständnis über die Natur der darzustellenden Services.
- Nutzer haben ein tiefgreifendes Verständnis für ihre Anwendungsdomäne und sind daher in der Lage, Services aus unterschiedlichen Perspektiven zu betrachten.

Im Vergleich dazu sind Annahmen an die inneren Mechanismen an die grundsätzliche Wirkungsweise geknüpft:

- Services können nicht mittels nur eines Modells vollständig erfasst und beschrieben werden.
- Eine Perspektive erfasst einen spezifischen Aspekt eines Service und wird durch ein Modell repräsentiert.
- Bzgl. der enthaltenen Informationen ergänzen sich Modelle unterschiedlicher Perspektiven zu einem Gesamtbild.
- Eine Modellierung findet werkzeuggestützt statt, sodass die resultierenden Artefakte digital vorhanden sind.

- Durch die Beteiligung mehrerer bei der Modellierung eines Services treten typische Probleme der kollaborativen Modellierung (Synonyme, Homonyme, alternative Modellierungspfade, etc.) auf.
- Eine Verknüpfung von Modellen auf Basis fachlicher Beziehungen verbessert die Qualität des Gesamtbilds im Vergleich zu einer auf Syntaxvergleichen beruhenden Modelltransformation.
- Die Funktionsweise des SMF ist unabhängig von der Anwendungsdomäne.

5.3.2 Voraussetzungen

Analog zu den Anforderungen sind auch Voraussetzungen zwingende Eigenschaften und Verhaltensweisen. Im Unterschied jedoch sind Voraussetzungen nicht an die Entwicklung gekoppelt, sondern zwingend notwendig für eine Anwendbarkeit des Frameworks. Sind diese nicht erfüllt, so kann das SMF nicht als Ansatz zur Servicemodellierung herangezogen werden. In diesem Zusammenhang würde auch die Möglichkeit bestehen, einige der Annahmen zu den Voraussetzungen zu zählen wie bspw. das Nutzerverständnis bzgl. der Services oder der Modellierung, je nachdem wie die Gewichtung dieser Eigenschaft erfolgt. Grundsätzlich ist eine Nutzung des SMF dennoch möglich, verbunden mit einem gesteigerten Aufwand oder einem verringerten, gestifteten Nutzen. Voraussetzungen, auch in ihrem eigentlichen Wortsinn, bilden dagegen ein Ausschlusskriterium. Daher bestand bei der Konzeption das Ziel, möglichst wenige Voraussetzungen an die Nutzung zu definieren:

- Vorhandensein oder Aufbau einer Servicelandschaft.
- Nutzung und Pflege von Modellen zur Beschreibung, keine Ontologien.
- Vorhandensein von modellbasierten Werkzeugen, d.h. sie besitzen ein Metamodell als formale Basis.
- Alle zum Einsatz kommenden Metamodelle lassen sich durch ein einziges Metamodell beschreiben (vgl. Ecore in 9.3).

Das SMF als modellgetriebenes Rahmenwerk zur Servicemodellierung ist bewusst so gestaltet, dass dem jeweiligen Nutzer eine weitreichende Flexibilität im Umgang mit Modellen bleibt. Anforderungen und Restriktionen sind bewusst so gewählt, dass im Gegensatz zu bestehenden Ansätzen (vgl. Kapitel 4) keine strikten Vorgaben an die Modellierung gemacht werden. Die nächsten beiden Kapitel 6 und 7 beinhalten die dazu notwendigen Konzepte und Methoden und zeigen wie diese in werkzeuggestützt umgesetzt werden können. Essentiell sind vor allem die Definition einer Sprache zur Beschreibung von Beziehungen zwischen Modellen sowie deren automatisierte Auswertung.

6 Multiperspektivische Servicemodellierung

Die wesentlichen Erkenntnisse aus den Modellierungsgrundlagen des Kapitels 2 beschreiben Modelle als Abbilder der Realität zum Zwecke der Vereinfachung oder der Beherrschbarkeit von Systemen. Insbesondere durch die Systemtheorie wird die Möglichkeit der Vereinfachung jedoch beschränkt, da die Komplexität eines Modells zur Beherrschung eines Systems direkt proportional zur Komplexität des Systems selbst definiert ist. Dies ist ohne Weiteres auch auf Modelle der WI übertragbar und zeigt sich insbesondere in der Unternehmens- oder in der Referenzmodellierung. Konsequenterweise haben sich dort multiperspektivische Ansätze etabliert, die die ganzheitliche Abbildung auf mehrere (disjunkte) Teilmodelle herunterbrechen und die Teilmodelle über ein zentrales, integratives Modell zueinander in Beziehung setzen. Dadurch wird zum einen die Übersichtlichkeit und Spezifität der Modelle gewährleistet und zum anderen der übergeordnete Zweck der Beschreibung oder Analyse erfüllt. Wesentliches Merkmal ist demnach auch hier der „Separation of Concerns“, also die Fokussierung der Teilmodelle auf einzelne Aspekte eines Unternehmens oder Informationssystems. In Analogie wird mit der modellgetriebenen Servicebeschreibung ein Ansatz beschrieben, der einzelne Aspekte eines Service mittels Teilmodellen beschreibt, die durch ein integratives Modell zueinander in Beziehung gesetzt werden. Dadurch soll verhindert werden, dass durch das spätere Hinzufügen von Aspekten ein bereits komplexes Servicemodell zu unübersichtlich wird oder gar inkonsistente Beziehungen eingeführt werden müssen, die das gesamte Modell widersprüchlich werden lassen.

Das SMF hat nicht zum Ziel ein neues Servicemodell als einzelnes Artefakt zu entwickeln, sondern nutzt spezifische Modelle und Modellierungssprachen, um Perspektiven auf einen Service zu beschreiben und so Aspekte von Services zu modellieren. Gerade aber die strikte Einhaltung einer Aufteilung erweist sich als Schwierigkeit. Dies ist teilweise der Mehrdeutigkeit der verwendeten Sprachen geschuldet. Beispielweise wird je nach Modell eine Beschreibung mehr im technischen Sinne als Spezifikation der Schnittstellen gesehen oder eben nur als eine textuelle Beschreibung des Leistungsumfangs eines Services. Dadurch werden, zumindest was die modellierten Elemente betrifft, ebenfalls Überschneidungen erzeugt. Neben der sprachlichen Mehrdeutigkeit können einzelne Servicebestandteile einer bestimmten Perspektive nicht eindeutig zugeordnet werden, was bei den analysierten Modellen bereits festgestellt wurde. Beispiele sind etwa die unterschiedliche Auffassung von Schnittstellen (konzeptuell oder softwaretechnisch) oder auch Aktivitäten bzw. Aktionen, die entweder einem Prozess oder dem Service selbst zugeordnet werden können.

Das Lösungskonzept dieser Arbeit besteht darin, über ein so genanntes Basismodell Vorgaben und Schnittstellen zu definieren, die Verbindungen zwischen Servicemodellen und diesem Basismodell ermöglichen. Elemente und deren Beziehungen untereinander innerhalb des Basismodells sorgen darüber hinaus für eine konsistente Verknüpfung der einzelnen Teilmodelle. Dazu wird ein Vorgehen beschrieben, mittels dessen Servicemodelle mit dem Basismodell verknüpft werden. Für das SMF wird hierzu eine vollständige Modellierungsmethode (vgl. 2.2 bzw. 6.4) beschrieben. Kern der Methode ist aus Sicht eines Modellierers zunächst die Überwindung der syntaktischen und semantischen Lücke zwischen den eingesetzten Modellierungssprachen, d.h. Modellelemente aus unterschiedlichen Sprachen werden über das Basismodell miteinander verknüpft. Die konkrete Notation, also das Aussehen der Sprachelemente bei der Verwendung der Sprache wird zunächst nicht betrachtet.

6.1 Modellhierarchie

Um Modelle verschiedener Modellierungssprachen auf syntaktischer Ebene miteinander verknüpfen zu können, wird das Konzept der Modellhierarchie, wie in Abbildung 6-1 dargestellt, konsequent genutzt. Das in Kapitel 2.3 beschriebene Konzept der Metamodellierung definiert die Zusammenhänge zwischen Modellen und den Modellen von Modellen. Dies ist für SMF insofern relevant, da Servicemodelle auf der Metamodellebene miteinander verbunden werden und neben Modellen auch die zugrunde liegenden Metamodelle in den Modellierungsprozess einbezogen werden müssen. Servicemodelle, die im SMF Verwendung finden, sind Modelle der Ebene M_1 . An einem Beispiel der Logistikdomäne sollen die Zusammenhänge verdeutlicht werden. Die Abbildung eines realen Transports wird zu Planungszwecken in einem Prozessmodell (BPMN) auf der Ebene M_1 festgehalten, oder präziser, der Transport wird mit Sprachmitteln der BPMN-Notation beschrieben. Das BPMN-Metamodell enthält alle notwendigen Konstrukte und Beziehungen, um die Sprache vollständig zu beschreiben und um somit das Konstruieren eines validen Prozessmodells zu ermöglichen. Da ein solches Prozessmodell nur einen möglichen Serviceaspekt darstellt und darüber hinaus eine Vielzahl an Sprachen zur Prozessbeschreibung existiert, ist eine Vielzahl von Metamodellen für das SMF relevant.

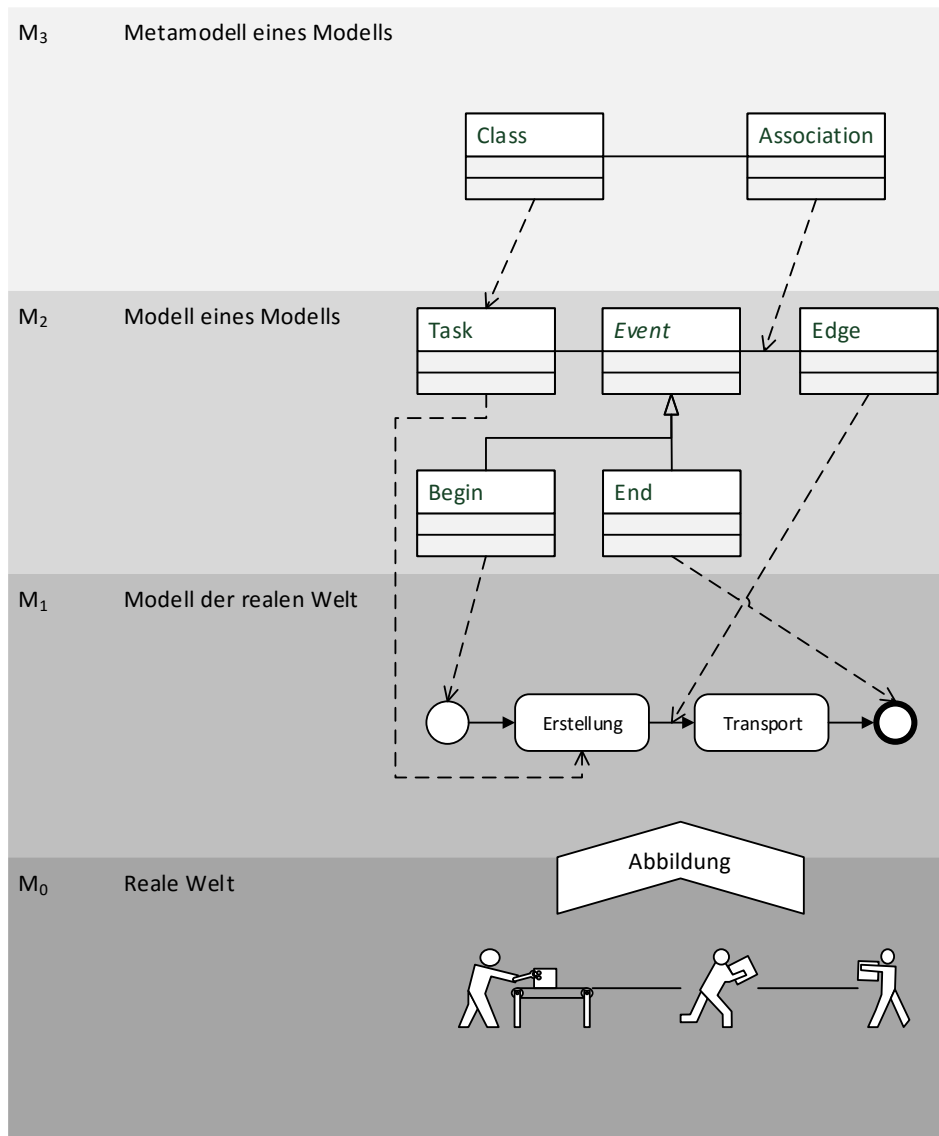


Abbildung 6-1: Modellhierarchie

Im SMF werden Perspektiven auf der Metamodellebene miteinander verknüpft, so dass auf der Metametamodellebene (M₃) eine Sprache notwendig wird, mittels derer die Sprachen auf der Metaebene einheitlich definiert und beschrieben werden können. Das Metametamodell enthält dazu alle Bausteine, die notwendig sind, um Elemente einer Modellierungssprache hinreichend zu definieren. Die gestrichelten Pfeile, die in Abbildung 6-1 zwischen Elementen unterschiedlicher Ebenen eingezeichnet sind, zeigen beispielhaft den Zusammenhang zwischen der Definition eines Elements auf Ebene M_i und dessen Ausprägung auf Ebene M_{i-1}. Im Falle der BPMN-Notation werden Elemente wie „Tasks“, „Events“, aber auch Beziehungen zwischen diesen Elementen wie „Edges“ als Konzepte mit Mitteln der Metametasprache beschrieben. Das Metametamodell im Beispiel enthält dazu die Konzepte „Class“ und „Association“, wobei alle Konzepte der Ebene M₂ als Class modelliert werden und das Konzept Association für die Beschreibung der Beziehungen zwischen den

Konzepten in M₂ Verwendung findet. Das Association-Konzept ist aufgrund der selbstdefinierenden Eigenschaft in M₃ doppelt vorhanden. Einerseits als Kante zwischen den beiden Konzepten, aber auch als Konzept. Zu beachten ist, dass keines der dargestellten Modelle vollständig ist und zur besseren Übersichtlichkeit nur auszugsweise abgebildet ist.

Im Ergebnis erreicht man dadurch, dass alle eingesetzten Modellierungssprachen der Ebene M₂ mit Hilfe derselben Metamodellierungssprache beschrieben sind, deren Modell wiederum auf Ebene M₃ definiert ist. Dadurch sind die Modellierungssprachen untereinander kompatibel innerhalb des SMF, sodass es prinzipiell möglich ist, eine Verknüpfung von Elementen unterschiedlicher Modellierungssprachen herzustellen. Diese Vereinheitlichung ist auch ein wesentliches Merkmal und gängige Praxis bei Modelltransformationsansätzen (s. 0). Auf diese Weise wird im SMF ein multiperspektivisches Servicemodell zunächst auf syntaktischer Ebene realisiert. Das gemeinsame Metametamodell schließt damit die syntaktische Lücke zwischen den eingesetzten Modellierungssprachen.

6.2 Modellkomposition

Das Schließen der semantischen Lücke zwischen den Modellen geht über eine einheitliche Syntax hinaus. Unterschiedliche Modelle können aus Sicht eines Modellierers in verschiedenen, syntaktischen Elementen dieselben Informationen beinhalten (z. B. ein Task-Element aus BPMN und ein Activity-Element aus UML). Die semantische Lücke kann in dem Fall geschlossen werden, indem ein Modellierer eine Beziehung zwischen solchen Elementen explizit definiert. Daher soll mit diesem Ansatz auf semantischer Ebene ein Modellierer in die Lage versetzt werden, Modellinhalte zueinander in Beziehung zu setzen. Das bedeutet, zu jedem Modell muss bekannt sein welche Inhalte vorhanden sind, und wie sich Inhalte aus verschiedenen Modellen zueinander verhalten.

Weitere Beispiele für Beziehungen zwischen Modellelementen können disjunkte, ergänzende oder identische Informationen sein. Während die Identifikation der Modellinhalte über den Namen der Modellelemente, also über die Syntax erfolgen kann, erfordert ein Vergleich der Inhalte einen wesentlich höheren Aufwand. Bei Transformationsansätzen wird über Regeln oder Transformationsmodelle auf syntaktischer Ebene definiert, wie der Inhalt aus einem oder mehreren Ausgangsmodellen in ein oder mehrere Zielmodelle transferiert wird. Viele der betrachteten Ansätze erlauben dabei nur die Berücksichtigung syntaktischer Elemente bei der Auswahl und führen die Transformation auf der Metaebene durch, indem sie analog zu 6.1 auf eine gemeinsame Syntax zur Beschreibung zurückgreifen. Dabei werden auf der Metaebene Beziehungen zwischen Elementen erstellt und teilautomatisiert eine Transformation auf Modellebene durchgeführt. Im Vergleich zu diesen

Ansätzen wird mit dem SMF ein anderer Ansatz zur Modellintegration verfolgt, um die folgenden Probleme zu vermeiden: Zum einen werden bei bestehenden Transformationsansätzen Modelle auf der Metaebene direkt über Regeln miteinander verknüpft. Bei n möglichen Modelltypen müssten für das SMF zunächst $n*(n-1)/2$ Transformationen angegeben werden, wie das bspw. auch bei (Kett, Voigt et al. 2008; Scheithauer, Voigt et al. 2009) der Fall ist. Außerdem müssten für jeden hinzugenommenen Modelltyp ebenfalls entsprechend viele Modelltransformationen definiert werden. Zum anderen können Transformationen nur durch Vergleich der abstrakten Syntax automatisiert durchgeführt werden, so dass ein zusätzlicher, manueller Schritt notwendig wird, um eine vollständige Transformation zu gewährleisten (s. hierzu die Etablierung extensionaler Verbindungen bei (Romero, Jaén et al. 2009) oder Intermodellkorrespondenzen bei (Selonen und Kettunen 2007)). Existierende Modelltransformationen sind darüber hinaus nur dann sinnvoll einsetzbar, wenn sowohl mit dem Ausgangs- als auch mit dem Zielmodell ein ähnlicher Sachverhalt modelliert werden soll (z. B. die Transformation eines BPMN-Modells in ein BPEL-Modell zur Automatisierung). Das SMF umfasst jedoch Modelle, die unterschiedliche Aspekte eines Service betreffen (z. B. Prozess, technische Beschreibungen, Kennzahlen oder Preismodelle).

Statt einem Transformationsmodell wird das im nächsten Abschnitt definierte Basismetamodell eingesetzt, um Verbindungen auf der Metaebene herzustellen, was zu einer veränderten Beziehung zwischen Quell- bzw. Zielmodellen und dem Basismetamodell führt. Im Unterschied zu Transformationsregeln oder -modellen werden Servicemodelle mit dem Basismetamodell ausschließlich auf fachlicher Ebene, d.h. über Elemente der Anwendungsdomäne, miteinander verbunden. Dazu werden Elemente der Quell- und Zielmodelle, die über das gemeinsame Metametamodell bereits syntaktisch miteinander verknüpft werden können, mit Hilfe domänenspezifischer Angaben und mit Hilfe des Basismetamodells miteinander verbunden. Dadurch entsteht eine lose, flexible Kopplung zwischen Modellen, die jedoch automatisiert auswertbar ist. Die Vorgehensweise hierzu wird im Abschnitt 6.4 detailliert beschrieben. Die semantische Lücke zwischen Modellen wird daher nicht ausschließlich durch die Angabe von Transformationsregeln geschlossen, sondern durch ein separates Modell, das relevante Modellelemente deskriptiv miteinander verbindet.

6.3 Modellbasis

Im Vergleich zu einem Transformationsmodell, das auf eine bestimmte Art von Transformation ausgelegt ist, ist das Basismetamodell so konzipiert, dass es als Bindeglied zwischen verschiedenen Ausprägungen von Servicemodellen eingesetzt werden kann. Analog zu Transformationsmodellen wird mit seiner Hilfe spezifiziert was in einem späteren

Schritt transformiert werden soll. Die Gültigkeit der Verknüpfung kann dabei entweder einen bestimmten Service oder aber einen Servicetyp, durch den viele gleichartige Services gruppiert werden können, umfassen. Im letzteren Fall reduziert dies die Entwicklungsarbeit signifikant.

Das Basismetamodell selbst befindet sich in der Hierarchie auf der Ebene M_2 . Es abstrahiert von konkreten Modellierungssprachen und enthält Konzepte, die für die Beschreibung von Services notwendig sind. Sind mehrere Servicemodelle mit dem Basismetamodell verbunden, so repräsentiert das daraus entstandene Konstrukt ein multiperspektivisches Servicemetamodell auf der Metamodellebene. Auf der Modellebene M_1 führt dies analog zu einem multiperspektivischen Servicemodell und die Durchführung der Transformation entsprechend der auf der Metaebene beschriebenen Verknüpfungen zu einem Informationsübergang zwischen den Modellen und somit zu einer durchgängigen Modellierung. Die nachfolgenden Abschnitte widmen sich der Herleitung und Beschreibung des Basismetamodells.

6.3.1 Basismetamodell

Die Konstruktion des Basismetamodells leitet sich direkt aus den Erkenntnissen der Untersuchung von Servicedefinitionen und Servicemodellen (vgl. Kapitel 3) ab. Die Erstellung des Basismetamodells erfolgt dabei sukzessive entsprechend des in Abbildung 6-2 dargestellten Schemas.

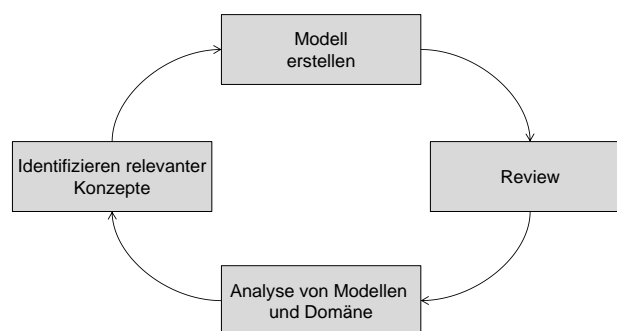


Abbildung 6-2: Ableitung des Basismetamodells (angelehnt an (Frank, Klein et al. 1998))

Im iterativen Entwicklungsprozess stellt die „Analyse von Modellen und Domäne die erste Phase dar. Die dafür zugrundeliegende Rechercharbeit wurde in (Augenstein, Ludwig et al. 2012) präsentiert und in Kapitel 3.2 zusammengefasst. Neben den Modellierungs- und Definitionsansätzen spielen bei der Bewertung des Modells auch die verwandten Ansätze aus Kapitel 4 eine wichtige Rolle. Allerdings stehen hier nicht Anwendbarkeit oder Vollständigkeit eine Rolle, sondern Unterschiede im Vorgehen und eine Bewertung hinsichtlich bestehender Vor- oder Nachteile. Im Sinne der Modellkonstruktion ist zunächst aber die

Frage zu klären, welche Arten von Modellelementen prinzipiell relevant sein können. Dadurch wird die Analyse der eingesetzten Sprachen ebenso notwendig, wie die der Modelle selbst. Die Art und Weise der Modellgestaltung, das Metaisierungsprinzip als wesentliches Abstraktionselement sowie die Definition einer konkreten Modellierungsmethode wurden bereits in Kapitel 2 erörtert.

Um das Identifizieren relevanter Konzepte zu erleichtern, muss eine Einschränkung relevanter Modellelemente erfolgen, die eine Sprache zur Verfügung stellt. So können prinzipiell *Entitäten*, *Operatoren* und *Verbinder* als Typen von Modellelementen identifiziert werden. Jede Modellierungssprache verfügt darüber hinaus über weitere, spezifische Elemente, die jedoch einem der drei Typen untergeordnet werden können. Dabei sind Entitäten solche Elemente einer Sprache, die Informationen aus der Domäne enthalten, die wesentliche Konzepte einer Domäne widerspiegeln und die hinsichtlich der Sprachdefinition eine zentrale Rolle spielen und in der Regel auch viele Verknüpfungen zu anderen Sprachelementen besitzen. Operatoren einer Modellierungssprache sind für den Kontrollfluß zuständig und dienen der logischen Verknüpfung von Entitäten oder spielen eine Rolle bei der Modellierung kausaler Zusammenhänge, indem sie als Auslöser für Reaktionen eingesetzt werden. Verbinder ermöglichen ein konsistentes Modell, indem sie die tatsächlichen Beziehungen zwischen Modellelementen repräsentieren. Dies unterstützt einerseits den Modellierer in seiner Arbeit durch Kenntlichmachung der Beziehung, trägt andererseits aber auch durch spezifische Ausprägung einer Verbindung zur vollständigen Modellierung bei.

Für eine Verknüpfung von Servicemodellen wird in erster Linie auf Entitäten einer Modellierungssprache zurückgegriffen. Außerdem müssen Verbinder in Betracht gezogen werden, da mit ihrer Hilfe Konzepte in einem Modell zueinander in Beziehung gesetzt werden können und das daraus entstehende, zusammengesetzte Konzept relevant für die weitere Verarbeitung ist. Die erste Phase ist abgeschlossen, sobald relevante Modelle und Modellelemente identifiziert sind.

Die Phase „Identifizieren relevanter Konzepte“ dient im Anschluss der Extraktion wesentlicher Elemente aus jedem Modell. Für das Basismetamodell als integrativem Modell sind diejenigen Konzepte relevant, die Services näher beschreiben. Je nach Perspektive und Zweck eines Modells sind solche Konzepte sehr unterschiedlich und teilweise auch widersprüchlich. Die Herausforderung bei der Auswahl der Konzepte besteht dabei in der richtigen Granularität. Sind die Konzepte zu grob oder zu abstrakt liefern sie keinen Mehrwert im Sinne der Informationsmodellierung. Sind die Konzepte dagegen zu fein bzw. zu detailliert, besteht die Gefahr, dass das zu konstruierende Basismetamodell zu viele Elemente

enthält und damit unübersichtlich oder gar unbeherrschbar wird. Neben dem Fokus auf Aspekte der Servicemodellierung wurde bei der Domänenanalyse explizit auf die Eignung des Modells für eine Anwendungsdomäne geachtet. Exemplarisch wurde hierzu die Logistikdomäne (s. Kapitel 8) herangezogen.

Auf Basis der identifizierten Konzepte wird ein Modell erstellt und einem Review unterzogen. Das Review bezieht sich sowohl auf die Domänenanalyse als auch auf die Brauchbarkeit des zuvor erstellten Modells im Kontext der Anwendung. Die Analyse der bestehenden Konstruktion hat dabei ergeben, dass keine unmittelbare Notwendigkeit besteht, Elemente der Anwendungsdomäne direkt in das Basismetamodell zu integrieren. Vielmehr hat sich gezeigt, dass entsprechende Einflüsse bereits durch konkret zu modellierende Services und deren Eigenschaften in ausreichendem Maße einfließen. Hinzukommt der Anspruch an die Flexibilität, Modularität und Erweiterbarkeit des Modells. Eine zu starke Verflechtung mit der Anwendungsdomäne würde diese Eigenschaften negativ beeinflussen. Daraus ergibt sich das in Abbildung 6-3 dargestellte Basismetamodell, in englischen Veröffentlichungen Common Service Model (CSM) genannt, zur Integration unterschiedlicher Servicemodelle in einen einheitlichen Rahmen. Innerhalb der Hierarchie in Abbildung 6-1 ist das Basismetamodell auf der Ebene M₂ einzuordnen und wird mit Sprachelementen des Metametamodells beschrieben. Das hier gezeigte Beispiel benutzt das Metametamodell Ecore als Sprache zur Modellierung.

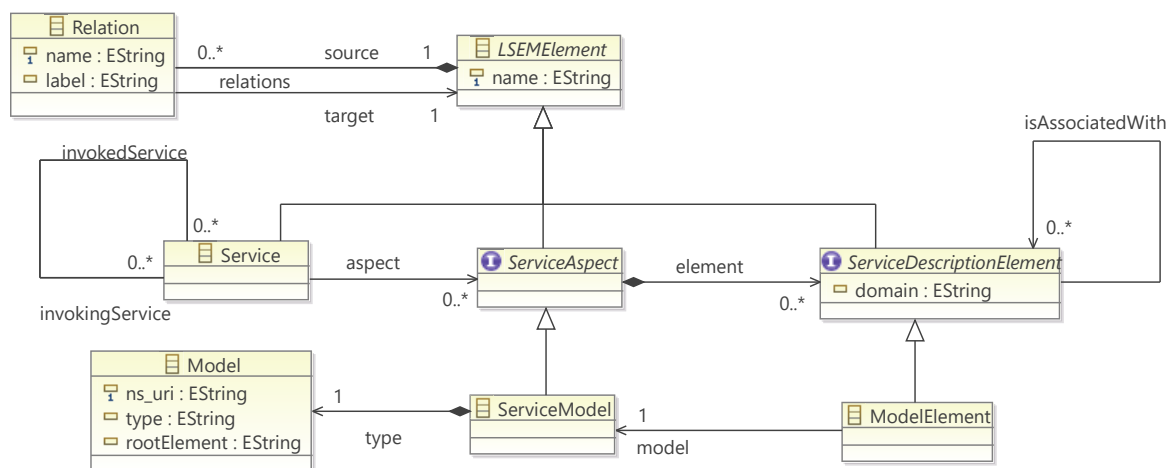


Abbildung 6-3: Basismetamodell

Im Vergleich zu den ersten Versionen des Basismetamodells wurde auf eine Einführung von Kategorien von Eigenschaften (z. B. Interface, Prozess, Ressource) zugunsten einer allgemeineren Beziehung zwischen Services und zugehörigen Modellen verzichtet. Der Grund für diese Entwurfsentscheidung liegt in der diffusen Definition von Services und der

Vielzahl von Modellen begründet. Insbesondere hat sich gezeigt, dass eine Zuordnung von Eigenschaften zu Kategorien nicht eindeutig ist. Das hätte zur Folge, dass entweder eine Zuordnung festgelegt oder eine Doppel- bzw. Mehrfachnennung erfolgen müsste. Eine feste Zuordnung führt jedoch zu Problemen hinsichtlich einer vollständigen Modellierung, womit die Gesamtkonsistenz u. U. gefährdet wäre. Auch kann nicht garantiert werden, dass bei einer künftigen Erweiterung die Inkonsistenzen vergrößert und das resultierende Modell somit nicht mehr einsatzfähig wäre und deshalb einer grundlegenden Überarbeitung bedürfen würde. Bei Doppel- oder Mehrfachnennung derselben Serviceeigenschaften in unterschiedlichen Kategorien verschlechtert sich sukzessive die Wartbarkeit des Modells und es wird der Separation of Concerns verletzt.

Im Vergleich zu den meisten Modellierungsansätzen (vgl. (Augenstein, Ludwig et al. 2012), 3.2 oder 4) erscheint das Basismetamodell sehr generisch und zunächst nicht geeignet, Services hinreichend präzise zu beschreiben. Mit nur sehr wenigen Elementen beschreibt es jedoch die Grundstruktur des SMF, indem Services, Modelle und deren Elemente zueinander in Beziehung gesetzt werden, mit dem Service als zentralem Element. Das Basismetamodell erlaubt es einen *Service* innerhalb einer Hierarchie von aufrufenden und aufgerufenen Services einzubetten. Ein Service ist das zentrale Element. Er wird durch verschiedene Modelle repräsentiert bzw. definiert, mittels derer Aussagen über den Zustand, das Verhalten und Beteiligte getroffen werden.

Die Beziehung zwischen einem Service und ihm zugeordneten Modellen wird über das Konzept des *ServiceAspect* (SA) beschrieben. Ein Service kann beliebig viele Aspekte besitzen. Ein Modell wird als eine Perspektive eines Service bzw. als *ServiceModel* definiert, wenn es einen spezifischen Aspekt eines Service beschreibt. Da auch Variantenbildung bei Modellen eine Rolle spielt bzw. unterschiedliche Modellierungssprachen eingesetzt werden können, um denselben Sachverhalt zu beschreiben, kann das Konzept SA stellvertretend für eine Reihe von Modellen stehen. Hinsichtlich der Modelltransformation repräsentieren SA sowohl Quell- als auch Zielmodelle. Zur Spezifikation eines konkreten *ServiceModels* wird diesem über das Konzept *Model* ein konkreter Modelltyp zugeordnet.

Analog zur Repräsentation von Modellen durch SA werden Modellelemente jeweils durch ein *ServiceDescriptionElement* (SDE) bzw. durch ein *ModelElement* repräsentiert. Dabei gilt, dass einem SA beliebige viele SDE zugeordnet werden, umgekehrt jedoch jedem SDE genau ein SA zugeordnet wird. Für die Modelltransformationen spielen Modellelemente eine zentrale Rolle, da mit ihnen der zu transformierende Inhalt modelliert wird. Aus diesem Grund müssen SDE zueinander in Beziehung gesetzt werden können, um den inhaltlichen Zusammenhang von Modellelementen zu beschreiben. Dabei gilt, dass ein SDE be-

liebig viele Beziehungen zu anderen SDE besitzen kann. Auf der Ebene der konkreten Modelle (*ServiceModel* und *ModelElement*) existiert nur eine unidirektionale Beziehung von *ModelElement* zu *ServiceModel*, um bei der Integration eine Zuordnung von Inhalt zu Quell- oder Zielmodell zu ermöglichen.

Als abstraktes Oberelement für alle genannten Elemente dient das *LSEMElement* dazu, gemeinsame Eigenschaften zu definieren, die für eine Modellierung mit diesen Elementen relevant ist. Dazu zählen bspw. eine grafische Repräsentation sowie die Modellierung von Beziehungen (*Relation*) zwischen den oben genannten Elementen. Das Element *Relation* sorgt in diesem Zusammenhang dafür, dass die zu modellierenden Beziehungen als Entitäten in Werkzeugen zur Verfügung stehen. Dies ist notwendig, um die konzeptuell beschriebenen Beziehungen zwischen den Modellelementen in einem Werkzeug nutzen zu können und damit ein Modell der Ebene M₂ (multiperspektivisches Servicemetamodell) zu konstruieren.

Diese Angaben und Zusammenhänge sind für die Grundstruktur des modellgetriebenen SMF-Ansatzes notwendig. Der nächste Abschnitt 6.3.2 zeigt, wie das Basismetamodell außerdem als Ausgangsbasis für eine Reihe von Modellen verwendet wird, die zum einen konzeptioneller, zum anderen technischer Natur sind und die ebenfalls mit Hilfe modellgetriebener Verfahren entwickelt werden. Insbesondere kann durch diesen Entwurf des Basismetamodells vermieden werden, dass durch spätere Ergänzungen oder durch Hinzufügen neuer Modellierungssprachen unterschiedliche, inkompatible Versionen existieren.

Schließlich wird ausgehend vom Basismetamodell in 6.4 eine Modellierungsmethode definiert, die beschreibt wie mittels des Basismetamodells Modelle und Modellelemente verschiedener Serviceaspekte verbunden werden können, so dass eine automatisierte Interpretation der Abhängigkeiten möglich ist, die zu einer durchgängigen Modellierung führt. Das Basismetamodell übernimmt dabei die Aufgabe, semantisch ähnliche Modellelemente miteinander zu verbinden. Die konkrete Umsetzung und Realisierung eines multiperspektivischen Modells wird in den Kapiteln 7 bzw. 9 detailliert erläutert.

6.3.2 Abgeleitete Modelle

Der durchgehend modellgetriebene Ansatz des SMF soll eine schnelle Entwicklung und Implementierung geeigneter Werkzeuge ermöglichen. Durch die enge Verzahnung konzeptueller Modelle und zugehöriger Werkzeuge über gemeinsame Metamodelle bei der Entwicklung sind die entstehenden Werkzeuge im Aufbau und in der Struktur spezifisch für den konkreten Anwendungsfall ausgerichtet. Ausgehend vom Basismetamodell können,

wie in Abbildung 6-4 angedeutet, mehrere Softwarekomponenten für den Umgang mit Modellen generiert werden.

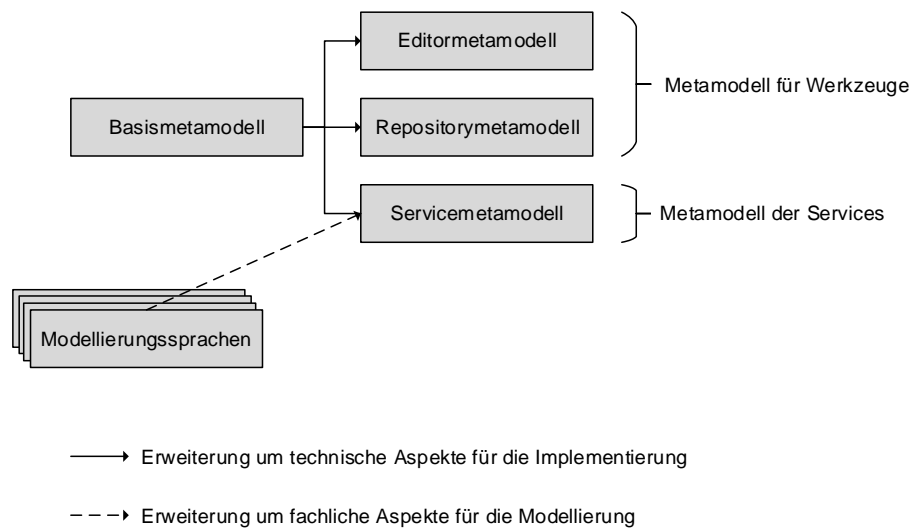


Abbildung 6-4: Integrierter Entwicklungsansatz

Für einen Editor, mit dessen Hilfe auf der Ebene M₂ ein multiperspektivisches Servicemetamodell entsteht, wurde dies in (Augenstein und Ludwig 2013b) gezeigt. Für die Entwicklung des Editors wurde hierzu das Relation-Konzept erweitert, so dass alle im Basismetamodell vorhandenen Beziehungen als Entitäten im Editor verfügbar sind (s. Abbildung 6-5 bzw. 9.3.4). Sämtliche im Basismetamodell vorhandenen Beziehungen zwischen Services, Modellen und Modellelementen sind als Modellierungsprimitive vorhanden.

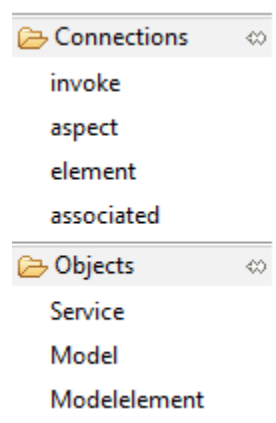


Abbildung 6-5: Modellierungsprimitive

Auf dieselbe Art und Weise wurde ein erweitertes Modell entwickelt, mit dessen Hilfe die vorhandenen Services und Modelle in einem Repository strukturiert abgebildet werden können. Der modellgetriebene Entwicklungsprozess ist jedoch auf die Funktionalität der Werkzeuge eingeschränkt. Für eine vollständige Entwicklung der Werkzeuge bedarf es darüber hinaus Softwarekomponenten, die bspw. die grafische Repräsentation implemen-

tieren. Erklärtes Ziel dabei ist, möglichst alle Werkzeugkomponenten weitestgehend automatisiert generieren zu lassen. Das Basismetamodell ist somit das zentrale Element der modellgetriebenen Servicebeschreibung, sowohl hinsichtlich der Entwicklung eines multiperspektivischen Servicemodells als auch hinsichtlich der dafür notwendigen Werkzeuge. Die modellgetriebene Entwicklung wird in allen Bereichen so eingesetzt, dass möglichst viele Artefakte auf dieselbe Art und Weise erzeugt werden und dass ein einheitlicher Entwicklungsprozess möglichst ohne größere, manuelle Eingriffe eingehalten werden kann.

6.4 Modellierungsmethode

Nachdem die wesentlichen Kernkonzepte des SMF vorgestellt sind, wird in diesem Abschnitt das konkrete Vorgehen beschrieben, mit dessen Hilfe die Modellierung eines multiperspektivischen Servicemetamodells durchgeführt wird. Wie bereits im Exkurs I vermerkt wird die adaptive Referenzmodellierung nach (Delfmann 2006) als Referenz für die Methodendefinition angesehen. Im Kern des SMF repräsentieren somit einzelne Servicemodelle Projektionen auf das zu entwickelnde Gesamtmodell. Im Gegensatz zu einem vollständigen Top-down-Ansatz wird jedoch kein integratives, eine Domäne vollständig erfassendes Metamodell entwickelt, um so die Schwierigkeit eines vollständigen Servicemodells zu umgehen. Essentiell für die strukturierte und konsistente Modellierung ist die Verwendung einer Modellierungsmethode, die sich gemäß der Definition 2-2 bzw. der Beschreibung aus 2.2 in weitere Komponenten untergliedern lässt:

- Modellierungstechnik bzw.
- Modellierungssprache (Syntax, Semantik, Notation) und
- Modellierungsvorgehen (Schritte Ergebnisse)

Abbildung 6-6 untergliedert die Sprach- und Modellbestandteile noch weiter auf und stellt Beziehungen zwischen den für die Modellierungsmethode notwendigen Bestandteilen her.

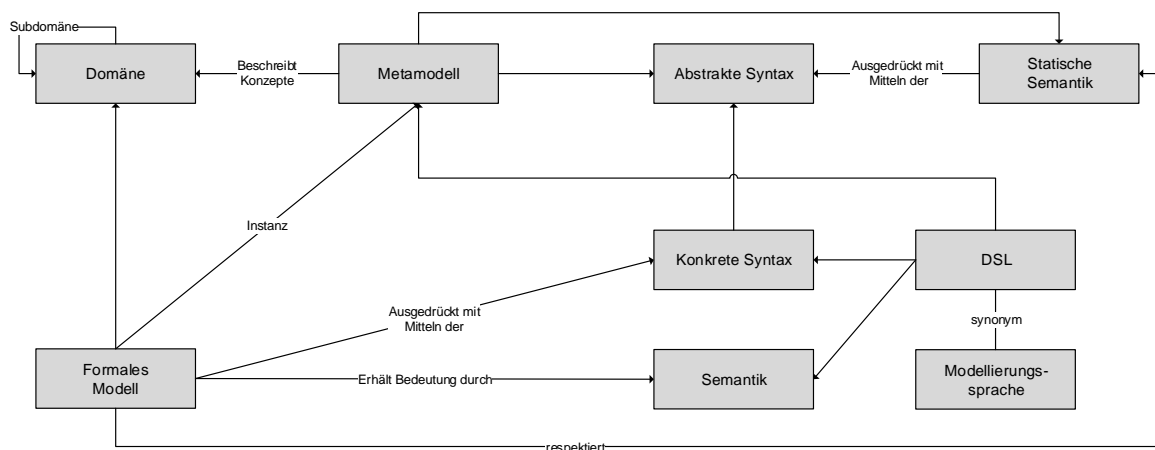


Abbildung 6-6: Modellkomponenten nach (Stahl, Völter et al. 2007)

Ausgangspunkt der zu erstellenden Modellierungsmethode bildet eine so genannte domänenspezifische Sprache (engl. Domain Specific Language; DSL) (vgl. (Becker, Pfeiffer et al. 2007, S. 68ff), (Pfeiffer 2008, S. 114), (Algermissen, S. 62)). Das schlussendlich generierte, multiperspektivische Servicemodell des SMF wird durch Anwendung einer DSL, deren Struktur über das Basismetamodell definiert ist, beschrieben. Dadurch wird erreicht, dass eine semantische Verbindung mit der Anwendungsdomäne hergestellt werden kann (vgl. (Pfeiffer 2007, S. 877)) und dass durch einen höheren Formalisierungsgrad eine Verbesserung der operativen Einsatzfähigkeit möglich ist, einem wesentlichen Ziel des Ansatzes (vgl. Kapitel 1).

Das Basismetamodell bildet vor diesem Hintergrund das Metamodell der DSL, die konkrete Syntax, oder auch Notation der Sprache, kann je nach Ausprägung sowohl textuell als auch grafisch in Form eines Knoten-Kanten-Modells gebildet werden. Kapitel 9.2 zeigt die Anwendung des SMF mit Hilfe einer grafischen Modellierung. Die Semantik der DSL, ergibt sich, im Unterschied zur statischen Semantik in Abbildung 6-6, durch die Anwendung des SMF als Mittel zur Konstruktion eines multiperspektivischen Servicemetamodells (Formales Modell in Abbildung 6-6) für eine gegebene Anwendungsdomäne (exemplarisch die Logistik in Kapitel 8). Die abstrakte Syntax stellt die grundlegenden Modellierungskonstrukte, also zur Verfügung stehende Modellelemente und Beziehungen, im Metamodell bereit. Sie wird über die Metametasprache des Metametamodells (Strahinger 1996, S. 23) beschrieben und über eine (austauschbare) Form der konkreten Syntax dargestellt. Die statische Semantik schränkt dabei die Verwendung der Modellkonstrukte ein, indem sie einzelnen Modellelementen eine spezifische Bedeutung verleiht und gültige Beziehungen zwischen diesen Elementen definiert. Sie definiert somit wann Syntaxelemente korrekt bzw. falsch verwendet werden oder verhindert dies sogar. Beispielsweise wird über die statische Semantik des Basismetamodells definiert, dass ein Service eine beliebige Zahl von Aspekten besitzen kann und Aspekte sich aus einer Menge von Servicebeschreibungselementen zusammensetzen. Es widerspricht jedoch der Semantik, dass ein Servicebeschreibungselement direkt einem Service zugeordnet werden kann. Die genannten Elemente des Basismetamodells stellen zugleich Bestandteile der abstrakten Syntax dar.

Das Vorgehen zur Modellierung als zweitem Teil der Modellierungsmethode beinhaltet notwendige Schritte zum Umgang mit den sprachlichen Elementen und definiert Ergebnisse zu den Schritten und damit das Entstehen der Modelle. Das hier angewandte Vorgehen wird schematisch in Abbildung 6-7 dargestellt. Für jede Ebene der Modellhierarchie (rechts) werden Beispiele zur Modellierung (links) sowie Ergebnisse (Mitte) der Analyse (A) und Generierung (G) von Artefakten dargestellt.

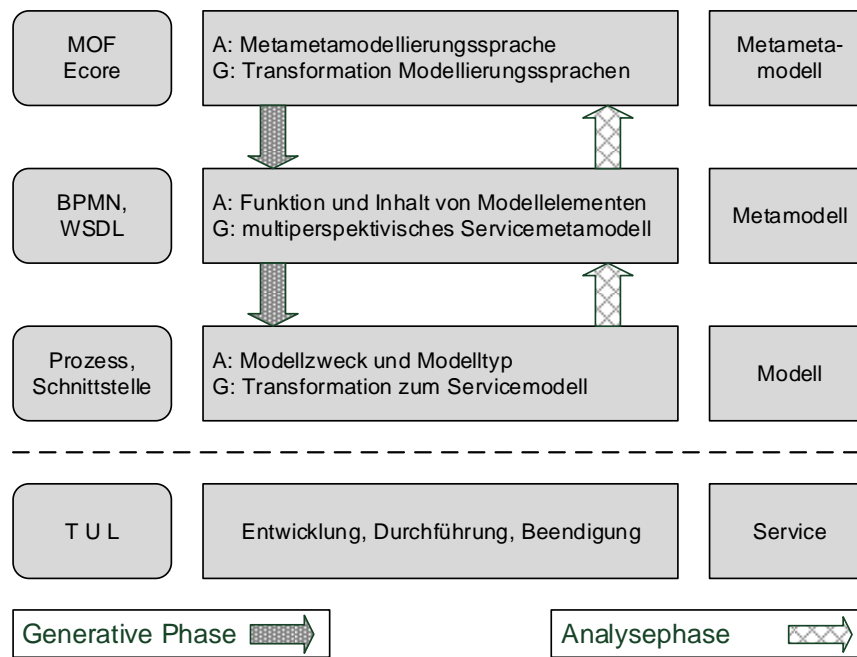


Abbildung 6-7: Modellierungsvorgehen

Die einzelnen Schritte des Vorgehens zur Modellierungsmethode sind in zwei Phasen aufgeteilt. Ein erste, analytische Phase, in der relevante Modelle, Modellelemente und zu modellierende Aspekte untersucht bzw. ausgewählt werden und eine zweite, generative Phase, in der das multiperspektivische Servicemetamodell sukzessive durch Hinzufügen neuer Modelle und Modellelemente auf der Ebene M_2 entwickelt wird. Wesentlich für das Vorgehen ist, dass die Analyse auf der Instanzebene (M_1) beginnt und auf der Ebene M_3 endet, während die zu realisierenden Artefakte in umgekehrter Reihenfolge entstehen, so dass das Vorgehen dann beendet ist, wenn auf der Ebene M_2 ein multiperspektivisches Servicemetamodell beschrieben ist und die Transformation auf der Ebene M_1 angestoßen werden kann.

Für eine umfängliche Darstellung eines Service werden gemäß den hier vorgestellten Voraussetzungen unterschiedliche Modelle zur Beschreibung einzelner Aspekte herangezogen. Eine konsistente Abbildung kann nur dann erfolgen, wenn zwischen den einzelnen Aspekt-Modellen Beziehungen hergestellt und umgesetzt werden, d.h. wenn gewährleistet wird, dass Modelle bzw. Modellelemente, die äquivalente Informationen beinhalten auch in Beziehung zueinander gesetzt werden. Im Zuge der Servicemodellierung muss auf der Ebene M_1 daher eine Analyse der zu beschreibenden Tatsachen erfolgen. Daraus lassen sich direkt Modellierungssprachen ableiten, die zur Beschreibung der benötigten Aspekte geeignet sind. Die wesentliche Fragestellung lautet: *Was soll modelliert werden und mit welchen Modellen soll es modelliert werden?* Demzufolge sind die Ergebnisse der Ebene M_1 die zur Modellierung notwendigen Modellierungssprachen sowie die zu modellieren-

den Sachverhalte. Auf Ebene M_2 kann im Anschluss daran analysiert werden, welche Konstrukte der gewählten Modellierungssprachen notwendig sind, um die Sachverhalte korrekt umzusetzen. Außerdem muss geklärt werden, ob die zu modellierenden Aspekte hinsichtlich der beinhalteten Eigenschaften eine Überdeckung aufweisen, d.h. ob in unterschiedlichen Aspekten äquivalente Informationen potenziell enthalten sind. Die wesentliche Fragestellung dazu lautet: *Welche Modellelemente einer Sprache sind notwendig und existieren Abhängigkeiten zwischen Modellelementen unterschiedlicher Modellierungssprachen für die konkrete Modellierungsaufgabe?* Das Ergebnis auf Modellebene M_2 besteht in einer Menge relevanter Modellelemente der verwendeten Sprachen sowie in einer daraus abgeleiteten Teilmenge von Modellelementen mit wechselseitigem, inhaltlichen Bezug auf der Instanzebene. Als letzten Schritt muss zudem noch sichergestellt sein, dass das Metamodell jeder im ersten Schritt ermittelten Modellierungssprache vorhanden und mit Hilfe des gewählten Metametamodells (s. 6.1) beschrieben ist. Das Ergebnis dieses Schrittes fasst die vorhandenen Metamodelle zusammen und benennt die Fehlenden, auch solche, die noch nicht mittels der gewählten Sprache beschrieben sind. Damit ist zugleich die analytische Phase beendet und es schließt sich die generative Phase der Modellierungsmethode an.

Ausgehend von den Ergebnissen der Analysephase ist die Entwicklung des multiperspektivischen Servicemodells, sowohl auf Ebene M_2 wie M_1 , das zentrale Ergebnis der generativen Phase. Zunächst werden alle noch fehlenden oder in einer anderen Sprache beschriebenen Metamodelle erzeugt, bzw. angepasst. Bei mehrmaliger Anwendung des SMF werden dabei sukzessive neue Modellierungssprachen hinzugefügt und bilden die Grundlage der Servicebeschreibung. Bei Nichtvorhandensein kann das gewünschte Metamodell über eine M2M-Transformation in die Sprache des Ziel-Metametamodells übertragen werden. Die dadurch gewonnene, syntaktische Vergleichbarkeit der Elemente der Modellierungssprachen wird nun genutzt, um Modelle und Modellelemente mit dem Basismetamodell zu verbinden. Dabei entstehen zwischen dem Basismetamodell und den Modellen $M_x \{E_1, E_2\}$ und $M_y \{E_3, E_4\}$ für einen Service S Beziehungen der Art:

- $S \Leftrightarrow M_x$ und $S \Leftrightarrow M_y$, wobei \Leftrightarrow eine unidirektionale Beziehung (Aspektrelation) zwischen einem Service und Modellen repräsentiert, die jeweils eine Perspektive eines Service repräsentieren.
- $M_x \rightarrow E_1$, $M_x \rightarrow E_2$, $M_y \rightarrow E_3$, $M_y \rightarrow E_4$, wobei \rightarrow eine unidirektionale Beziehung (Elementrelation) zwischen einem Modell und den für die Übertragung von Informationen relevanten Modellelementen repräsentiert.

- $M_x.E_1 \Leftrightarrow M_y.E_3$, wobei \Leftrightarrow eine typisierte, bidirektionale Beziehung (Assoziationsrelation) zwischen zwei Modellelementen repräsentiert, die eine gegenseitige Abhängigkeit der Modellelemente hinsichtlich der enthaltenen Informationen anzeigt.
- Zusätzlich kann auch eine Beziehung zwischen zwei Services S_1 und S_2 modelliert werden, mit $S_1 \langle \rangle S_2$, wobei $\langle \rangle$ eine bidirektionale Beziehung (Hierarchierelation) zwischen zwei Services repräsentiert, bei der ein Service im Rahmen einer Komposition Bestandteil eines anderen Service ist und umgekehrt.

Die Angabe von Beziehungen zwischen Modellelementen unterschiedlicher Modellierungssprachen (Assoziationsrelation) erfolgt mit Hilfe der in den Modellelementen enthaltenen Attribute. So kann bspw. das Modellelement Task der BPMN (s. 6.1) Attribute beinhalten, die einen Task identifizieren (id), einen Namen zuweisen (name) oder eine Dauer spezifizieren (duration). In der Assoziationsrelation werden für Quell- und Zielelement diejenigen Attribute spezifiziert, die äquivalente Informationen beinhalten sollen. Im einfachsten Fall sind alle Attribute beider Elemente identisch und somit die Elemente quasi identisch, so dass eine Vereinfachung der Typisierung erfolgen kann. In diesem Fall spricht man von einer Äquivalenzrelation und spezifiziert so die Identität zweier Merkmale hinsichtlich der enthaltenen Informationen. Durch Einbeziehung der Fachdomäne kann zusätzlich die Präzision der Verbindung von Modellen und Modellelementen erhöht werden. Dadurch wird nicht nur beschrieben, dass bspw. zwei Attribute dieselben Informationen beinhalten, sondern auch welcher Art die Information ist.

Sind für alle in der ersten Phase identifizierten Modellelemente solche Beziehungen definiert, ist das Servicemetamodell als Ergebnis dieses Schrittes konstruiert. Dieses dient dann als Grundlage für eine automatisierte Transformation auf der Instanzebene und somit als Blaupause eines multiperspektivischen Servicemodells. Im letzten Schritt der generativen Phase muss schließlich ausgewählt werden, welcher Service mit Hilfe des zuvor konstruierten Metamodells auf der Instanzebene beschrieben werden soll. Möglich ist sowohl die Angabe eines spezifischen Service, einer Menge von Services oder eines bestimmten Servicetyps. Letztere Option bietet die Möglichkeit eine standardisierte, multiperspektivische Sicht auf eine Reihe von Services zu definieren. Auf diese Weise können wiederkehrende Modellierungsmuster ohne erneuten Aufwand im SMF persistiert werden. Die Generierung des multiperspektivischen Servicemodells kann dann auf Basis derselben Vorlage mit den existierenden Modellen ohne Analyseaufwand angestoßen werden.

7 Werkzeuggestützte Servicegestaltung

Die werkzeuggestützte Servicegestaltung beschreibt eine mögliche Ausprägung der im vorherigen Kapitel beschriebenen Modellierungsmethode. Es werden hierzu zunächst konkrete Anforderungen (7.1) an die Umsetzungsvariante beschrieben sowie eine Form der Persistierung (7.2). Für eine graphische Herangehensweise beschreibt 7.3 die sprachbasierte Servicemodellierung mit Hilfe eines Knoten-Kanten-Modells, dessen Nutzung zur Entwicklung des multiperspektivischen Servicemodells in 7.4 beschrieben wird. Die werkzeuggestützte Servicegestaltung ist dabei als Zwischenstufe zur prototypischen Implementierung zu sehen. Aus einer Zahl möglicher Varianten zur Umsetzung, die den Anforderungen und Restriktionen genügen werden die Konzepte der modellgetriebenen Servicebeschreibung so konkretisiert, dass eine anwendungsnahe Beschreibung der wesentlichen Komponenten zur Umsetzung gegeben ist.

7.1 Anforderungen

Im Gegensatz zu den Anforderungen an das gesamte Rahmenwerk in 5.2 beziehen sich die hier genannten Anforderungen auf eine konkrete Umsetzungsvariante des SMF. Die allgemeinen Anforderungen besitzen jedoch auch weiterhin ihre Gültigkeit. Im Rahmen der Konkretisierung sind jedoch insbesondere an die Persistierung und an die Sprachentwicklung zusätzliche Anforderungen zu erheben, die den Lösungsraum weiter einschränken und konsistentes Lösungskonzept ermöglichen.

Über das SMF hinaus sorgt die Persistenzkomponente (Repository) dafür, dass auch aus anderen Werkzeugen im Rahmen des LSEM konstruierte Modelle sowie die zugehörigen Metamodelle zentral verwaltet werden können. Aus architektureller Sicht sind daher die folgenden Anforderungen zu sehen: Die Konzeption des Repository setzt die Integrationsfähigkeit für andere Werkzeuge bzw. das Anbieten einer API voraus. Idealerweise wird das Repository nur mit standardisierten Komponenten implementiert bzw. es werden Open-Source-Komponenten eingesetzt. Im Zusammenhang mit dem SMF und der modellgetriebenen Servicebeschreibung ist auch eine modellgetriebene Entwicklung des Repository wünschenswert, so dass auch durch diese Anforderung die Umsetzung weiter konkretisiert werden kann. Schließlich muss das Repository mittels einer Client / Serverarchitektur implementiert werden, um alle Informationen und Artefakte zentral speichern zu können. Der Client übernimmt dabei die Aufgaben der Authentifizierung und Synchronisation der Informationen mit dem Server. Aus informationstechnischer Sicht muss das Repository in der Lage sein, alle darin enthaltenen Artefakte eindeutig zu identifizieren, diese zu persistieren und auf Anfrage bereitzustellen. Dies trifft sowohl auf Modelle, als Artefakte der

unterschiedlichen Werkzeuge, als auch auf Metamodelle, als Artefakte der Werkzeugentwicklung, zu. Dementsprechend müssen Identifizierungsmerkmale für Modelle und Metamodelle vorhanden sein. Auch eine Versionierung und damit verbundene Archivierung alter Modelle muss im Rahmen eines Repository implementiert sein, um eine konsistente Modellierung über einen längeren Zeitraum hinweg zu garantieren. Vergleichbare Anforderungen werden auch von (Buchwald, Tiedeken et al. 2010) bei der Entwicklung von Repositories gefordert. Analog zu den bereits genannten, stehen bei den Autoren die Werkzeugintegration für einen zentralen Zugriff und Verwaltung von Informationen, aber auch das Repository als Basis für sämtliche Informationen und Artefakte im Vordergrund. Bezüglich der inhaltlich getriebenen Anforderungen stellen (Buchwald, Tiedeken et al. 2010) Repositories die Basis für einen Lebenszyklus-basierten Ansatz dar und fordern ebenso eine Versionierung sowie eine Archivierung als Kernbestandteil.

Mit der Sprachentwicklung eng verbunden ist die Umsetzung der Modellierungsmethode. Die zu definierende Sprache muss es ermöglichen, mit Hilfe des Basismetamodells unterschiedliche Modelltypen zu integrieren und das multiperspektivische Servicemodell zu konstruieren. Aufgrund der Vorgaben aus Kapitel 5 muss die Sprache auf einem Modellansatz basieren und eine modellgetriebene Entwicklung unterstützen. Die Sprache muss auch auf Basis des verwendeten SMF-Metametamodells darstellbar sein, wodurch sich Anforderungen an die Wahl des Metametamodells ergeben, die jedoch bereits zu den allgemeinen Anforderungen zu zählen sind. Für eine leichte Erlernbarkeit sollte die zu entwickelnde Sprache einen grafischen Ansatz (z. B. ein Knoten-Kanten-Modell) verfolgen, allerdings auch eine maschinelle Auswertung der resultierenden Modelle ermöglichen. Dementsprechend muss sie eine textuelle (auf XML-basierende) Serialisierung unterstützen. Bezüglich des zu modellierenden Inhalts muss die Sprache geeignet sein, insbesondere unterschiedliche Beziehungen auf Anwendungsdomänen-bezogener, d.h. fachlicher, Ebene zwischen verschiedenen Entitäten zu modellieren und die Konsistenzwahrung auf Basis des ihr zugrundeliegenden Metamodells zu garantieren. Aufgrund der fachlichen Beziehungen muss die Sprache die Anwendungsdomäne, innerhalb deren das SMF angewendet werden soll, so unterstützen, dass Konzepte der Anwendungsdomäne Bestandteil der Sprache werden. Das bedeutet nicht, dass solche Konzepte fest in das Metamodell der Sprache integriert sein müssen, aber dass es möglich sein muss, Begriffe der Anwendungsdomäne zur Beschreibung von Entitäten oder Beziehungen zwischen Entitäten zu nutzen.

7.2 Persistente Modellierung

Der Zugriff auf sämtliche Modelle ist ein wesentlicher Aspekt des SMF. Notwendigerweise müssen zur Integration alle zu einem Service gehörenden Modelle bei der eigentlichen Durchführung verfügbar sein. Ein Bestandteil des SMF bildet daher eine Persistenzkomponente, die das Lesen und Schreiben von Modellen sowie jederzeitigen Zugriff auf die Modelle ermöglicht. Für das SMF, aber auch in Bezug zum LSEM-Rahmen wird ein Repository-Ansatz verwendet (vgl. auch (Augenstein, Ludwig et al. 2012)). Das Repository verwaltet sowohl die zu speichernden Objekte (Services und Modelle) als auch Metainformationen und Beziehungen zu diesen Objekten und ist ein entscheidendes Element bei der Umsetzung einer durchgehenden Modellierungsmethodik (Buchwald, Tiedeken et al. 2010). Dazu wird eine entsprechende Komponente (s. 9.3.3) nicht nur für das SMF, sondern auch für alle im LSEM eingesetzten Werkzeuge implementiert, um so eine Integration und zentralen Zugriff bzw. eine zentralisierte Modellverwaltung zu ermöglichen. Es bildet die Basis für alle entstehenden Artefakte und einen Kernbaustein zur Informationswiederverwendung aus den unterschiedlichen Modellen in einem multiperspektivischen Servicemodell. Die wesentlichen Aufgaben des Repositories sind daher die sichere Speicherung bzw. Archivierung von modellierten Services sowie das Anbieten einer Versionierung für die einzelnen Modelle. Archivierung und Versionierung sind in diesem Zusammenhang eng verbunden, da eine Versionierung nur aufgrund der Speicherung auch älterer Modell- bzw. Serviceversionen möglich ist. Zur Umsetzung dieser Aufgaben sind drei Komponenten bei der Umsetzung entscheidend: Schnittstellen zur Kommunikation, Modell- und Serviceverwaltung.

Die durchgängige Modellierung findet über einen Informationsaustausch zwischen den Werkzeugen statt, der durch Schreiben und Lesen der Informationen im Repository realisiert wird. Mittels Schnittstellen werden diese Zugriffe implementiert. Eine Schnittstelle ist so konzipiert, dass neue Werkzeuge ihre Modelle als Metamodell für die Servicebeschreibung im Repository registrieren können. Die Schnittstellen ermöglichen dann einen lesenden und schreibenden Zugriff auf die zu dem Werkzeug gehörenden Serviceinformationen. Für einen lesenden Zugriff kann ein Service entweder direkt oder über eine Liste aller bisher mit einem Werkzeug modellierten Services ausgewählt werden

Die Modell- bzw. Serviceverwaltung unterscheiden sich hauptsächlich in der Art der zu speichernden Artefakte. Während in der Modellverwaltung des Repositories alle Metamodelle der Werkzeuge gespeichert werden, werden in der Serviceverwaltung Services und zugehörige Servicemodelle gespeichert. Die Modellverwaltung erfüllt zum einen den Zweck der Registrierung der Werkzeuge im Repository und zum anderen werden alle bis-

lang genutzten Metamodelle (sowie unterschiedliche Versionen) zur Servicebeschreibung gespeichert. Dementsprechend können dort auch Metamodelle hinterlegt sein, für die noch kein integriertes Werkzeug existiert, etwa für den Fall, dass Werkzeuge von Drittanbietern genutzt werden. Analog zur Modellverwaltung existiert die Serviceverwaltung als der Bestandteil des Repository, indem die modellierten Informationen vorgehalten werden. Jeder lesende oder schreibende Zugriff eines Werkzeugs über die Schnittstellen liefert einen bestimmten Ausschnitt aus der Serviceverwaltung. Das Repository verfügt somit über zwei inhaltlich getrennte Bereiche, die für die Metamodelle (Modellverwaltung) bzw. Modelle (Serviceverwaltung) verantwortlich sind. Während die Modellverwaltung quasi auch die Werkzeugverwaltung darstellt und eine Zuordnung von Metamodellen zu Werkzeugen implementiert, werden in der Serviceverwaltung die konkret erstellten Modelle sowie die entsprechenden Services verwaltet. Die Verknüpfung der Metamodelle mit dem Basismetamodell zum multiperspektivischen Servicemetamodell wird ebenfalls in der Serviceverwaltung persistiert, da dieses die Vorlage zur Generierung des Servicemodells ist.

7.3 Sprach-basierte Servicemodellierung

Die Erstellung des multiperspektivischen Servicemetamodells aus einer Verknüpfung des Basismetamodells mit Elementen von Modellierungssprachen wird konkret mit Hilfe einer grafischen, domänenspezifischen Sprache umgesetzt. Als Repräsentation (vgl. Abbildung 6-6) bzw. konkrete Syntax wird ein Knoten-Kanten-Modell eingesetzt. Man kann in diesem Fall auch von der Entwicklung einer konzeptuellen Modellierungssprache sprechen (vgl. (Gehlert 2007, S. 34)), da zwar eine formalisierte (abstrakte) Syntax in Form des Basismetamodells vorliegt, die Semantik der Sprache jedoch erst durch Einbeziehung der Anwendungsdomäne vollständig ist. Die resultierende DSL als Knoten-Kanten-Modell nutzt somit das Basismetamodell als eigenes Metamodell zur Definition der abstrakten Syntax und verwendet Elemente der Anwendungsdomäne zur näheren Beschreibung der Beziehung zwischen Elementen. Realisiert wird die Nutzung der DSL im SMF-Kontext durch die modellgetriebene Entwicklung eines Editors, der auf diese DSL zugeschnitten ist (vgl. auch (Augenstein und Ludwig 2013b) bzw. 9.3.4). Zur Entwicklung der DSL und des Editors wird zunächst das Basismetamodell (s. auch Abbildung 6-3) um spezifische Relationen erweitert. Diese sind:

- Hierarchierelationen, zur Kennzeichnung von Abhängigkeiten zwischen Services,
- Aspektrelationen, zur Kennzeichnung der Beziehung zwischen Modell und Service,
- Elementrelationen, zur Kennzeichnung der Beziehung zwischen Modell und Modellelementen sowie

- Assoziationsrelationen, zur Kennzeichnung von Beziehungen zwischen Modellelementen

Die entsprechenden Relationen sind eng verbunden mit den Modellierungsprimitiven „invoke“ (Hierarchie), „aspect“ (Aspekt), „element“ (Element) und „associated“ (Assoziation) (vgl. auch Abbildung 6-5) des generierten Editors. Ein Beispiel für die Modellierung mit Hilfe der DSL ist in Abbildung 7-1 dargestellt.

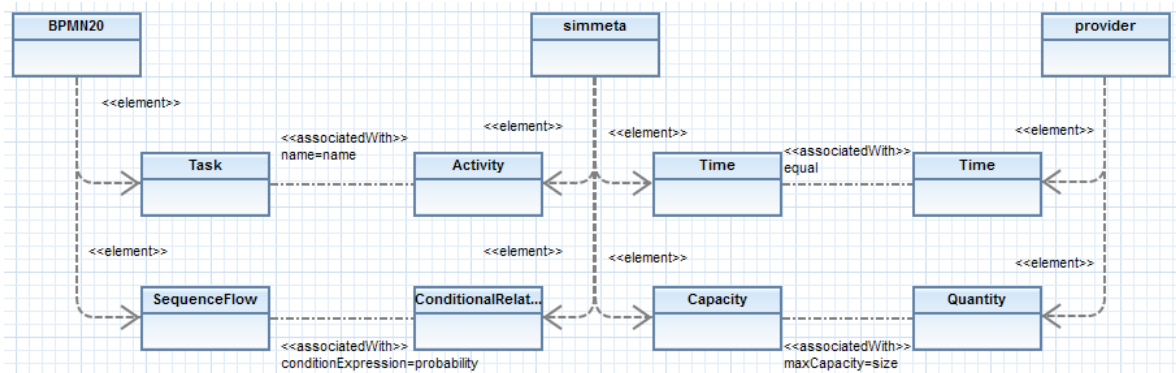


Abbildung 7-1: Beispiel eines modellierten Service

Die jeweils zum Tragen kommenden Modellierungsprimitive bzw. Relationen sind in der Abbildung mit Hilfe von Guillemets (<< >>) dargestellt. Unterschiedliche Relationen (im Bild etwa Element- und Assoziationsrelationen) werden durch unterschiedliche Linienarten repräsentiert. Assoziationsrelationen können darüber hinaus auch annotiert werden (Text ohne Guillemets), um eine exakte Beschreibung der Beziehung zwischen Modellelementen zu definieren. Durch Nutzung von Begriffen der Anwendungsdomäne kann so eine präzise Abbildung zwischen Elementen der Quell- und Zielmodelle erfolgen und bei der anschließenden Modellintegration umgesetzt werden. Modelle wie auch Modellelemente werden mittels Rechtecken dargestellt. Ein Vertauschen von Modellen und Modellelementen ist jedoch nicht möglich, da die Implementierung des Editors die Zulässigkeit von Kanten zwischen Elementen prüft, so dass bspw. keine Aspektrelationen zwischen Modellen und Modellelementen definiert werden können. Sämtliche Validierungsmöglichkeiten des Editors basieren dabei auf der Integration des Basismetamodells in den Editor und ermöglichen so eine strikte Einhaltung des Metamodells. Gleichfalls ist somit gewährleistet, dass das resultierende Modell tatsächlich das Servicemetamodell repräsentiert und dessen darauffolgende Interpretation zum Servicemodell führt.

Diese Umsetzung der Modellierungsmethode basiert nicht auf einer strikten Anwendung bereits existierender Modelltransformationsansätze. Vielmehr basiert dieser Ansatz auf dem so genannten „Model Weaving“ (s. 0), bei der Korrespondenzen zwischen Modellen

definiert und ausgewertet werden. Im Unterschied zu Transformationsansätzen basiert der Ansatz im SMF nicht auf einem syntaktischen Vergleich der beteiligten Modelle auf Metamodellebene. Es entsteht durch die manuelle Modellierung des Servicemetamodells auch ein im Vergleich höherer Aufwand, da Syntaxvergleiche vollautomatisiert durchgeführt werden können. Der Mehrwert im SMF entsteht durch die Definition fachlicher Beziehungen zwischen Modellen und Modellelementen. Auf diese Weise können zum einen auch Modelle integriert werden, denen unterschiedliche Zwecke zugrunde liegen und bei denen damit ein rein syntaktischer Vergleich nicht zielführend ist. Die umrissene Natur eines Service führt jedoch zu einem differenzierten Bild, das nur mit heterogenen Modellen vollständig erfasst und beschrieben werden kann. In der Regel sind dann existierende Modelltransformationsansätze nicht oder nur bedingt geeignet und ein hoher Grad an manueller Interaktion wird notwendig. Zum anderen wird über den SMF-Ansatz in erster Linie der Inhalt von Modellen zusammengeführt und nicht die Modelle selbst. Über das Servicemetamodell wird spezifiziert woher diese Inhalte kommen und wie die Inhalte, fachlich gesehen, zueinander in Beziehung stehen. Auf diese Weise können auch Fehler in den Teilmodellen identifiziert werden, da sich der Modellierer des Servicemetamodells über den Inhalt bzgl. der Anwendungsdomäne im Klaren sein muss. Im Sinne der Transformation ist es auch nicht notwendig, dass eine bestimmte Zielsprache definiert sein muss, in der das resultierende Modell beschrieben ist. Stattdessen werden in einem Schritt mehrere Modelle verändert, indem Informationen hinzugefügt werden. Insgesamt bleibt das Servicemetamodell anpassbar, indem bspw. zu einem späteren Zeitpunkt neue Modelltypen hinzugefügt werden können und das aktualisierte Modell erneut ausgewertet werden kann, sodass auch unterschiedliche Versionen definiert werden können. Die Definition des Servicemetamodells zunächst auf der Metamodellebene führt außerdem dazu, dass das resultierende Servicemetamodell auf einen spezifischen Service, auf einen spezifischen Servicetyp oder auch auf eine Liste gleichartiger Services angewendet werden kann. Das Servicemetamodell ist vor diesem Hintergrund als Blaupause für die Modellintegration zu verstehen und wird zusammen mit den bereits vorhandenen Modellen innerhalb der Serviceverwaltung des Repositorys gespeichert.

7.4 Modellintegration

Die Modellintegration ermöglicht die transparente Informationsmodellierung über die Grenzen einzelner Modelle hinweg. Auf Basis der persistierten Modelle können bereits modellierte Informationen wiederverwendet und im Kontext neuer Modelle bzw. neuer Serviceaspekte eingesetzt werden. Dazu ist es notwendig, dass der Informationsübergang

zwischen Modellen realisiert wird. Bereits in Kapitel 2 wurden Modelltransformationen und das Model Weaving, also das Verknüpfen von Modellen, als notwendige Grundlagen erarbeitet. Die nachfolgende Abbildung 7-2 verdeutlicht die Zusammenhänge.

Für gegebene Modelle M_I und M_O , die bereits vorhandene Informationen enthalten (M_I) bzw. denen Informationen hinzugefügt werden sollen (M_O), wird ein mit einem Transformationsmodell vergleichbares Modell M_T erzeugt, das als Ausführungsbeschreibung der durchzuführenden Transformation dient (s. 7.3). Allerdings werden dazu Konzepte der jeweiligen Metamodelle MM_I bzw. MM_O verwendet und zueinander in Beziehung gesetzt, um einen Übergang gemäß der in 6.4 beschriebenen Modellierungsmethode zu beschreiben. M_T selbst folgt dabei MM_T , das in 6.3.1 als Basismetamodell definiert wurde. Insgesamt stellen sowohl die Definition von M_T als auch dessen Interpretation bei der Ausführung die wesentlichen Schritte dar.

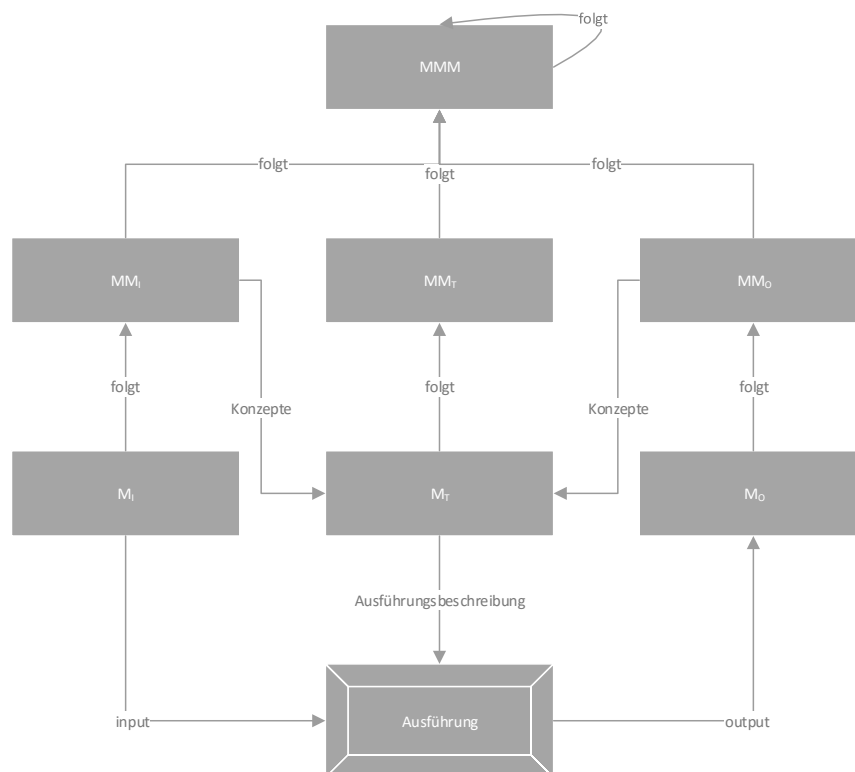


Abbildung 7-2: Modellabhängigkeiten

Die Komplexität von M_T kann durch eine Selektion möglicher Ausprägung des morphologischen Kastens in Tabelle 2-1 in 0 gezeigt werden. Das Ergebnis dazu zeigt Tabelle 7-1. Die grau hinterlegten Ausprägungen sind solche, die bei der Transformation von Modellen innerhalb des SMF auftreten können. Zunächst fällt dabei auf, dass endogene Transformationen kein Anwendungsbeispiel darstellen. Dies impliziert, dass bei der Verarbeitung von Informationen aus Modellen Quell- und Zielmodelle grundsätzlich durch unterschiedliche

Modellierungssprachen definiert sind (exogene Transformation bzw. multiple Metamodelle). Der Grund liegt im Zweck des SMF, nämlich in der multiperspektivischen Beschreibung eines Service durch Integration verschiedener Aspekte bei der Beschreibung. Endogene Transformationen wie bspw. eine Verfeinerung eines bestehenden Modells sind daher im Vorfeld bzw. unabhängig vor der Modellintegration durchzuführen und das entstandene Modell im Anschluss zu integrieren.

Einbezogene Modelle	1:1	1:n	n:1	n:m		
Einbezogene Metamodelle	1 (endogen)		n (exogen)			
Endogene Transformation	Optimierung	Refactor-ing	Vereinfachung	Normalisierung	Anpassung	Verfeinerung
Exogene Transformation	Synthese		Reverse Engineering		Migration	
Abstraktionsstufe	Horizontal			Vertikal		
Anwendungsbereich	In-place			Out-place		
Transformationsgegenstand	Syntax			Semantik		
Automatisierungsgrad	Manuell		Teil-automatisiert		Voll-automatisiert	
Komplexitätsgrad	einfach			komplex		
Invariante	Struktur	Verhalten	Integrität	Korrektheit	Vollständigkeit	
Paradigma	deklarativ (was)		imparativ (wie)		funktional (was)	
Anforderungen	vollständig	wohlgeformt	determiniert	korrekt	endlich	
Richtungsgebundenheit	unidirektional			bidirektional		
Prägnanz	wenige Konstrukte			viele Konstrukte		
Regelausführung	interpretiert			kompiliert		
Flexibilität	eingeschränkt		erweiterbar		konfigurierbar	

Tabelle 7-1: Transformationseigenschaften im Rahmen des SMF

Daraus ergibt sich, dass das Ergebnis der Transformation nicht innerhalb eines Modells, sondern in einem neuen Modell persistiert wird (out-place) und auch dass es sich bei der Art der Transformation um eine Migration handelt, bei der Informationen zwischen unterschiedlichen Modellen migriert werden. Die Abstraktionsstufe ist dabei variabel und kann sowohl horizontal als auch vertikal sein. Da der Zweck des SMF in der transparenten Informationsmodellierung begründet ist, ist der Transformationsgegenstand die Syntax, also letztlich die Beschreibung desselben Sachverhalts in verschiedenen Modellen. Die Semantik soll bei der Transformation dagegen zwingend erhalten bleiben. Daraus ergeben sich

unmittelbar die Invarianten der Transformation, nämlich das Verhalten, die Integrität sowie die Korrektheit. Durch die Absicherung dieser Invarianten soll gewährleistet sein, dass Eigenschaften von Modellen im Sinne von modellierten Zusammenhängen erhalten bleiben, dass modellierte Informationen durch die Transformation nicht verfälscht und ein Zielmodell grundsätzlich konsistent und fehlerfrei ist. Sowohl die Struktur als auch die Vollständigkeit können nicht Invarianten sein, da zwischen Modellen nur bestimmte Informationen übertragen werden sollen. Gleichfalls ist damit die Vollständigkeit als Anforderung an eine Transformation im Sinne eines vollständigen Übergangs aller Elemente vom Quell- zum Zielmodell nicht Bestandteil des SMF. Dagegen muss die im SMF angewandte Transformation eindeutige Ergebnisse (determiniert) produzieren, ein formales Modell als Grundlage besitzen (wohl-geformt, das Basismetamodell übernimmt diese Aufgabe), in endlich vielen Schritten beendet sein und konsistente Ergebnisse bei der Übertragung (korrekt) liefern. Die Konzepte des SMF und die definierte Modellierungsmethode aus Kapitel 6 führen zu einer konfigurierbaren Transformation in dem Sinne, dass über die Beziehung zum Basismetamodell die Beteiligung unterschiedlicher Modelle im Transformationsprozess definiert wird. Dabei kommen bei der Definition einer Transformation über das Basismetamodell nur wenige Konstrukte zum Einsatz (vgl. 6.3.1). Das Ziel der Transformation, das multiperspektivische Servicemodell als Komposition von Modellen, wird erreicht, indem die Verknüpfungen zwischen Basismetamodell und involvierten Konzepten der verwendeten Modellierungssprachen für konkrete Modelle interpretiert und im Zielmodell umgesetzt werden.

Die Interpretation des im vorigen Abschnitt gezeigten, grafischen Modells erfolgt über dessen serialisierte Form. Bei Verwendung des Ecore Metametamodells (s. 9.3) führt die Modellierung mit Hilfe des prototypisch entwickelten Editors zu einer XML- bzw. XMI-basierten, serialisierten Form. Eine solche Form erlaubt es, die modellierten Inhalte mit einfachen Mitteln zu traversieren und die beschriebenen Informationsübergänge sequentiell abzuarbeiten. Das multiperspektivische Servicemetamodell enthält dazu alle notwendigen Elemente (Quell- und Zielmodelle, betroffene Konzepte sowie eine Übergangsdefinition zwischen den Modellelementen). Durch die Angabe des betreffenden Service werden alle ihm zugeordneten und im Servicemetamodell durch Angabe des Modelltyps bzw. der Modellierungssprache definierten Modelle eingelesen und auf Modellebene entsprechend den Verknüpfungen Inhalte aus den Quellmodellen extrahiert und dem Zielmodell hinzugefügt. Dazu werden für jedes Quellmodell die Inhalte der betroffenen Modellelemente samt Zielelement sowie Art der Beziehung zwischen Quelle und Ziel ausgelesen und in das definierte Zielelement übertragen. Der Vorgang ist abgeschlossen, sobald alle Quellelemente

abgearbeitet sind. Sind bestimmte Zielmodelle noch nicht vorhanden, deren Modelltypen aber als Ziel im Servicemetamodell definiert, so werden auf Templates basierende, minimale Zielmodelle erstellt und befüllt. In diesem Fall muss außerdem nach erfolgter Transformation das oder die Zielmodelle mit Hilfe manueller Schritt vervollständigt werden (s. auch 9.3.5).

Aufgrund der Anforderungen an das SMF ist dieser Abschluss folgerichtig, da mit dem SMF nur der Zweck der transparenten Informationsmodellierung verfolgt wird, d.h. nur solche Informationen wiederverwendet werden können, die bereits modelliert sind. Eine darüber hinaus gehende Vervollständigung, bei der Transformation noch nicht existierender Modelle, kann nicht durch das SMF abgedeckt werden. Es wird aber dadurch erreicht, dass eine manuelle Übernahme bzw. wiederholte Modellierung vermieden wird. Insbesondere bei einer größeren Zahl an Services bzw. Modellen führt dieser Ansatz so zu einem verringerten Modellierungsaufwand.

Teil III

Anwendung und Validierung

Die Anwendung der Artefakte der modellgetriebenen Servicebeschreibung wird anhand einer spezifischen Domäne dargestellt. Vor der eigentlichen Validierung in Kap. 9 wird zunächst diese Anwendungsdomäne in Kap. 8 näher beleuchtet. Aufgrund der nachfolgend erläuterten Entwicklungen in der Logistik ist diese sehr gut geeignet, das Zusammenspiel und die Wirkungsweise der Konzepte und Komponenten zu testen. Insbesondere soll der Exkurs II: Servicemodellierung in LSEM die Zusammenhänge verdeutlichen. Im Mittelpunkt steht dabei die Beschreibung dynamischer Logistikdienstleistungssysteme, da diese für die Entwicklung der LSEM-Konzepte maßgeblich sind.

8 Anwendungsdomäne Logistik

Generell ist die Logistik verantwortlich alle Abläufe, die mit der Beschaffung, Produktion und dem Vertrieb von Gütern zusammenhängen, zu unterstützen. Grundlegend beschäftigt sie sich daher mit der zeitlichen oder räumlichen Überbrückung für logistische Objekte (Material, Personen oder Informationen) und der bedarfsgerechten Verfügbarkeit dieser Objekte (Arnold, Isermann et al. 2008). Ihre Hauptaufgabe besteht im effizienten Bereitstellen der geforderten Mengen in der richtigen Art und Weise, zur rechten Zeit und am richtigen Ort (Gudehus 2010, S. 3)). Diese Beschreibung zeigt, dass die Logistik als Anwendungsdomäne für die Modellierung von Services geeignet ist. Sämtliche Aktivitäten, die zur Unterstützung eingesetzt werden, lassen sich als separate Dienstleistungen verstehen. Das Potenzial dazu wurde bereits durch (Kummer 1992, S. 23ff) erkannt. Demnach kann die Logistik je nach Definition eine Dienstleistungs-, Prozess- oder auch eine Koordinationsorientierung aufweisen, mit unterschiedlichen daraus erwachsenden Eigenschaften bzw. Zielen. Bei der Dienstleistungsorientierung steht die Handhabung von Gütern, Informationen oder Energien und deren Transformation von einem Ausgangs- in einen Endzustand im Vordergrund, bei der sich mindestens eine der Systemgrößen Zeit, Ort, oder Menge ändert. In einer prozessorientierten Sichtweise ist der Kern der Logistik auch ein Transformationsprozess und zwar der von Ressourcen in Leistungen zwischen Beschaffungs- und Absatzmarkt. Die Logistik wird hier aber als Sammlung aller Flüsse (z. B. Waren- oder Informationsflüsse) aufgefasst, die in der Prozesskette zwischen Produktion und Kunden auftreten.

Neben dieses Sichtweisen ist für diese Arbeit die Definition von (Fortmann und Kallweit 2007, S. 20) relevant. Sie definieren die Logistik unter einem Koordinationsaspekt mit einer Flussorientierung, indem sie die Logistik als die integrierte Planung, Gestaltung, Abwicklung und Kontrolle von Lagerungs- und Transportvorgängen sowie der dazugehörigen Informationsflüsse beschreiben, die innerhalb und zwischen Unternehmen,

bzw. vom Unternehmen zum Kunden sowie bei der Entsorgung auftreten. Sämtliche Materialflüsse werden um vorausseilende oder begleitende Informationsflüsse ergänzt und bilden durch die vollständige Abdeckung von der Beschaffung über die Produktion bis hin zur Entsorgung einen Logistikkreislauf, der Wirtschaftsbereiche miteinander verbindet. Diese Sicht auf die Logistik entspricht im Wesentlichen auch der Entwicklung von Logistikdienstleistern hin zu Koordinatoren und Integratoren (s. 8.3) und repräsentiert somit einen wichtigen Aspekt der Logistik für diese Arbeit. Verbindet man diese Sichtweise mit der von (Pfohl 2010, S. 255f), bei der Logistik als Dienstleistungsfunktion aufgefasst wird, so ergibt sich ein weiterer Grund für die Anwendungsdomäne Logistik: Sie dient als Marketingkanal und Absatzhelfer. Die Logistik wird dadurch zu einem Preisfaktor des zu transportierenden Gutes und kann als Service auch fremdbezogen werden (vgl. 8.1).

Sämtliche logistische Aktivitäten unterstehen dabei dem Primat der Effizienz, bei (Fortmann und Kallweit 2007) ausgedrückt durch die 6 R's der Logistik. Hauptgründe für dieses Primat sind aus Sicht der Logistikunternehmen die geringen Margen. Aus Sicht der auftraggebenden Unternehmen wird Logistik als Unterstützungsfunktion bzw. als notwendiges Übel zur Beschaffung und zum Vertrieb der Güter verstanden. Für die Logistik bedeutet dies die Ausrichtung der eigenen Aktivitäten dahingehend, dass Güter rechtzeitig zum Ziel gelangen unter entsprechenden Nebenbedingungen, die teils vom Kunden, teils durch eigene Restriktionen vorgegeben sind. Je nachdem wie feingranular in der Literatur Restriktionen definiert sind, sind unterschiedliche „R's“ üblich (z. B. (Pfohl 2010, S. 12f) 4R-Regel, (Kummer, Grün et al. 2006, S. 197) 5R-Regel, (Koether 2007, S. 11) 6R-Regel oder (Jetzke 2007, S. 11) 8R-Regel). Allen Regeln gemeinsam ist dabei die Qualität, die Quantität, der Zeitpunkt und die Kosten des Transports als entscheidende Faktoren. Vergleichbar mit Services werden über die verschiedenen „R's“ Anforderungen an die logistischen Dienstleister gestellt. Es bleibt diesen jedoch überlassen, wie sie die Anforderungen erfüllen, um auf dem Markt bestehen zu können. Dementsprechend repräsentieren die zu transportierenden Güter das „was“ und die Art der Erfüllung das „wie“ bei einer logistischen Dienstleistung (vgl. die Einführung zur Serviceorientierung in 1.1). Somit bietet die Logistik einen sehr treffenden Anwendungsfall der modellgetriebenen Servicebeschreibung. Die nächsten Abschnitte zeigen wichtige Entwicklungsschritte der Logistik, die schlussendlich in Geschäfts- und Betreibermodellen enden, für die die modellgetriebene Servicebeschreibung einen eindeutigen Mehrwert hinsichtlich der Serviceentwicklung und -erbringung stiftet.

8.1 Trends in der Logistik

Die Logistik entwickelt sich stetig weiter. Ursprünglich war sie als Unterstützung innerhalb der produzierenden Unternehmen angesiedelt. Die Logistik zählt dabei zu den typischen Querschnittsfunktionen eines Unternehmens und ist material- und warenflussbezogen. Auszuführende Aktivitäten können dabei entweder phasenspezifisch (Beschaffung, Produktion, Distribution, Entsorgung) oder funktions- bzw. verrichtungsspezifisch (Transport, Lagerhaltung, Verpackung, Auftragsabwicklung) gruppiert werden (Jetzke 2007, S. 27/28), (Arnold, Isermann et al. 2008, S. 1104f). Diese Betrachtungsweise entspricht im Wesentlichen der ersten Stufe der logistischen Entwicklungen in (Schmitt 2006, S. 15). Die weitere Entwicklung der Logistik führte von einer zunehmenden Koordinationsfunktion (Stufe 2) über eine Steuerung der gesamten Waren- und Produktflüsse (Flussorientierung) eines einzelnen Unternehmens (Stufe 3) hin zu einer unternehmensübergreifenden Steuerung der Warenflüsse in einer Lieferkette (Supply Chain Management, SCM; Stufe 4). Vorreiter dieser Entwicklung in Deutschland war die Automobilindustrie, da sie ein wesentlicher Treiber für die ständige Weiterentwicklung logistischer Funktionen und Anbieter ist (Baumann 2008). Wichtigstes Merkmal der Veränderung dabei ist, dass die Logistik selbst nicht länger Bestandteil der Kernkompetenzen von OEMs ist (Boppert und Rinza 2007), sondern aus monetären Gründen an qualifizierte Dienstleister ausgelagert wird. Der prägende Begriff hierfür ist das so genannte „Outsourcing“ ein Kunstwort, das sich aus den Begriffen „outside“, „resource“ und „using“ zusammensetzt (Bretzke 2008, S. 240), (Vahrenkamp und Siepermann 2005, S. 382ff). Dieser Trend zur Auslagerung nicht essentieller Unternehmensaktivitäten an spezialisierte Anbieter ist jedoch nicht auf die Automobilbranche beschränkt geblieben, sondern wird mittlerweile in den meisten Industriezweigen intensiv genutzt. In der Logistik übernehmen in diesem Fall ein oder mehrere spezialisierte Anbieter teilweise oder vollständig die Realisierung aller anfallenden Aufgaben für ihre Auftraggeber (Gudehus 2010, S. 987).

Die Auslagerung logistischer Dienstleistungen an spezialisierte Unternehmen verbunden mit der immer zentraler werdenden Rolle der Logistik, im Sinne einer auch unternehmensübergreifenden Steuerung der Warenflüsse, führte in der Folge zu einer Auffächerung der Spezialisierungen (s. Abbildung 8-1 und (Gudehus 2010, S. 993ff)), wobei Unternehmen im oberen Teil der Pyramide deutlich mehr und differenziertere Leistungen anbieten können, als Unternehmen darunter.

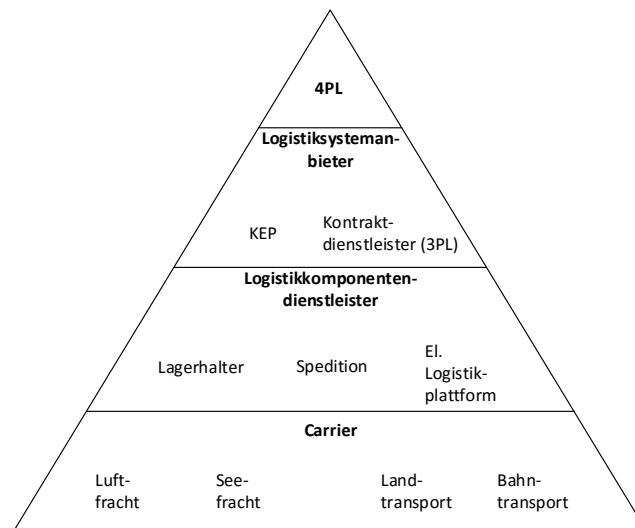


Abbildung 8-1: Marktsegmentierung in der Logistik (nach (Schmitt 2006, S. 34))

Von unten nach oben nimmt die Komplexität und Eigenständigkeit der verrichteten Aufgaben zu, ebenso die Nutzung von IT-Systemen. Die Beauftragung von Leistungen erfolgt dagegen von oben nach unten, d.h. Carrier oder Komponentendienstleister sind häufig Unterauftragnehmer der Systemanbieter oder eines 4th Party Logistics-Dienstleisters (4PL). Während Carrier lediglich spezialisierte Transportunternehmen sind, die im Auftrag eines Kunden den Warenfluss aufrechterhalten, übernehmen Komponentendienstleister wie Speditionen (s. §§453-466 HGB bzw. (Vahrenkamp und Siepermann 2005, S. 46ff)) teilweise bereits eigenständig die Kontrolle über den Warenfluss und die zu manipulierenden Güter. Systemanbieter wie der Kontraktlogistiker übernehmen zusätzlich zur Organisation der Logistik weitere Leistungen bis hin zur gesamten Logistiksteuerung für einen Kunden, teils auch über mittel- und langfristige, vertragliche Vereinbarungen.

Diese Veränderungen in der Firmenstruktur innerhalb der Logistikbranche sind teilweise auf veränderte Wahrnehmungen oder auf neue Zielstellungen zurückzuführen. Nach (Kummer, Grün et al. 2006, S. 195) entwickelt sich die Logistik weg vom notwendigen Übel produzierender Unternehmen hin zu einem strategischen Wettbewerbsvorteil und zu einem zentralen Unternehmensprozess, mit dessen Hilfe das gesamte Unternehmen gesteuert werden kann. Allerdings ist die Logistik ein wichtiger Kostenfaktor (Kummer, Grün et al. 2006, S. 197), so dass Unternehmen notwendigerweise die Effektivität und Effizienz logistischer Aktivitäten optimieren müssen. Konsequenterweise ändern sich auch die zugrundeliegenden Zielgrößen bei der Betrachtung logistischer Aktivitäten (Pfohl 2010, S. 46ff). Bedingt durch enge Margen und die zunehmende Globalisierung entwickelt sich in der Logistik ein Zwang hin zur Ausnutzung verschiedenster Größenvorteile, etwa

Economies of Arbitrage (Preisvorteile), Economies of Scale (Größenvorteile), Economies of Scope (Bündelungsvorteile), Economies of Speed (Geschwindigkeitsvorteile) oder Economies of Structure (Änderungsvorteile). Auch führen Produkt- und Prozessinnovationen³ der Vergangenheit zu einem zunehmenden Marktdruck durch die Kunden, etwa weil Erfahrungen und Innovationen zwischen bislang isoliert betrachteten Teilmärkten transferiert werden. Dies treibt die Senkung der Margen weiter voran, so dass (logistische) Unternehmen mit sinkender Ertragskraft umgehen müssen. Daher löst die Rentabilität das Umsatzdenken als wichtige Zielgröße ab. Insgesamt steigt auch bedingt durch neue IuK die Bedeutung des Produktionsfaktors Information. Damit zu erreichende Vorteile sind eine Verkürzung der Reaktionszeitung durch Ausbau von Planungs- und Kontrollsystemen sowie eine Flexibilisierung des Unternehmens. Als Konsequenz müssen sich Unternehmen neu orientieren, um im intensiven Wettbewerb bestehen zu können. Wer bestehen will muss ganzheitlich optimieren: kurze Lieferzeit, Termintreue, hohe Qualität, guter Service und passender Preis (Göpfert 2009, S. 52).

Als Antwort darauf setzt sich das 4PL-Konzept des Systemintegrators in der Logistik mehr und mehr durch (s. auch (Vahrenkamp und Siepermann 2005, S. 49) und Exkurs II bzw. 8.3). Im Gegensatz zum unternehmenszentrierten Kontraktlogistiker, weiten 4PL die Organisation und Steuerung der Logistik auf Supply Chains aus und bieten neben eigenen, vorwiegend Wissens- und technologiebasierten Ressourcen, Fremdleistungen der tieferliegenden Marktsegmente (s. Abbildung 8-1) als Komplettangebot an. Ein 4PL übernimmt dazu die Aufgabe der Entwicklung, Gestaltung, Lenkung und Realisierung effektiver und effizienter Flüsse von Objekten (Güter-, Informations-, Geld- und Finanzflüsse) in unternehmensweiten und unternehmensübergreifenden Wertschöpfungssystemen, was der Logistik als moderne Führungskonzeption entspricht (Göpfert 2009, S. 58). Die Integration von Fremdleistungen in ein Gesamtangebot durch einen 4PL zwingt diesen jedoch auch zu einer systematischen Entwicklung und Beschreibung der angebotenen (Teil-)Leistungen. Somit sind 4PL-Logistikdienstleister potenzielle Nutznießer der modellgetriebenen Servicebeschreibung.

8.2 Logistiksysteme und Supply Chain Management

Die aufkommende Steigerung der Prozess- und Dienstleistungsorientierung und die stärkere koordinative Rolle der Logistik wirken sich insbesondere auf die unternehmensübergreifende Gestaltung der Logistik aus. Während innerbetriebliche Logistiksysteme den Fokus

³ Beispiele hierfür sind etwa die Besetzung von Nischensegmenten durch logistische Dienstleister oder Überwachung der Lieferung (Tracking & Tracing) sowohl durch Versender wie Empfänger der Güter.

auf die Ver- und Entsorgung der Maschinen- und Handarbeitsplätze setzen (Arnold, Isermann et al. 2008, S. 18/19), bilden unternehmensübergreifende Logistiksysteme den Warenfluss zwischen Betriebsstätten eines Unternehmens oder die gesamte Lieferkette zwischen Unternehmen ab. (Gudehus 2010, S. 5, 550) definiert ein Logistiksystem als Leistungssystem, welches mit Hilfe „... wechselnder Strategien an physischen oder informatorischen Objekten materielle und imaterielle Transformationen ausführt“ (s. dazu auch (Pfohl 2010, S. 4)). Beispiele für Transformationen sind solche der Zeit (Lagerung) oder des Ortes (Transport). Die Entwicklung bzw. Realisierung von Logistiksystemen setzt dabei spezifizierte Aufgaben, Ziele sowie Leistungsergebnisse und die Definition von Schnittstellen und Rahmenbedingungen voraus. Aufbau, Netzwerk, Funktionen, Kapazitäten und Leistungsvermögen sind aus planerischer Sicht zu analysieren, wobei zur Verringerung der Komplexität diese unter einem Struktur- wie unter einem Prozessaspekt zu beleuchten sind (Gudehus 2010, S. 6-8). Die Zahl der Sichten kann jedoch auch weiter erhöht und um eine Datensicht, Produktsicht, Organisationsicht erweitert werden (Jetzke 2007, S. 25). Im Zusammenhang mit dem Trend zur Entwicklung von Systemintegratoren, die solche Logistiksysteme für ihre Kunden entwickeln, ergibt sich somit die Möglichkeit die modellgetriebene Servicebeschreibung auch auf die Planung eines Logistiksystems anzuwenden und die notwendigen Sichten in einem Gesamtbild zu vereinen.

Im Folgenden wird der Fokus auf mehrstufige Logistiksysteme mit einer netzwerkartigen Struktur gelegt. Solche Netzwerke logistischer Unternehmen sind insbesondere geeignet auch komplexe Lieferketten und damit unternehmensübergreifende Güterflüsse abzubilden (Vahrenkamp und Siepermann 2005, S. 9,13). Sie werden auch als großflächige Logistiksysteme bezeichnet und umfassen „...Lieferstellen, Produktionsstellen, Logistikstationen und Empfangsstellen, die durch Transportverbindungen miteinander verknüpft sind und von Material- und Warenströmen durchflossen werden.“ (Gudehus 2010, S. 16) (s. auch (Arnold, Isermann et al. 2008, S. 934)). Diese Logistiksysteme oder Lieferketten werden in der Regel über Kooperationen zwischen Unternehmen realisiert, wobei diese sowohl horizontal, vertikal, als auch diagonal entlang der gesamten Wertschöpfungskette erfolgen können ((Vahrenkamp und Siepermann 2005, S. 16/17), (Pfohl 2010, S. 289)). Als Treiber für Kooperationen und essentiell für die Umsetzung und operative Durchführung haben sich dabei die Intensivierung des Informationsaustauschs (vorausgehend oder begleitend) (Vahrenkamp und Siepermann 2005, S. 14) bzw. die intensive Nutzung von Informationssystemen wie z. B. Planungs- und Steuerungssysteme für Supply Chains herauskristallisiert (Arnold, Isermann et al. 2008, S. 592f). Dies wird auch durch die zentrale Rolle der IuK in einigen Netzwerkvarianten deutlich. Ein wichtiger

Erfolgsfaktor einer solchen Kooperation stellt die Integration von Leistungen dar, einerseits in Form von Leistungsbeschreibungen und andererseits in Form von zueinanderpassenden Teilleistungen mit entsprechenden Übergabepunkten und Schnittstellen, bei deren Beschreibung die modellgetriebene Servicebeschreibung ebenfalls Anwendung findet.

Eine strategisch immer bedeutsamere und zugleich komplexe Variante des mehrstufigen Logistiksystems ist die überbetriebliche Lieferkette oder auch Supply Chain (zur synonymen Verwendung der Begriffe s. (Göpfert 2009, S. 61)). Die netzwerkartige Struktur dieser „Kette“ zeigt sich in den teils komplizierten Zulieferverflechtungen, in denen unterschiedliche Kategorien von produzierenden Unternehmen und logistischen Dienstleistern interagieren, um die vorgelagerte Versorgung oder den nachgelagerten Vertrieb eines koordinierenden Unternehmens (z. B. ein OEM, oder Handelshaus) zu gewährleisten. Logistische Systemdienstleister (s. 8.1) versorgen bspw. fokale Unternehmen mit Halbfertigerzeugnissen oder Komponenten für die Endmontage oder übernehmen die kundengerechte Steuerung der Zulieferung für das Unternehmen (s. (Pfohl 2010, S. 295ff)).

Für die operative Unterstützung, Standardisierung und Realisierung von Supply Chains hat sich das Supply Chain Operations Reference-Modell⁴ (SCOR-Modell) als de-facto Standard etabliert. Im Kern besteht SCOR aus vier Ebenen wobei die oberste Ebene zunächst die fünf Prozesskategorien „Plan“, „Source“, „Make“, „Deliver“ und „Return“ definiert und die darunterliegenden Ebenen für jede Kategorie Prozesse (Ebene 2) bzw. Teilprozesse (Ebene 3) spezifiziert. Ebene 4 umfasst als logische Folge die Implementierung der SCOR-Prozesse, ist jedoch nicht Bestandteil des Modells, da die Implementierung unternehmensspezifisch erfolgen muss. Von besonderer Bedeutung ist auch die informationstechnische Unterstützung, gerade bei der Leistungsbeschreibung, um innerbetrieblich wie überbetrieblich Informationen über Prozesse und Aktivitäten entlang der Kette austauschen zu können. Zwar sind speziell entwickelte SCM-Systeme in der Lage Planungs- Optimierungs- und Steuerungsaufgaben entlang der gesamten Wertschöpfungskette zu realisieren, sie arbeiten jedoch Kennzahl-basiert und nur aus Sicht des fokalen Unternehmens, für das die Kette realisiert wird. Tiefergehende Leistungsbeschreibungen sind entweder gar nicht Bestandteil solcher Systeme oder können nicht für einen Informationsaustausch genutzt werden, so dass eine konsistente Gesamtsicht häufig nicht existiert, jedoch mit Hilfe der modellgetriebenen

⁴ <https://supply-chain.org/scor>

Servicebeschreibung sukzessive entwickelt werden kann. Dies trifft umso mehr zu, als dass die Planung einer solchen komplexen, logistischen Leistung in der Regel nicht mehr durch das (fokale) Unternehmen selbst durchgeführt. Spezialisierte Anbieter, wie der 4PL, übernehmen die Planung, Steuerung und Überwachung sämtlicher Flüsse zwischen den beteiligten Unternehmen. Gerade die Planung einer Supply Chain ist derzeit aber eine sehr zeitintensive Phase (s. (Gudehus 2010, S. 69ff)) Planungs- bzw. Anbahnungsphasen über mehrere Monate hinweg sind üblich. Insbesondere die Phase der Detailplanung der Systemaufbau sind mit 3-6 bzw. mit 12-18 Monaten sehr lang. Hier versucht die Arbeit als Bestandteil von LSEM einzuwirken, und über standardisierte und modularisierte Bausteine diese Zeit zu verkürzen.

8.3 Betreiber- und Geschäftsmodelle für Logistiksysteme

Die Basis für die Anwendung der in dieser Arbeit vorgestellten Konzepte bilden die Entwicklungen in der Logistik. Wie bereits in den vorangegangenen Abschnitten verdeutlicht kristallisiert sich ein Wandel in Bezug zur Durchführung logistischer Leistungen heraus. Auf der einen Seite wird die Logistik nicht mehr länger als notwendiges Übel begriffen. Unternehmen steuern vermehrt ihre Produktions- oder Handelsprozesse mit Hilfe der Logistik (s. Supply Chains). Daraus folgt, dass Kompetenzen in diesem Bereich verstärkt notwendig werden. Auf der anderen Seite liegen die Kernkompetenzen der Unternehmen nach wie vor im Bereich der Produktion oder im Handel mit Produkten. Die zunächst aufgebauten Kompetenzen im Bereich der Logistik erforderten jedoch zunehmend mehr Ressourcen, die im Zuge von Unternehmensumstrukturierungen wieder ausgelagert werden (Outsourcing) (Fortmann und Kallweit 2007, S. 152). Speziell solche Randkompetenzen wie die Logistik werden an dafür spezialisierte Unternehmen ausgelagert, mit dem Ziel durch eine Optimierung der Ressourcenallokation auf beiden Seiten Vorteile zu ermöglichen. Insbesondere sollen auf beiden Seiten Wettbewerbsvorteile erreicht werden, indem beim auslagernden Unternehmen Ressourcen frei, nicht notwendige Prozessschritte vermieden und so Kosten gesenkt werden und beim durchführenden Unternehmen bspw. Skaleneffekte erzeugt werden. Von Vorteil für das auslagernde Unternehmen sind gerade die Kosteneffekte, allen voran die Variabilisierung der Kosten, also die Verringerung der Fixkostenanteile durch dynamischen Zu- oder Verkauf von Leistungen. Die Erörterung der Materie des Outsourcings soll jedoch hier nicht weiter vorangetrieben werden. Die Outsourcing-Bestrebungen von Unternehmen führten jedoch zu einer Weiterentwicklung logistischer Dienstleister.

Eine frühe und auch rechtlich fixierte Form der Spezialisierung ist das Speditionsgeschäft. Es umfasst lediglich die Organisation von Transporten, nicht aber deren Ausführung. Dementsprechend verfügen Spediteure über Spezialwissen im verkehrsträgerübergreifenden Transportbereich und sind wichtige Mittler in der Konstituierung von Logistikketten. Sie sind in der Systematik in Abbildung 8-2 als Spezialanbieter von logistischen Einzelleistungen bzw. in der Kategorisierung von (Fortmann und Kallweit 2007, S. 153) für Logistikdienstleister als Funktionsspezialisten einzuordnen. Die Kategorisierung unterscheidet darüber hinaus Generalisten, Funktionsspezialisten, Branchenspezialisten und Nischenspezialisten.

Die Art der Unterscheidung oder Einteilung logistischer Dienstleister wird bei (Arnold, Isermann et al. 2008, S. 979) oder bei (Gudehus 2010, S. 993ff) im Wesentlichen durch das Leistungsangebot bestimmt. Hauptmerkmal der Unterscheidung ist dabei zunächst die Spezifik der Leistung, also ob eine Einzelleistung wie z. B. Transport oder Umschlag oder eine Kombination aus Leistungen (Umschlag und Lager) angeboten wird. Darüber hinaus wird der Grad der Komplexität der Leistung bzw. die Kundenorientierung als Unterscheidungsmerkmal herangezogen und so zwischen vier unterschiedliche Typen von Dienstleistern in der Logistik unterschieden (s. Abbildung 8-2).

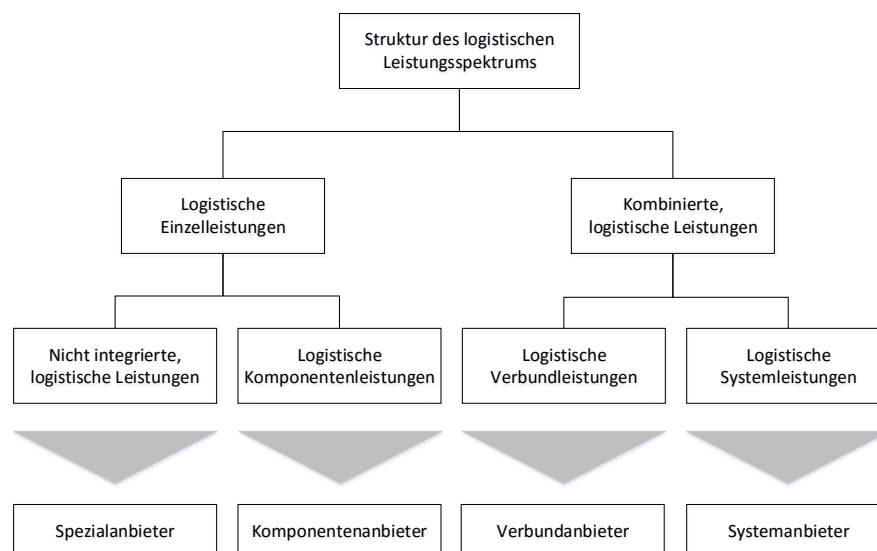


Abbildung 8-2: Typisierung von Logistikdienstleistern (nach (Arnold, Isermann et al. 2008, S. 979))

Spezialanbieter und Komponentenanbieter besitzen ein sehr spezifisches Leistungsangebot. In Abbildung 8-1 sind dies Beispiele aus den unteren Marktsegmenten. Verbund- und Systemanbieter sind dagegen in den beiden oberen Segmenten zu finden. Ihr Leistungsangebot ist entweder sehr breit gefächert oder sie bieten ein sehr kundenspezifisches Produkt, etwa die Herstellung und die Versorgung mit Halbfertigerzeugnissen oder die Organisation ei-

nes Teils oder der gesamten Logistik für einen Kunden. Dabei treten Spezial- und Komponentenanbieter häufig als Subdienstleister auf.

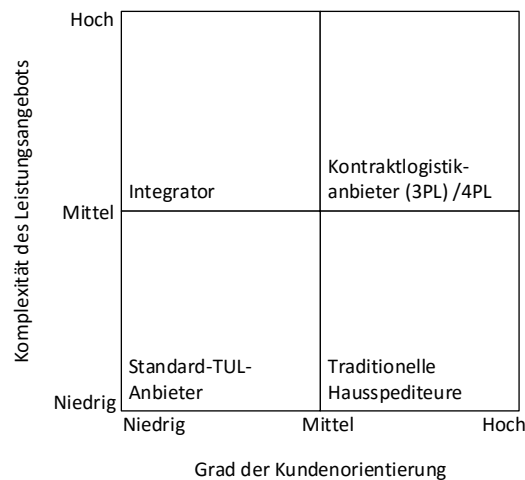


Abbildung 8-3: Spannungsfeld von Leistungskomplexität und Kundenorientierung (nach (Arnold, Isermann et al. 2008, S. 979))

Die Zusammenhänge zwischen Umfang des Leistungsangebots, dem Grad der Kundenorientierung und des korrespondierenden Dienstleistertyps verdeutlicht Abbildung 8-3. Ausgehend vom linken, unteren Quadranten mit standardisierten Einzelleistungen steigt zunächst in der Dienstleisterentwicklung der Grad der Kundenorientierung, so dass sich das Speditionsgeschäft als Möglichkeit für neue Leistungen etablierte. Getrieben durch das steigende Ausmaß an ausgelagerten, logistischen Funktionen und den steigenden Bedarf an externem Wissen über das logistische Management haben sich zunächst Integrator-Konzepte entwickelt, die mehrere standardisierte Leistungen als Gesamtangebot zur Verfügung stellen.

Die Übernahme der gesamten Logistik als Komplettlösung, in der insbesondere auch die Organisation, Steuerung und Überwachung zumindest eines Teils eines Logistiksystems inbegriffen ist, stellt die jüngste Entwicklung dar. Da hier Bündel standardisierter Leistungen nicht mehr ausreichen, sondern individuelle, hochflexible Lösungen gefordert sind, steigt schlussendlich der Grad der Kundenorientierung. Der Fokus liegt hierbei vor allem im Management-Bereich. Eigene physische Ressourcen werden zugunsten des Wissens eher verdrängt und über Subdienstleister eingekauft. Als Name hat sich für solche Logistik-, „Generalisten“ zunächst der Kontraktlogistiker (oder auch 3PL) eingebürgert ((Arnold, Isermann et al. 2008, S. 979), (Vahrenkamp und Siepermann 2005, S. 46ff)). Mit dem Namen wird dabei der typischerweise langjährigen, vertraglichen Zusammenarbeit Rechnung getragen, die sich aus der kompletten Übernahme der Logistik ergibt. Als Weiterentwicklung im Zuge der Übernahme des gesamten Logistiksystems bzw. der Steuerung einer ge-

samten Supply Chain wurde schließlich das 4PL-Konzept entwickelt. Dieses bündelt sämtliche logistische Schnittstellen zwischen sämtlichen Zulieferern und logistischen (Sub-) Dienstleistern und sorgt für eine lückenlose Steuerung und Überwachung sowie für ein reibungsloses Zusammenspiel aller Beteiligten im Auftrag eines Kunden. Das Konzept ist zwar in der Logistikforschung aufgrund der Umsetzungschancen umstritten (Arnold, Isermann et al. 2008, S. 979), in der Realität mehreren sich aber Unternehmen, die versuchen dieses Konzept umzusetzen.

Dazu passend zeigen regelmäßig durchgeführte Umfragen (Teleroute 2005; von der Gracht, Däneke et al. 2008; Langley 2009; Terry 2009; Terry 2010; Terry 2012) einen stetig wachsenden Markt im Bereich der System- und so genannten Mehrwertleistungen. Die Umfragen bestätigen sowohl den Trend zur Auslagerung als auch die Inanspruchnahme logistischer Funktionen. Die Auslagerung umfasst dabei nicht mehr die reine Durchführung des Warenflusses bzw. der Lagerung, sondern zunehmend Planungs- und Koordinationsaufgaben der Supply Chain (Blanchard 2008; Bamford 2010). Diese neuen Marktteilnehmer werden je nach Grad der Integration von Managementfunktionen in 3PL, 4PL oder LLP (Lead-logistics provider) unterschieden (Nissen und Bothe 2002; Kleine-Kleffmann und Bößer 2006; Schmitt 2006; Kutlu 2007; Lischke 2008). Im Zuge der Entwicklung neuartiger Managementkonzepte für die Logistik sind daher auch Veränderungen in den Unternehmensstrukturen zu erkennen. Einerseits bilden oder entwickeln sich bestehende Unternehmen weiter und vergrößern ihr Leistungsangebot (z. B. Deutsche Post, DB Schenker). Andererseits ist aber auch zu erkennen, dass sich netzwerkartige Strukturen bilden, in denen Logistikunternehmen gemeinschaftlich miteinander interagieren und gegenseitig unterstützen. Dies tritt vor allem bei kleineren und mittleren Logistikunternehmen auf, die ihre Größennachteile durch Kooperationen auszugleichen versuchen (Arnold, Isermann et al. 2008, S. 981).

Die Herausforderung bei der Anwendung des 4PL-Managementkonzepts besteht unabhängig von der konkreten Ausprägung jedoch darin, kundenindividuelle Leistungen anbieten zu können, und gleichzeitig die Vorteile des Outsourcings wie Skaleneffekte nutzen zu können. Dies führt zwangsläufig zur Einführung standardisierter Komponenten, die einfach in der Nutzung und im Austausch sind. Allerdings bestehen die Komponenten, im Vergleich zu Sachgütern etwa, nicht aus physischen Bestandteilen, sondern aus Lagerkapazitäten, Transportmöglichkeiten, Veredlungsleistungen wie das (Um-)Verpacken von Gütern, das Kommissionieren oder die Zollabfertigung, also aus Dienstleistungen. Für eine Umsetzung des 4PL-Konzepts in einem Unternehmensnetzwerk müssen dann Servicebeschreibungen aller Beteiligten beim 4PL vorliegen, damit dieser nach Kundenwunsch ge-

eignete Services auswählen und zu einer Gesamtleistung kombinieren kann. Eine Integration dieser einzelnen Servicebeschreibungen zu einem vollständigen Gesamtbild liefert die modellgetriebene Servicebeschreibung und kann so bei der Umsetzung des 4PL-Konzepts unterstützen.

9 Validierung und Evaluation

Die Validierung als Einordnung, Bewertung und praktische Übertragbarkeit bzw. Anwendbarkeit des hier vorgestellten Ansatzes schließt zugleich dessen Erarbeitung ab. Die Validierung soll zunächst zeigen, dass die entwickelten Konzepte, Methoden und Techniken wissenschaftlich stichhaltig (rigour, 9.1) sind und andererseits, dass sie innerhalb einer Domäne auch anwendbar sowie praktisch relevant sind (9.2 und 9.3). Dazu wird eine in (Becker, Krcmar et al. 2009, S. 73) als gültig angesehene Fallstudienbeschreibung verwendet und ein Überblick über die einzelnen Komponenten des Prototyps gegeben, sowie anhand eines Beispiels (9.3.6) gezeigt, wie die Komponenten zur Lösung eingesetzt werden. Die darauffolgende Evaluation (9.4) bewertet abschließend den hier vorgestellten Ansatz anhand des Vergleichs mit den verwandten Ansätzen.

9.1 Wissenschaftliche Validierung

Die Stichhaltigkeit der Konzepte, Methoden und Techniken des SMF wurden nach Erreichen eines definierten Stabilitätsniveaus in verschiedenen nationalen sowie internationalen Konferenzen publiziert sowie verteidigt. Nachfolgend sind ausschließlich die veröffentlichten Publikationen entsprechend ihrer thematischen Zugehörigkeit aufgelistet:

Konzept und Komponenten:

- (Augenstein, Ludwig et al. 2012) Integration of service models - preliminary results for consistent logistics service management. In: SRII Global Conference 2012, San Jose, 2012.
- (Augenstein und Ludwig 2013a) Interconnected service models - Emergence of a comprehensive logistics service model. In: Fourth international Workshop on models and model-driven methods for service engineering (3M4SE 2013), Vancouver, 2013.
- (Augenstein und Ludwig 2013b) The Service Meta Modeling Editor - Bottom-up integration of service models. In: DESRIST 2013, Helsinki, 2013.

Anwendungsszenarien und Anwendung:

- (Kunkel, Kerkhoff et al. 2013) Generierte Mensch-Maschine-Schnittstellen zur ad-hoc Integration von Logistikketten. In: Reich, S. (Hrsg.). HMD Praxis der Wirtschaftsinformatik. S.76-86. Heidelberg, 2013.
- (Augenstein, Mutke et al. 2013) Integration von Planungssystemen in der Logistik - Ansatz und Anwendung. In: WI 2013 - 11. Internationale Tagung Wirtschaftsinformatik, Leipzig, 2013.
- (Mutke, Augenstein et al. 2013) Model-based integrated planning for logistics service contracts. In: Fourth international Workshop on models and model-driven methods for service engineering (3M4SE 2013), Vancouver, 2013.
- (Glöckner, Augenstein et al. 2014) Metamodel of a Logistics Service Map. In: Abramowicz, W., Kokkinaki, A. (Hrsg.). 17th International Conference on Business Information Systems. S.185-196. Berlin; New York, 2014.
- Eine erweiterte Version des nachträglich beschriebenen Anwendungsszenarios wurde veröffentlicht im Journal of Object Technology: (Mutke, Augenstein et al. 2015) Real-time information acquisition in a model-based integrated planning environment for logistics contracts. In: Journal of Object Technology (14), S.2:1-25, 2015.

9.2 Servicemodellierung für Logistiksysteme

Die praktische Relevanz des Ansatzes soll mit einem Beispiel in der Logistikdomäne verdeutlicht werden. Kapitel 8 beschreibt diese mit einem besonderen Fokus auf Logistikdienstleistungssystemen, einem mit dem Supply Chain Management verwandten Ansatz, bei dem jedoch die zentrale Steuerung durch einen Logistikdienstleister anstatt durch ein fokales, produzierendes oder Handel treibendes Unternehmen stattfindet.

9.2.1 Grundlagen

Dienstleistungssysteme werden dadurch charakterisiert, dass Services nicht mehr isoliert durch einzelne Unternehmen, sondern durch ein Netzwerk von Unternehmen erbracht werden. Gründe für die Entwicklung von Dienstleistungssystemen sind bspw. die Konzentration von Kompetenzen und Kapazitäten innerhalb von Unternehmen, so dass diese nicht mehr in der Lage sind, nachrangige Aktivitäten oder Ressourcen aus eigener Kraft bereitzustellen (Zahn und Stanik 2006, S. 303). Stattdessen entstehen Unternehmensnetzwerke, die komplementär oder konkurrierend, horizontal oder vertikal verbunden Leistungen im System bereitstellen (Zahn und Stanik 2006, S. 304). Im Falle der Logistik kommt erschwerend hinzu, dass viele Logistikdienstleister, die sich gemäß Abbildung 8-2 in den

unteren Ebenen einordnen lassen, eine unzureichende Ressourcenausstattung besitzen, um wirtschaftlich bedeutsame Aufträge für sich allein bewältigen zu können. Darüber hinaus entstehen in der Netzwerkgestaltung Herausforderungen bei der Errichtung und im Ablauf wie z. B. die Partnerauswahl oder die Gesamtkonzeption des Netzwerks. Bei der Partnerauswahl besteht bspw. die Aufgabe, dass für eine gegebene Leistung die Partner ausgewählt werden, die die geforderte Qualität möglichst effizient erbringen können. Die Erstellung der Gesamtleistung wiederum setzt die Verbindung der einzelnen Bausteine sowohl auf konzeptioneller Ebene als auch das Zusammenspiel der Teilleistungen bei der konkreten Serviceerbringung voraus. Dies ist gerade in der Logistik ein dringliches Problem, wenn unternehmensübergreifende Versorgungsketten durch Netzwerke erbracht werden (vgl. (Arnold, Isermann et al. 2008, S. 932), (Vahrenkamp und Siepermann 2005, S. 14)). Erschwerend kommt hinzu, dass aufgrund natürlicher Schwankungen der Nachfrage die Ressourcenbelastung innerhalb solcher Netzwerke stark schwankend sein kann und daher einer dynamischen Anpassung bedarf (vgl. (Gudehus 2010, S. 558f)). Ein Gesamtansatz zur Etablierung eines (Logistik-) Dienstleisternetzwerks kann mit Hilfe einer Plattformstrategie die Leistungserstellung im Netzwerk erleichtern. Wichtige Konzepte sind dazu die Modularisierung und Standardisierung von Komponenten, die im Netzwerk auf einer Plattform zur Verfügung gestellt werden können. Hinsichtlich der Konzeption eines Logistiknetzwerks mit ausschließlich mittelständischen Partnern haben sich aus technischer Sicht die folgenden Probleme als schwerwiegend erwiesen:

- Ein IT-technisches Verständnis ist insbesondere bei kleineren Unternehmen kaum oder gar nicht vorhanden.
- Eine Formalisierung mit Hilfe von spezifischen Modellierungssprachen ist nur schwer möglich, da weder Werkzeuge noch Kenntnisse zu deren Einsatz vorhanden sind.
- Viele Unternehmen scheuen neue Ansätze, weil die bestehende Herangehensweise etabliert ist und gelebt wird.

Als Folge davon werden neue Lösungen nur schwer angenommen. Mit dem LSEM Ansatz (s. Exkurs II) wird versucht, die Hemmnisse zu umgehen und einen Ansatz für mittelständische Logistikdienstleister zu etablieren, mit dessen Hilfe die Etablierung eines Netzwerks und einer kollaborativen Leistungserbringung realisiert werden kann. Ein Plattformansatz integriert eine Reihe von Werkzeugen, mit denen logistische Services erfasst und Aspektweise beschrieben werden können. Der hier beschriebene Ansatz der modellgetriebenen Servicebeschreibung hat im Rahmen des LSEM dann die Aufgabe, die verschiedenen Sichten zu integrieren, um für den Plattformbetreiber ein vollständiges Bild auf die

verfügbaren Services zu generieren und so die Auswahl von Partnern oder die Entwicklung einer kundenspezifischen Dienstleistung zu ermöglichen. Insbesondere modellgetriebene Ansätze eignen sich hier, um die Komplexität der IT von der der Logistik trennen, da Modelle das System auf einer höheren Abstraktionsstufe beschreiben und sich die Komplexität somit auf das Modell selbst und auf die Abbildung des Modells auf die Zielsprache aufteilen lässt (Kunkel, Kerkhoff et al. 2013, S. 78).

In der Logistik sind Modelle weit verbreitet im Einsatz. Man unterscheidet bspw. Beschreibungs-, Erklärungs-, Prognose- und Optimierungsmodelle (Pfohl 2004, S. 289ff), mittels derer Logistiksysteme beschrieben, Zusammenhänge erklärt oder auch Zielsysteme aufgestellt und optimiert werden können. Zur Entwicklung einer durchgehenden Lieferkette werden dazu regelmäßig unterschiedliche Modelle für unterschiedliche Sichten eingesetzt, die vor allem Prozess- und ressourcenorientierte Aspekte beschreiben (Arnold, Isermann et al. 2008, S. 934). Insbesondere werden Dienstleisternetzwerke auf unterschiedlichen Ebenen (institutionelle, informatorische, prozessuale und ressourcenorientierte Ebene) getrennt analysiert und dargestellt (Arnold, Isermann et al. 2008, S. 937), wobei die unterschiedlichen Ebenen zwar für sich genommen betrachtet werden, aber wechselseitige Abhängigkeiten zwischen den Sichten existieren, die über gesteigerten IT-Einsatz konsistent gehalten werden müssen. Verschiedene Werkzeuge und daraus resultierende Modelle im LSEM-Ansatz strukturieren die Informationen über das Dienstleisternetzwerk in vergleichbarer Weise. Speziell für das in 9.2.2 beschriebene Fallbeispiel einer Planung einer Logistikkette wird dies deutlich. Spezifische Modelle beinhalten institutionelle (Dienstleister) oder prozessuale (Prozess- und Simulation) Informationen. Die modellgetriebene Servicebeschreibung erfüllt in diesem Umfeld den Zweck einer Integration von Informationen aus unterschiedlichen Modellen und ermöglicht eine schnellere Modellierung des Netzwerks sowie der angebotenen Dienstleistung. Das übergeordnete Ziel ist dabei die drastische Verkürzung des Planungszeitraumes, der insbesondere für Logistiksysteme einen erheblichen Zeitraum einnimmt (Gudehus 2010, S. 69ff).

Exkurs II: Servicemodellierung in LSEM

Das Logistik Service Engineering & Management (LSEM) ist Forschungsprojekt und inhaltlicher Schwerpunkt einer Stiftungsprofessur an der Universität Leipzig. Im Kern beschäftigt sich das LSEM mit der kollaborativen und integrativen Entwicklung sowie Wiederverwendung logistischer Dienstleistungen. Der Entwicklungsansatz basiert auf dem Gestaltungsparadigma der Service-Orientierung und modelliert sowohl fachliche, logisti-

sche Funktionen (Transport, Umschlag, Lagerung, Mehrwertleistungen) als auch unterstützende Informationssysteme einheitlich als Services.

Ausgehend von einem Integrator-Ansatz (4PL), werden logistische Services eines angeschlossenen Netzwerks von Unternehmen zentral verwaltet und je nach Kundenbedarf zu einer Gesamtleistung nach dem Baukastenprinzip kombiniert. Dem Integrator fallen dabei die Aufgaben zu, geeignete Dienstleister in sein Netzwerk zu integrieren, deren Services zu katalogisieren, Kunden für das Netzwerk zu akquirieren sowie deren Aufträge bedarfsgerecht und optimal umzusetzen und dabei Synergiepotenziale aufzudecken und im Netzwerk zu verteilen. Der wesentliche Nutzen des LSEM ergibt sich insbesondere für kleine und mittlere Unternehmen, da diese aufgrund mangelnder Kapazitäten oder Ressourcen in der Regel keine größeren Aufträge für sich gewinnen können, in der Zusammenarbeit mit weiteren Dienstleistern jedoch dazu in die Lage versetzt werden.

Definition 9-1: LSEM

Das gestaltungsorientierte Forschungsziel von LSEM ist die Entwicklung integrierter Artefakte (Konstrukte, Modelle, Methoden, Instanzen) zum Engineering und Management von arbeitsteiligen, unternehmensübergreifenden Mehrwertlogistikdienstleistungen über deren gesamten Lebenszyklus.

LSEM setzt dabei auf eine werkzeuggestützte Planung, Steuerung und Kontrolle logistischer Dienstleistungen, die über einen mehrphasigen Lebenszyklus miteinander in Beziehung stehen (s. Abbildung 9-2). Die Einführung eines Lebenszyklus sorgt in diesem Kontext für eine Strukturierung der Dienstleistungserbringung (Gronau, Bahrs et al. 2010, S. 5). Grundsätzlich beschreiben Lebenszyklen dabei die Zeitspanne von der (systematischen) Entwicklung bis zur (kontrollierten) Beendigung (s. Abbildung 9-1). Die Abschnitte des abgebildeten, generischen Lebenszyklus werden dabei als Phasen bezeichnet und beinhalten spezifische Arbeitsvorgänge und Artefakte.

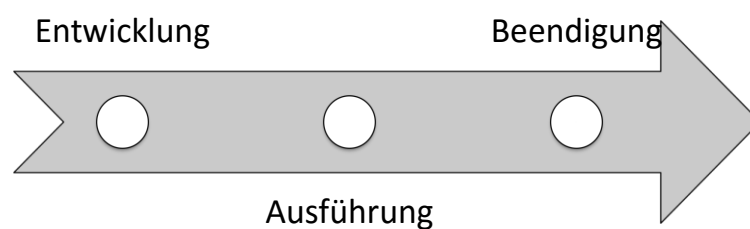


Abbildung 9-1: Lebenszyklus eines Service

Eine konkrete Ausprägung eines Lebenszyklus stellt das drei-phasige Modell von (Fährnich und Opitz 2006, S. 95) dar, wobei die dort genannten Phasen Leistungsergebnis,

Leistungserbringung und Leistungsbereitschaft alle der Entwicklungsphase im generischen Lebenszyklus untergeordnet werden können. Die zugehörigen Artefakte Produkt-, Prozessmodell und Ressourcenkonzept spiegeln die Notwendigkeit einer systematischen Serviceentwicklung zur effizienten Ressourcennutzung wieder. Mittels der Artefakte wird ein Service definiert, der Erbringungsablauf gestaltet und notwendige Ressourcen vorbereitet bzw. bereitgestellt. Der in LSEM angewandte Lebenszyklus (s. Abbildung 9-2) umfasst im Gegensatz zu letztgenanntem alle Phasen bis hin zur Beendigung. Die Werkzeuge jeder Phase unterstützen dazu ein oder mehrere spezifische Modelltypen (beispielhaft in der Abb. genannt), mit deren Hilfe ein konkreter Aspekt eines Service modelliert werden kann. Ein logistischer Service ist somit insgesamt durch eine heterogene Beschreibung mittels verschiedener Modelle gekennzeichnet, die sich bzgl. des Inhalts jedoch auch überschneiden können, etwa im Falle unterschiedlicher Beschreibungen innerhalb einer Phase oder in einer sukzessiven Verfeinerung über mehrere Phasen hinweg. Dadurch ergibt sich, dass Modellinhalte in mehreren Werkzeugen wiederverwendet werden können und damit die Problemstellung, dass Informationen zwischen verschiedenen Modellen zum Zwecke einer vereinfachten und fehlervermeidenden, ganzheitlichen Servicemodellierung ausgetauscht werden müssen.

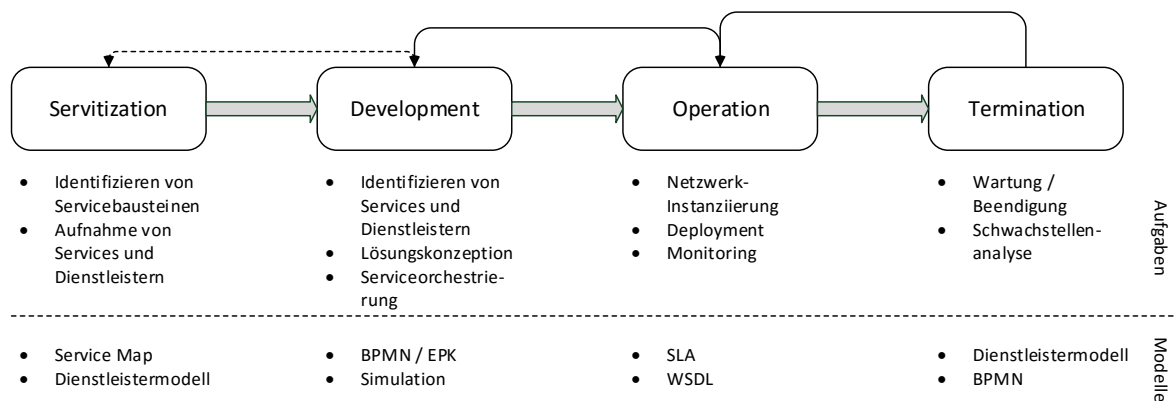


Abbildung 9-2: LSEM-Lebenszyklus

Der abgebildete, vier-phasige Lebenszyklus unterstützt eine konsistente und robuste Dienstleistungsentwicklung, eine nachhaltige Ausführung sowie eine geplante und geordnete Beendigung der modellierten Services. Treten Probleme auf, kann in jeder Phase ein Rücksprung auf die vorhergehende Phase erfolgen und die damit verbundenen Aktivitäten erneut ausgeführt werden.

Der Lebenszyklus beginnt mit der Phase „Servitization“, in der Services der Unternehmen im Netzwerk strukturiert aufgenommen und in einem Katalog mit Hilfe einer Service Map

(Glöckner, Augenstein et al. 2014) abgebildet werden. Die Hauptaufgabenstellungen dieser Phase stellen somit die Analyse und Design (Erl 2008), das Identifizieren und Modellieren (Papazoglou und Van Den Heuvel 2006) sowie die Konzeptualisierung (Blake 2007) der Services dar. Als Ergebnis der Phase stehen dem Integrator die (atomaren) Bausteine in einem Repository zur Verfügung, mit deren Hilfe kundenindividuelle, logistische Dienstleistungen entwickelt werden können. Außerdem dient diese Phase dazu neben den Services gleichfalls die bereitstellenden Unternehmen im Netzwerk zu registrieren. Im Gegensatz zu den folgenden Phasen wird diese nicht regelmäßig durchlaufen, sondern nur initial zur Erstellung des Netzwerks bzw. immer dann, wenn neue Dienstleister oder Services aufgenommen werden sollen oder müssen.

In der Phase „Development“ werden die zuvor bereitgestellten Services zu einer Gesamtleistung komponiert. Die Hauptaufgaben dieser Phase sind daher Entwicklung und Tests (Blake 2007; Gu und Lago 2007; Erl 2008), das Publizieren der Artefakte innerhalb der Systemlandschaft (Papazoglou und Van Den Heuvel 2006; Gu und Lago 2007) und die Orchestrierung der einzelnen Services (Gu und Lago 2007). In dieser Phase werden mehrere Werkzeuge und Modelle eingesetzt, um die Gesamtleistung zu entwickeln. So wird etwa zunächst ein Prozessmodell zur Ablaufbeschreibung erstellt und nach erfolgter Grobplanung diese mit Hilfe einer Simulationsumgebung auf Schwachstellen bzw. auf Engpässe oder Überkapazitäten zur Synergiebildung untersucht. Im Ergebnis werden in dieser Phase die benötigten Services bzw. deren Leistungserbringer identifiziert und so komponiert, dass eine individuelle, den Anforderungen genügende, logistische Dienstleistung (etwa die Realisierung einer Supply Chain) entwickelt werden kann.

In der Phase „Operation“ wird die zuvor entwickelte Gesamtleistung physisch als auch informationstechnisch durchgeführt, d.h. die Fracht wird entsprechend den Kundenanforderungen behandelt (transportiert, gelagert, verpackt, etikettiert, etc.) und gleichzeitig werden vorgelagerte, begleitende und nachgelagerte Informationsflüsse zwischen Integrator und den beteiligten Dienstleistern durch den Integrator koordiniert. Die Hauptaufgaben dieser Phase sind Deployment und Ausführung (Papazoglou und Van Den Heuvel 2006; Blake 2007), Monitoring (Papazoglou und Van Den Heuvel 2006; Gu und Lago 2007) sowie Durchführung der Zahlungsmodalitäten (Blake 2007). Insbesondere ist das Monitoring, also die Überwachung der Ausführung von Bedeutung, da zum einen der effiziente Ablauf überwacht werden kann und zum anderen Profile der beteiligten Dienstleister erstellt bzw. angepasst werden können. Diese dienen dazu passende Unternehmen auszuwählen bzw. Simulationen adäquat zu parametrisieren. Auch dient die Überwachung dazu im Fehlerfall (z. B. Zeitverzug oder Schäden) angemessen reagieren zu können. Eng damit

verbunden ist ebenfalls die (leistungsabhängige) Vergütung der Dienstleister, abhängig von der erbrachten Leistungsqualität.

Die letzte Phase „Retirement“ dient der geordneten Beendigung der logistischen Dienstleistung am Ende der Laufzeit des Vertrages. Die Hauptaufgaben lauten dementsprechend Wartung (Gu und Lago 2007) und Beendigung bzw. Zurücksetzen der diversen Dienstparameter (Blake 2007). Darüber hinaus werden im Anschluss an die erfolgreiche Beendigung die Leistungsparameter der Dienstleister tiefergehend analysiert, um die bereits angesprochenen Profile mit den neu gewonnenen Daten zu aktualisieren.

Die werkzeughafte Unterstützung dieses Lebenszyklus wird über eine integrierte Werkbank mit mehreren Werkzeugen zu den Lebenszyklusabschnitten gewährleistet. Jedes der Werkzeuge ist unmittelbar auf eine der Phasen zugeschnitten und trägt zur Lösung spezifischer Probleme bei. Somit ist mittels LSEM die ingenieurmäßige Erstellung von komplexen, logistischen Dienstleistungen über den vorgestellten Lebenszyklus möglich. Zur Steigerung der Flexibilität im Einsatz und insbesondere zur Unterstützung kleiner und mittlerer Unternehmen ist das langfristige Ziel, sämtliche Werkzeuge innerhalb einer Cloud-Infrastruktur „on demand“ anzubieten, wodurch die Anforderungen an die jeweilige Unternehmensinfrastruktur gesenkt werden können.

9.2.2 Fallbeispiel

Das im Folgenden beschriebene Fallbeispiel aus der Praxis soll die Zusammenhänge bei der Planung einer logistischen Dienstleistung zeigen, bei der mehrere Dienstleister kollaborativ eine Leistung für einen Kunden erbringen. Das fokale Unternehmen, das eine Lieferkette für einen Kunden plant und durchführt ist in diesem Beispiel ein 4PL als Systemintegrator (vgl. 8.1). Ihm fällt die Aufgabe zu, die Anforderungen und Wünsche eines Kunden auf Segmente einer Lieferkette abzubilden, die geeignetsten Dienstleister auszuwählen, die Gesamtleistung zu planen und deren Ausführung zu überwachen, um im Fehlerfall Abweichungen gegensteuern zu können. Der 4PL tritt im Verhältnis zum Kunden damit als eine Art Generalunternehmer auf, der für den Kunden transparent Subaufträge an spezialisierte Dienstleister vergibt. Die Basis der unterschiedlichen Aktivitäten bildet der LSEM-Gesamtansatz und als technische Grundlage die LSEM-Plattform.

Die Planung und Ausführung eines Kundenauftrags wird auf der Plattform mit Hilfe von Methoden, Modellen und Werkzeugen unterstützt. Die Plattform dient dabei als integrierte Werkzeugumgebung. Entlang des in Exkurs II beschriebenen vierphasigen Service-Lebenszyklus werden Services aus bestehenden Prozessen und Systemen als funktionale Einheiten modelliert, in Kundenaufträgen des 4PL eingeplant und ausgeführt und abschlie-

ßend die laufenden Instanzen der Services kontrolliert beendet. Jede Phase bedingt die Nutzung von Werkzeugen der Plattform. Dabei werden auf fachlicher Ebene Services in einem Prozess orchestriert und mit Hilfe von Daten aus vergangenen Aufträgen der Ablauf simuliert. Jedes Werkzeug nutzt dabei ein spezifisches Modell für Services, um die wesentlichen Informationen zu definieren. Die Aufgabe der modellgetriebenen Servicebeschreibung liegt dann in der Integration dieser verschiedenen Modelle. Das Anwendungsszenario beschreibt den Verlauf einer Grobplanung einer Dienstleistung. Sämtliche zuvor notwendigen Aktivitäten, wie die Registrierung und Beschreibung von Dienstleistern zum Zwecke der Einbindung in eine Planung werden hier außer Acht gelassen und als gegeben vorausgesetzt. Ebenso gilt das Szenario als abgeschlossen, sobald die Planung abgeschlossen ist. Die daran anschließenden Phasen im Lebenszyklus (Ausführung und Beendigung) sind ebenfalls nicht Bestandteil der Betrachtung. Zum besseren Verständnis der Aktivitäten innerhalb der Grobplanung wird mit dem Fallbeispiel auf den Entwicklungsaspekt einer Logistikdienstleistung eingegangen sowie Schritte und Modelle beschrieben, die innerhalb der Planungsphase auftreten:

Ein Produktionsunternehmen benötigt für die Endmontage an den Fertigungsbändern diverse Kleinteile (z. B. Schrauben, Befestigungsmaterial, Spritzgußkleinteile, Sicherungen). Aus Kostengründen bezieht es diese von mehreren Anbietern aus Asien. Dazu spezifiziert es Anforderungen (bspw. Art der Güter, durchschnittliche Bestellmengen, spezielle Verpackungen, Lieferzeiten/Durchlaufzeiten, Start-/Zielort, usw.), die beispielhaft in Tabelle 9-1 (links) definiert sind. Als Generalunternehmer übernimmt ein 4PL die Beschaffungslogistik und führt aufgrund der Kundenanforderung eine Grobplanung der Lieferkette durch (s. Abbildung 9-3).

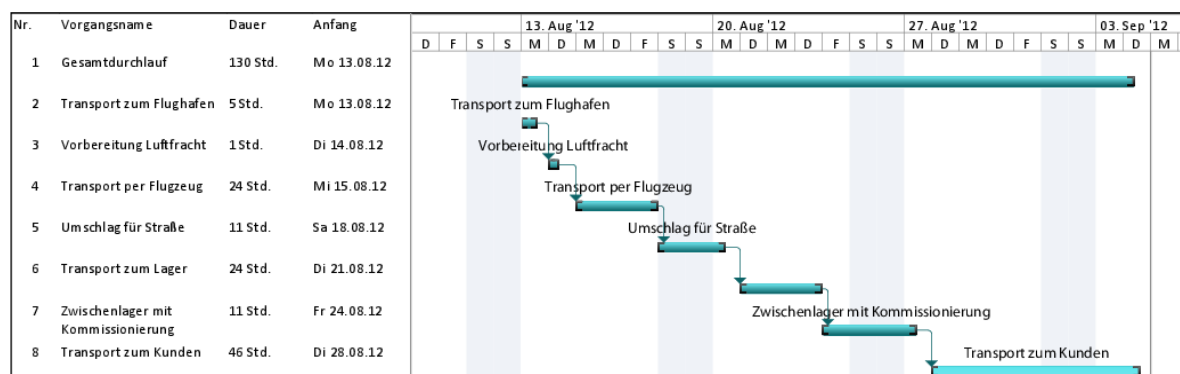


Abbildung 9-3: Zeitliche Grobplanung des 4PL

Im Beispiel werden die Kleinteile in zwei Werken hergestellt und anschließend von den Produzenten in den Werken auf Paletten verpackt. Der 4PL ist ab diesem Zeitpunkt für die gesamte Logistik bis zum Kunden verantwortlich. Im ersten Schritt muss der Transport

vorbereitet werden. Dazu gehören die Zollabwicklung und das Einholen aller benötigten Papiere. Danach werden die Paletten im Werk verladen und zum Flughafen per LKW transportiert. Am Flughafen angekommen, werden die Paletten in spezielle Container luftfrachttauglich verladen und für den Lufttransport vorbereitet. Die Container werden per Flugzeug nach Deutschland transportiert. Am Zielflughafen werden die Kleinteile wieder auf Paletten verpackt und per LKW in ein Zentrallager befördert und nach einer Qualitätsprüfung eingelagert. Aus dem Zentrallager werden gemäß der durchschnittlichen Bestellmenge die Zwischenlager an den Produktionsstandorten des Produktionsunternehmens und schließlich die Produktionsbänder „Just in Time“ per LKW beliefert.

Transportgut	Autozubehör			Umschlag	
Bestellmenge	1 Palette/36 Std.	Name der Dienstleistung	Transport Flughafen Seoul	Luftfracht Flughafen Seoul	Transport Flughafen Berlin
Durchlaufzeit	6 Tage	Art der Dienstleistung	Transport Straße	Umschlag Luftfracht	Transport Luft
Servicegrad	98%	Durchlaufzeit	5 Std.	1 Std.	24 Std.
Produktionsort	Südkorea	Servicegrad	99,50%	99,80%	99,50%
Zielort	Kassel (DE), Toledo (ES), Sibiu (RO), Östersund (SW)	Startort	Busan	Seoul	Seoul
		Zielort	Seoul	Seoul	Berlin

Tabelle 9-1: Kundenanforderung (links), Anforderung an Teildienstleistung (rechts)

Die zeitliche Grobplanung ist zugleich die Grundlage für die Identifizierung der Teilleistungen und damit für die Anbietersauswahl. In einem weiteren Schritt werden die Anforderungen des Kunden auf Anforderungen an die Teilleistungen herunter gebrochen (s. Tabelle 9-1 rechts) und potenzielle Anbieter auf Basis der verfügbaren Dienstleistermodelle ausgewählt. Das Verfahren zum Vergleich der Anforderungen mit den Kapazitäten der Anbieter wird aus Platzgründen nicht näher beschrieben.

Das Ergebnis der Grobplanung ist ein ausführbares Simulationsmodell der Logistikkette (s. Abbildung 9-4, repräsentiert durch ein Modell in Enterprise Dynamics⁵ (ED)). Führt der 4PL das Simulationsmodell aus, stellt er fest, dass innerhalb des Prozesses ein Engpass besteht. Die Anforderungen an die Teilleistung für den Transport vom Flughafen zum Zwischenlager müssen daher nachträglich angepasst werden, da nach Einschwingen des Systems die Auslastung des Dienstleisters demnach 99,8% beträgt und somit das Lager am Flughafen vollläuft. Die Grobplanung ermöglicht daher bereits im Vorfeld potenzielle

⁵ incontrol Simulation Solutions – <http://www.incontrolsim.com/>

Probleme zu erkennen und Alternativen auszuwählen. Daraufhin können Anforderungen angepasst und bspw. ein neuer Dienstleister ausgewählt werden. Eine weitere Möglichkeit besteht darin, einen zusätzlichen Service und somit einen weiteren Prozessschritt für die Logistikkette zu modellieren. Der Einsatz der LSEM-Werkzeuge und der modellgetriebenen Servicebeschreibung ermöglicht in diesem Beispiel eine effiziente Erzeugung und Ausführung mehrerer Simulationsmodelle zur Anpassung und Überprüfung der Gesamtleistung.

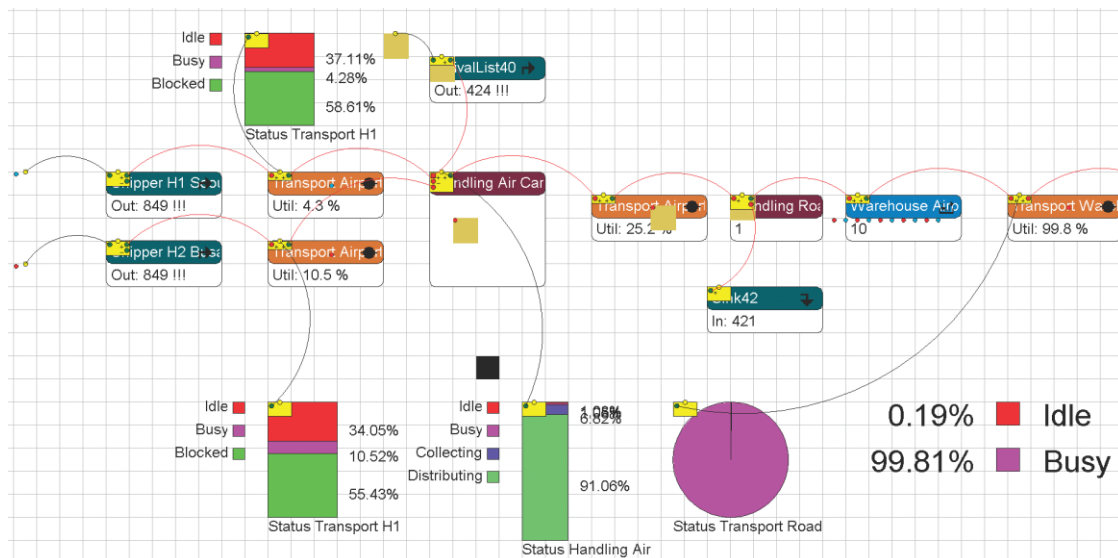


Abbildung 9-4: Simulationmodell (Ausschnitt) des Beispielszenarios in ED

Das hier beschriebene Szenario soll zeigen, dass bereits auf einer sehr grobgranularen Ebene unterschiedliche Informationen einbezogen werden müssen, die teilweise unabhängig voneinander sind (Prozessplanung, Dienstleisterauswahl), jedoch bei gewissen Planungsschritten kombiniert werden müssen (Prozesssimulation).

Die modellgetriebene Servicebeschreibung unterstützt Modellierer bei ihrer Arbeit mit unterschiedlichen Modellen, indem eine einheitliche Zugriffsweise auf Modelle und auf einzelne Informationen aus diesen Modellen zur Verfügung gestellt wird. Im nachfolgenden Abschnitt 9.3 wird der hierzu entwickelte Prototyp beschrieben und die Nutzung im Kontext des Fallbeispiels aufgezeigt. Mit Hilfe des Prototyps wird ausgehend von mehreren vorhandenen Modellen ein generisches Simulationsmodell erzeugt, aus dem in einem weiteren Schritt ein konkretes Simulationsmodell (s. Abbildung 9-4) generiert werden kann.

9.3 Prototypische Validierung

Die praktische Validierung des SMF erfolgt über einen, während der Entwurfsphase konzipierten, Prototyp, der Kernkomponenten des Rahmenwerks realisiert. Wie bereits in Ka-

pitel 7 beschrieben sind die Persistierung, die Modellierung als auch die Modellintegration darunter zu fassen. Der vollständige Prototyp umfasst diese drei Kernkomponenten und integriert diese in einen einheitlichen Rahmen zur Manipulation von Modellen. Die softwaretechnische Basis dafür bildet die Eclipse-Plattform mit ihrem Plugin-Mechanismus, der es erlaubt, die Komponenten als alleinstehende Plugins zu entwickeln und miteinander zu verwenden. Die Modellierungsbasis, insbesondere das Metametamodell, das zur Beschreibung aller Modellierungssprachen bei der Validierung benutzt wird, basiert auf dem Eclipse-Projekt „Eclipse Modeling Framework“ (EMF). In dieser konkreten Ausprägung der modellgetriebenen Servicebeschreibung wird auf das zu EMOF⁶ (Essential Meta Object Facility) kompatible Metametamodell Ecore⁷ des EMF zurückgegriffen. Ein Ausschnitt des Ecore-Modells ist in Abbildung 9-5 dargestellt. Wie bei objektorientierten Programmiersprachen (z. B. Java) sind auch bei Ecore alle Konzepte von einem Objekt (EObject) abgeleitet. Diese Ähnlichkeit ist nicht zufälliger Natur, sondern gewollt, um aus Ecore-Metamodellen auch Modelle in Form von Java-Quellcode zu generieren. Im Vergleich zu UML bzw. den Metametamodellen MOF oder EMOF wurde bei Ecore die Mächtigkeit der Sprache signifikant verringert und bspw. einige Referenztypen wie Aggregationen, Realisierungen oder Abhängigkeiten entfernt. Ebenfalls sind bei Ecore nur unidirektionale, gerichtete Referenzen definiert, durch eine gegenseitige Referenz zweier Klassen kann aber auch eine bidirektionale, ungerichtete Referenz modelliert werden.

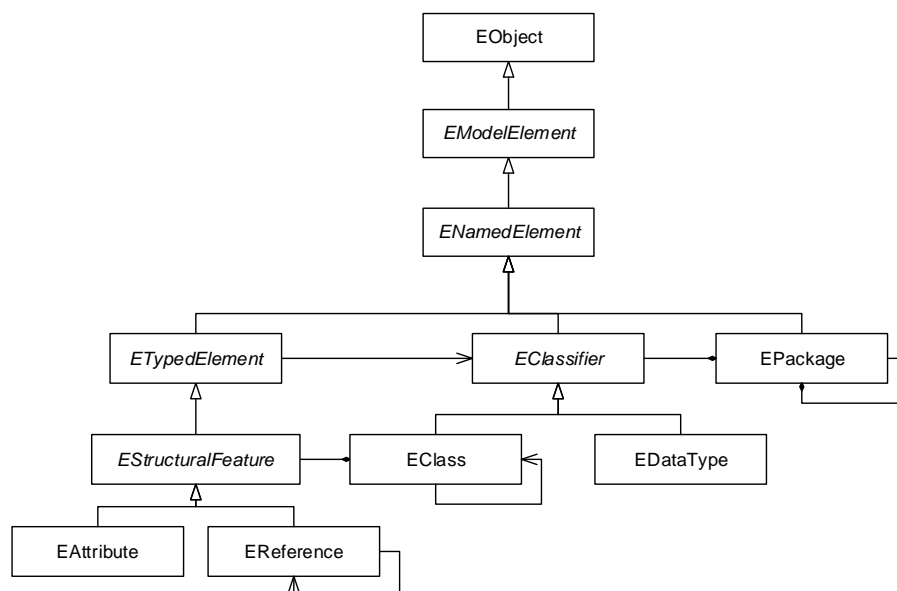


Abbildung 9-5: Vereinfachte Darstellung des Ecore-Metamodells (Ausschnitt)

⁶ Siehe hierzu: <http://www.omg.org/mof/>

⁷ Siehe hierzu: <http://www.eclipse.org/modeling/emf/?project=emf>

Mit Ecore ist es möglich, domänenspezifische Metamodelle zu entwickeln, die mit Hilfe des übergeordneten Frameworks die Grundlage für die Generierung von Werkzeugen bilden. Hierbei handelt es sich häufig um Editoren, mit denen aus den modellierten Zusammenhängen des Metamodells valide Modelle erzeugt werden können. Der Service Editor in 9.3.4 basiert teilweise auf dieser Eigenschaft des EMF. Die weite Verbreitung von EMF hat darüber hinaus den Vorteil, dass bereits einige Modellierungssprachen mit Hilfe von Ecore modelliert wurden und so auch im Rahmen der modellgetriebenen Servicebeschreibung genutzt werden können. Die Ableitung von Ecore aus MOF erlaubt aber auch eine sehr große Bandbreite von Sprachen mittels Ecore abzubilden, so dass mit der Wahl von Ecore keine Einschränkungen zu erwarten sind.

Die nachfolgenden Abschnitte widmen sich den einzelnen Komponenten, aus denen der Gesamtprototyp besteht. Zu jeder Komponente werden eine funktionale Beschreibung, die Integration mit anderen Komponenten und eine Architekturübersicht beschrieben. Abbildung 9-6 zeigt vorab die Package-Struktur. Der gesamte Prototyp besteht aus drei grafischen Komponenten (`de.lsem.repository.ui.*`) sowie aus den beiden zentralen Komponenten `.model` und `.core` und der Hilfsbibliothek `.api`. Die Abhängigkeiten sind so gestaltet, dass jeweils nur Abhängigkeiten zu den zentralen Komponenten existieren, jedoch nicht umgekehrt.

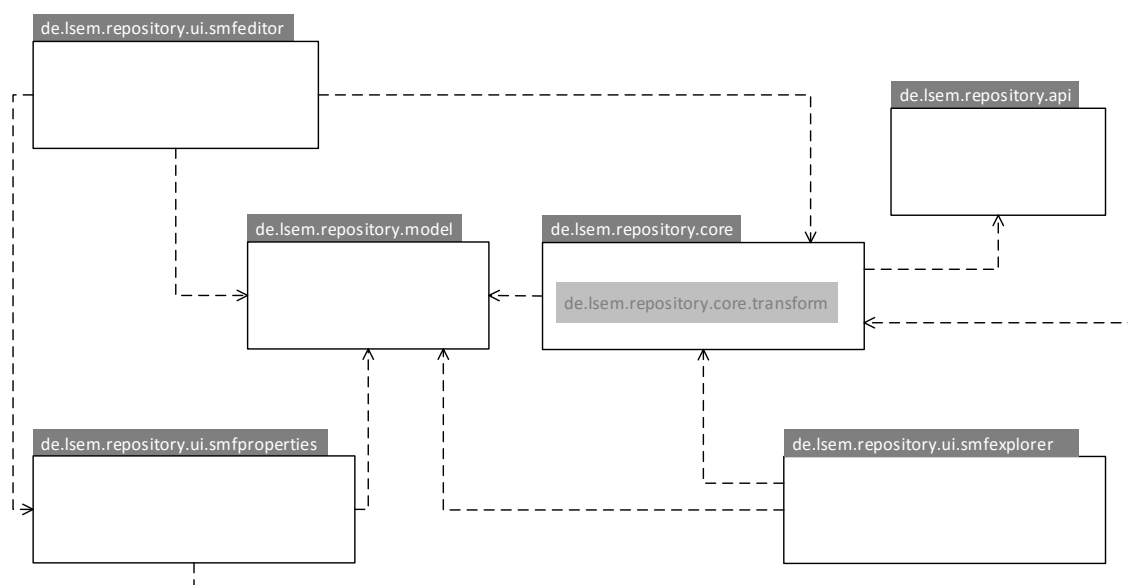


Abbildung 9-6: Package-Abhängigkeiten zwischen Komponenten

9.3.1 Basisfunktionen (SMF-Core)

Das „Core“-Package stellt das Bindeglied zwischen den drei Hauptkomponenten zur Speicherung, Modellierung und Integration dar. Es beinhaltet Basis- und Querschnitts-

funktionalitäten, die von mehreren Komponenten genutzt werden, wie bspw. APIs zu einem zentralen Speicher für Modelle, einem einheitlichen Zugriffsschema auf die Modelle sowie einer Reihe von Transformationsstrategien (enthalten in `de.lsem.repository.core.transform`, s. auch 9.3.5) für die Informationsübertragung zwischen Modellen. Außerdem enthält es ein zentrales Exception-Management, mit dessen Hilfe Laufzeitfehler aus allen Komponenten abgefangen und verarbeitet werden können und bietet eine Integration in die Eclipse-Verwaltung zur Konfiguration aller Laufzeitkomponenten, bspw. zur Hinterlegung von Nutzer-spezifischen Passwörtern.

Abbildung 9-6 zeigt, dass das „Core“-Package selbst eine Abhängigkeit zum Package `de.lsem.repository.api` besitzt. Dies ist zugleich eine eigenständige Komponente in der die konkrete Implementierung zur Kommunikation mit dem zentralen Speicher hinterlegt ist. Die API innerhalb des „Core“-Packages kapselt diese wiederum, so dass auch bei der einer technologisch bedingten Änderung der Zugriff für alle anderen Komponenten stabil bleibt und auch das „Core“-Package selbst nicht verändert werden muss. Daneben sind wichtige Sub-Packages:

- `de.lsem.repository.core.exception` - Exception Hierarchie und zentrales Fehlermanagement für alle Komponenten
- `de.lsem.repository.core.model` - Initialisieren von EMF-spezifischen Ressourcen zum Bearbeiten von Modellen
- `de.lsem.repository.core.transform` - Beinhaltet Transformationsstrategien zur Extraktion und Speicherung von Informationen aus unterschiedlichen Modellen

9.3.2 Modellzugriff (SMF-Model)

Das „Model“-Package beinhaltet eine lokale Kopie der verwendeten Modelle, vor allem solche, die wie das Basismetamodell notwendig für die allgemeine Funktionsfähigkeit sind. Die gesamte Komponente bildet außerdem den Container für den vom EMF generierten Modell-Quellcode. Dazu ist zunächst die Erzeugung eines Zwischenmodells, des so genannten Generatormodells (`genmodel`) notwendig, indem für den Generator zusätzlich notwendige Informationen hinterlegt werden. Der EMF-Generator wird dadurch so konfiguriert, dass er in der „Model“-Komponente die Java-API für den Modellzugriff generiert, d.h. dort alle Klassen erzeugt, die auf der Modellebene den Zugriff auf die modellierten Informationen steuern (s. Abbildung 9-7). Aufgrund der Komplexität kann jedoch nicht im Detail auf die Fähigkeiten des EMF eingegangen werden. Zum Verständnis sei auf die entsprechenden Webseiten bzw. auf vorhandene Literatur verwiesen.

Mit Hilfe der so generierten APIs wird der Schritt der Modellintegration bzw. der Informationsübertragung ermöglicht. Eine wichtige Rolle spielt dabei die im XMI-Format serialisierte Form eines Modells. So genannte EMF-Ressourcen repräsentieren ein Modell und erlauben, via der generierten API, den Zugriff sowohl auf die Attribute eines Modells als auch auf die darin enthaltenen Werte. Auf diese Art und Weise können die beschriebenen Informationen auch in andere Modelle übertragen werden. Die Container-Eigenschaft der Komponente bedingt, dass durch Generation von Quellcode, nur Packages enthalten sind, die die APIs der jeweiligen Metamodelle repräsentieren, sowie eine Kopie der Metamodelle selbst.

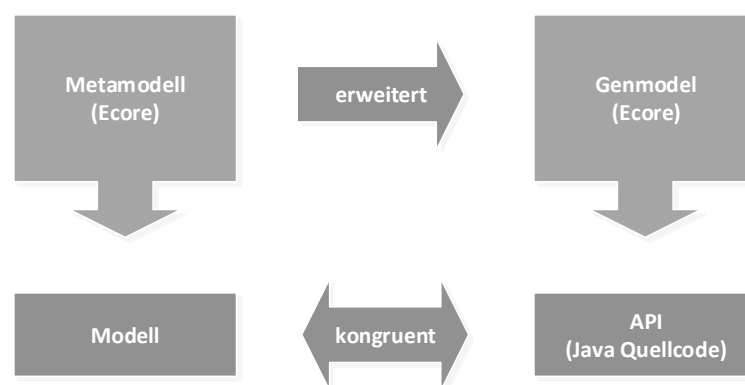


Abbildung 9-7: Quellcodegenerierung für Modellzugriff

9.3.3 Service Repository (SMF-Explorer)

Die Repository-Komponente des SMF ist eine visuelle Komponente und verantwortlich für die Darstellung des Repository-Inhalts und für die Speicherung und Bereitstellung von Modellen bzw. Metamodellen, die im Rahmen der LSEM-Plattform eingesetzt werden. Sie stellt die grundlegenden Operationen zur Manipulation (CRUD, create, read, update, delete) von Modellen zur Verfügung und abstrahiert zugleich von der konkret eingesetzten Speichertechnologie. Die Komponente ist als so genannte TreeView in Eclipse implementiert (s. Abbildung 9-8) und stellt sowohl Services (gelbe Ordnersymbole) als auch Modelle (blaue Symbole, inkl. Versionierung) dar.

Modelle und können zwischen Repository und Editor (s. 9.3.4) per Drag & Drop gezogen werden, wobei das Ziehen aus dem Editor zur Speicherung einer neuen Modellversion führt. Über einen Doppelklick können vorhandene Modelle auch in separaten, dafür zur Verfügung gestellten, Modelleditoren geöffnet werden, um diese bspw. zu bearbeiten. Die konkrete Implementierung der Speicherung wird für den einzelnen Nutzer transparent gestaltet. In einem ersten Prototyp wurde zur Persistierung ein so genanntes Java Content Repository verwendet, das in den Spezifikation JSR 170 bzw. JSR 283 definiert ist.

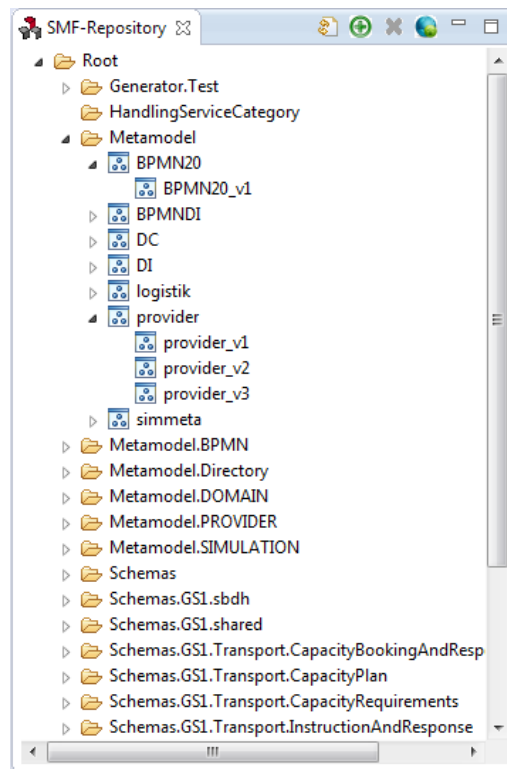


Abbildung 9-8: Repository-Komponente

Die Kommunikation zwischen dem Repository Explorer und dem Content Repository folgt dem in Abbildung 9-9 dargestellten Schema. Neben Klassen der Repository-Komponente sind, wie im Bild dargestellt, Klassen des Core- sowie des API-Packages an der Kommunikation beteiligt.

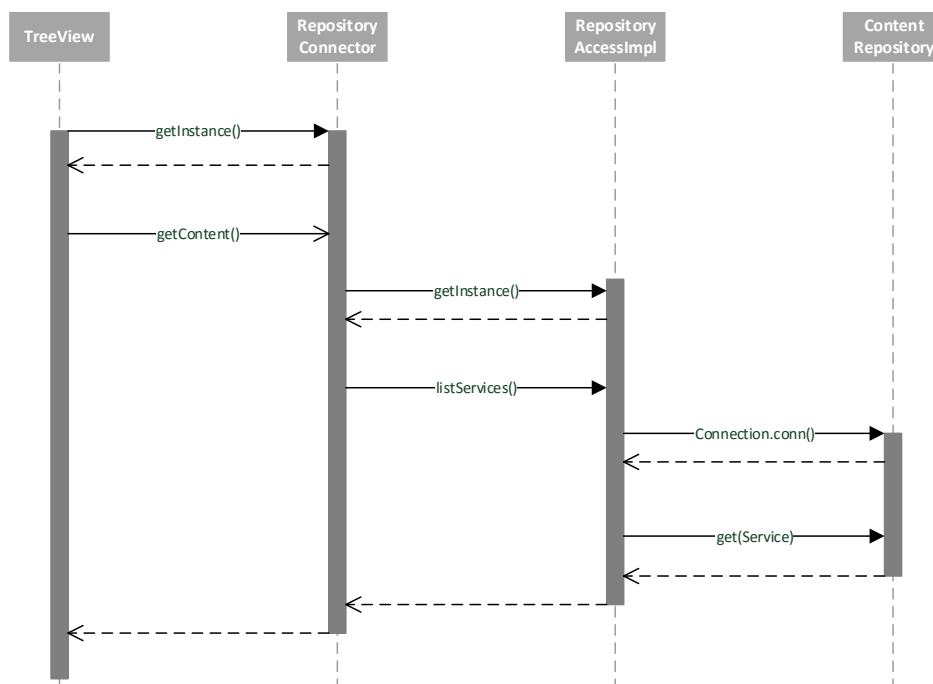


Abbildung 9-9: Kommunikation zwischen Komponenten

Die Architektur der Repository-Komponente ist um eine zentrale TreeView aufgebaut. Die Komponente ist als Singleton konzipiert, um Seiteneffekte beim Aufrufen von Services und Modellen zu verhindern. So genannte Provider füllen die View und erzeugen die Baumstruktur, unabhängig vom eigentlichen Speicherkonzept. Über einen Observer werden Änderungen an Modellen in der View kenntlich gemacht und zur Speicherung vorbereitet. Die Klassen des Observers sowie die verwendete Datenstruktur sind im `de.lsem.repository.core.model` enthalten, da diese auch für die Properties-Komponente als Bestandteil des Editors verwendet werden. Handler sind verantwortlich für die Ausführung der Änderungen wie bspw. das Erstellen bzw. Löschen von Services oder Modellen oder deren Aktualisierung im Repository über den in Abbildung 9-9 dargestellten „RepositoryConnector“ des Core-Packages. Die Repository-Komponente implementiert außerdem einen spezifischen Drag&Drop-Mechanismus für die SMF-Modelle sowie die Identifizierung und Initialisierung der notwendigen Editoren zur Manipulation der Modelle.

9.3.4 Service Editor (SMF-Editor)

Die Editor-Komponente verfügt über insgesamt zwei visuelle Bestandteile, den Editor selbst (s. Abbildung 9-10) sowie einem zusätzlichen Fenster (s. Abbildung 9-11), indem zu jedem im Editor markierten Element Kontextinformationen bereitgestellt werden. Das Editor-Fenster ist im Bild zweigeteilt: in einem großen, karierten Bereich befindet sich das eigentlich Modellierungsfenster und in einem kleinen Bereich daneben die Elemente, die zur Modellierung verwendet werden können. Diese Elemente sind in Verbindungen und in Objekte (Services, Modelle und Modellelemente) untergliedert. Modelle können aber auch direkt aus dem Service Repository per Drag&Drop in das Editor-Fenster gezogen werden.

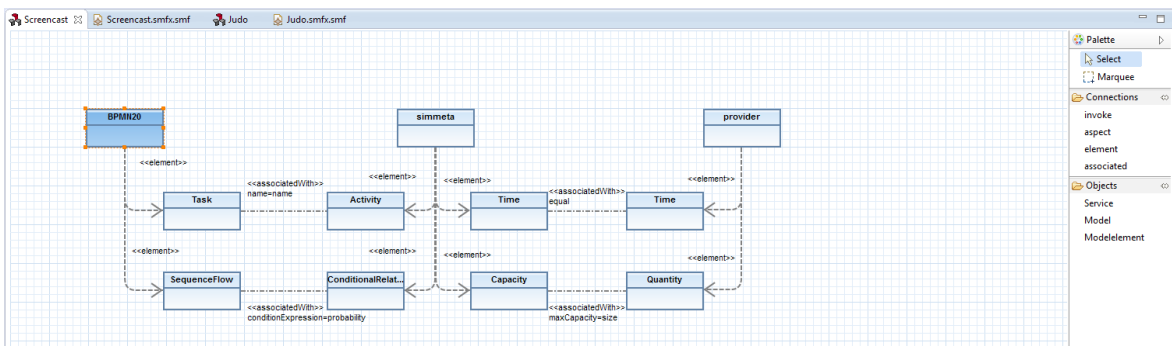


Abbildung 9-10: Editor-Komponente

Die Fähigkeiten des hier dargestellten Editors sind mächtiger als die in EMF automatisch zu jedem Metamodell generierbaren Editoren. Im zugrunde liegt das Eclipse Graphiti

Framework⁸, mit dessen Hilfe grafische Editoren entwickelt werden können. Im Vergleich zu den generierbaren Editoren müssen zum Referenzieren von Objekten (z. B. Modelle oder Modellelemente) keine zusätzlichen Dateien erzeugt und geladen werden. Bei der Implementierung wurde auch darauf geachtet, dass die grafischen Eigenschaften des Modells (z. B. Lage, Farbe oder Art eines Objekts) von den inhaltlichen Bestandteilen getrennt persistiert werden, so dass der eigentliche Zweck der Modellerstellung, ein multiperspektivisches Servicemodell, nicht durch zweckfremde Informationen beeinträchtigt wird. Durch die Einbeziehung des Basismetamodells bei der Entwicklung des Editors ist sichergestellt, dass nur valide Modelle erzeugt werden können. Eine eingebaute Validierung stellt dies sicher. Da das Basismetamodell zugleich das Metamodell des multiperspektivischen Servicemodells ist, sind mit dem SMF-Editor konstruierte Modelle automatisch multiperspektivische Servicemodelle.

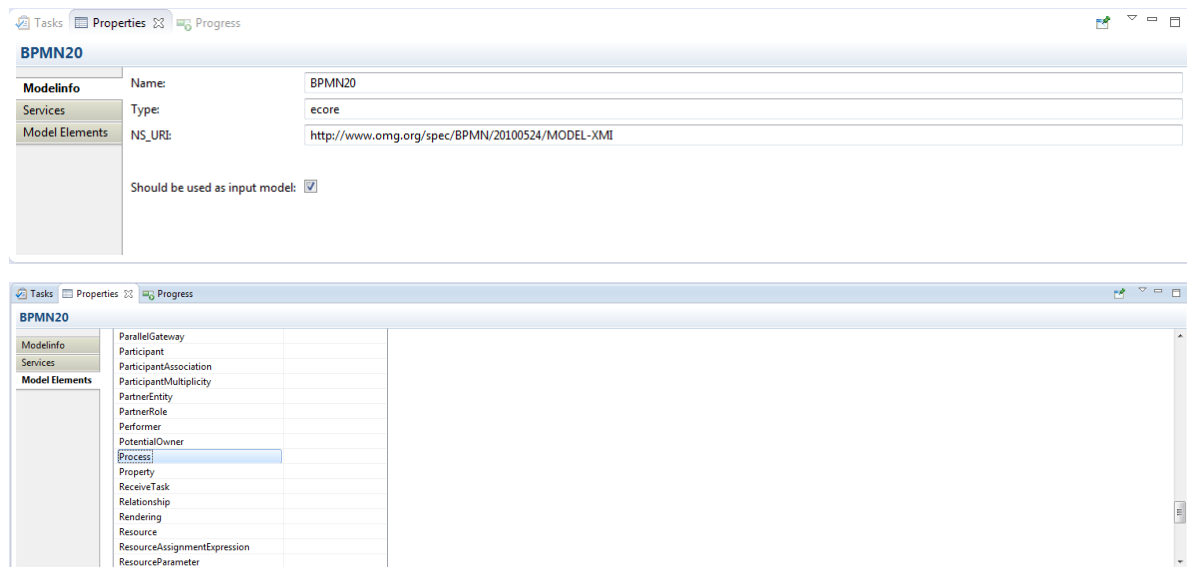


Abbildung 9-11: Properties-Komponente

Die in der Properties-Teilkomponente des Editors bereitgestellten Kontextinformationen sind abhängig vom aktuell ausgewählten Element. In Abbildung 9-11 sind bspw. Informationen für das BPMN-Metamodell dargestellt. Im Bild oben werden Informationen zum Modell selbst (Name, Typ sowie URI) angezeigt. Im Bild unten sind dagegen Modellelemente des Metamodells aufgelistet, die zur Modellierung im Editor-Fenster zur Verfügung stehen, aber bislang noch nicht verwendet wurden und können per Drag&Drop in das Fenster gezogen werden.

⁸ Ref. <http://www.eclipse.org/graphiti/>

Beide Teilkomponenten setzen bzgl. der Architektur auf den jeweiligen Frameworks (EMF, Graphiti bzw. Eclipse tabbed properties) auf. Die Objekte (Service, Modell oder Modellelement) im Editor folgen dabei einem spezifischen Lebenszyklus, der mit Hilfe von Klassen auch dementsprechend implementiert ist. So werden Objekte zunächst erzeugt, d.h. als Element im Servicemodell instanziiert, bevor sie im Fenster als grafisches Element hinzugefügt werden. Dies gilt ebenfalls für die unterschiedlichen Relationen, die zwischen den Objekten modellierbar sind. Darüber hinaus stehen Klassen zur Verfügung, die eine Aktualisierung oder Löschung der Objekte bzw. Relationen ermöglichen. Utility-Klassen sind verantwortlich für Dialoge mit dem Nutzer oder für das Design, also die grafische Repräsentation der Objekte. Spezielle Listener sorgen dafür, dass zu ausgewählten Objekten im Editor-Fenster die korrekten Kontextinformationen in der Properties-Teilkomponente dargestellt werden. Eine transaktionale Behandlung der Objekte sorgt dafür, dass Änderungen in den Eigenschaften auch in den Objekten des Editors persistiert werden.

9.3.5 Modellintegration (SMF-Core)

Die Modellintegration ist keine visuelle Komponente und Bestandteil des Core-Packages (`de.lsem.repository.core.transform`). Die Komponente ist dafür verantwortlich, dass ausgehend vom modellierten, multiperspektivischen Metamodell (Artefakt des Service Editors) für einen Service auf Modellebene die Informationsextraktion und -übertragung durchgeführt wird, d.h. die transparente Informationsmodellierung realisiert wird. Dies geschieht in drei Phasen:

Zunächst werden die betroffenen Quell- und Zielmodelle anhand der Beschreibung im Servicemetamodell identifiziert. EMF unterstützt eine Identifikation über die Verwendung von URIs (Universal Resource Identifier) als Pflichtfeld bei der Definition von Modellen. Anschließend wird überprüft, ob im Model-Package bereits die API für die identifizierten Modelltypen generiert wurde. Falls nicht werden die noch nicht vorhandenen Klassen generiert. Über diesen Mechanismus können auch unterschiedliche Versionen der betroffenen Metamodelle Berücksichtigung finden. Das URI-Konzept erlaubt nicht nur die Identifikation eines bestimmten Modelltyps, sondern auch eine Versionierung. Dementsprechend kann nach der Identifikation auch die API der verwendeten Version generiert werden.

In der zweiten Phase wird überprüft, ob alle vorgesehenen Zielmodelle für einen Service bereits im Repository vorhanden sind. Falls nicht, werden in dieser Phase alle Zielmodelle initialisiert. Im Gegensatz zur Generierung der API kann dies nicht über einen einfachen Mechanismus abgebildet werden, da die Komplexität der Modelle sehr unterschiedlich ist.

Stattdessen werden im Vorfeld Templates manuell erzeugt. In diesem Kontext sind unter Templates kleinstmögliche, zu ihrem jeweiligen Metamodell valide Modelle zu verstehen, d.h. alle Pflichtfelder sind mit Standardwerten versehen. Zusätzlich werden den Metamodellen so genannte OCL-Constraints (Object Constraint Language) hinzugefügt, die die Definition zusätzlicher Randbedingungen über rein strukturell vorhandene Bedingungen (z. B. Pflichtfelder) hinaus ermöglichen. Da OCL Bestandteil von UML ist und damit über MOF definiert wird, ist eine Integration mit Ecore und damit eine Annotation der Metamodelle möglich. Insbesondere werden Invarianten, initiale Werte sowie Definitionen der OCL genutzt, um Metamodelle zu annotieren und um daraus Templates auf der Modellebene zu entwickeln. Zum Abschluss der zweiten Phase werden sowohl Templates als auch bereits vorhandene Zielmodelle mit Hilfe des EMF Validation Frameworks auf Korrektheit überprüft. Diese Überprüfung berücksichtigt ebenfalls die OCL-Definitionen. Mit dieser Phase enden die vorbereitenden Schritte, so dass in der letzten Phase schließlich die Informationsextraktion aus den Quellmodellen und die Übertragung in die Zielmodelle durchgeführt werden kann.

Über das multiperspektivische Servicemetamodell sind sowohl die Modelltypen als auch die Modellelemente definiert, die bei der Informationsübertragung betroffen sind. Für jedes Quellmodell werden die relevanten Modellelemente und darin enthaltene Attribute ausgelesen und in die Elemente des Zielmodells übernommen. Sobald alle Quellmodelle abgearbeitet sind, ist die Übertragung abgeschlossen und eine erneute Validierung wird durchgeführt, um zu prüfen, dass bei der Informationsübertragung keine Konsistenzfehler aufgetreten sind.

Die Extraktion sowie Übernahme von Informationen in bestimmte Modelle wird über ein Strategiemuster und eine vom Basismetamodell abgeleitete bzw. erweiterte Datenstruktur realisiert. Für jeden Modelltyp wird dabei eine konkrete Strategie implementiert, die die jeweiligen Besonderheiten, im Sinne der dort vorhandenen Attribute und Einschränkungen, der Modelle berücksichtigt. Die Strategie wird ebenfalls über die spezifische URI der Modelle identifiziert und mittels einer Kontext-Klasse geladen. Zur Extraktion und Übernahme der Informationen greifen alle Strategie-Klassen auf dieselbe Datenstruktur zurück. Diese ist in Abbildung 9-12 abgebildet. Teile der Datenstruktur werden bereits in anderen Komponenten (z. B. im Repository) verwendet, für die Informationsmodellierung aber um ein „SourceModel“ sowie „ModelElement“ erweitert. Diese Elemente referenzieren die EMF Ressourcen und damit die persistenten Modelle und beinhalten die zu übertragenden Informationen und werden pro oben beschriebenem Schritt, also für jedes Quellmodell, initialisiert und befüllt.

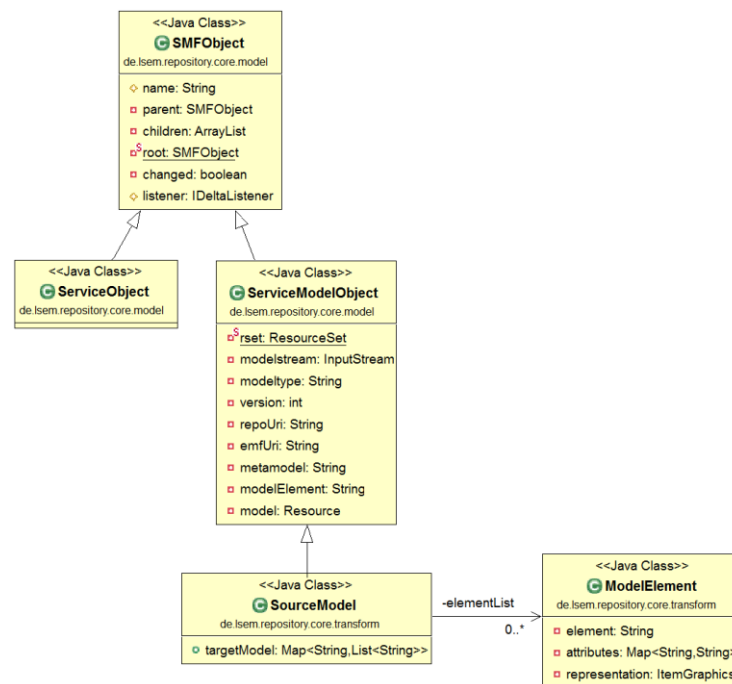


Abbildung 9-12: Datenstruktur zur Informationsmodellierung

9.3.6 Umsetzung des Fallbeispiels

Ausgehend vom beschriebenen Prototyp wird nun die Umsetzung des Fallbeispiels gezeigt. Die Validierungsmodelle in Anhang 1 dienen dabei als Vorlage für die durchzuführende Modellierung mit dem Basismetamodell. Auf der Metamodellebene repräsentiert das Simulationsmetamodell in Anhang 1.1 das Zielmodell, in das Informationen übertragen werden sollen. Das Dienstleistermetamodell in Anhang 1.2 sowie das standardisierte BPMN repräsentieren die Quelle für die Informationen. Die Beschreibung wichtiger Elemente des Prototyps in Anhang 2 soll die Beschreibung der Umsetzung im Folgenden unterstützen.

Aus dem Service Repository (9.3.3) werden die genannten Metamodelle in den Service Editor (9.3.4) geladen. Sowohl das BPMN- als auch das Providermetamodell werden als Inputmodell spezifiziert. Abbildung 9-13 zeigt einen Auszug der notwendigen Modellierung von Beziehungen zwischen Modellen. Auf der obersten Ebene finden sich die genannten Metamodelle. Zu jedem Metamodell werden exemplarisch zwei bzw. vier darin definierte Modellelemente modelliert. Für je zwei Elemente der Inputmodelle werden in diesem Beispiel je zwei Elemente des Zielmodells benötigt. Für BPMN sind dies Task und SequenceFlow, für die Simulation Activity, ConditionalRelation, Time und Capacity und für das Dienstleistermodell Time und Quantity. Jedes der Modellelemente ist über eine <<element>> Beziehung mit dem Metamodell verknüpft. Zwischen Modellelementen unterschiedlicher Modelle können eine oder mehrere <<associatedWith>> Beziehungen spezifiziert werden. Für die spätere Ausführung der Transformation wird die <<associated-

With>> Beziehung zusätzlich annotiert. Eine spezielle, „equal“ genannte, Annotation (Äquivalenzrelation) weist den Interpreter des Servicemetamodells an, alle Attribute, die in einem Modellelement eines Inputmodells vorhanden sind, auf gleichnamige Attribute im spezifizierten Modellelement des Zielmodells zu übertragen bzw. nicht vorhandene Attribute zu erzeugen. Andernfalls wird für jedes zu modellierende Attribut eine <<associatedWith>> Beziehung modelliert und die Namen der Attribute als Annotation ergänzt.

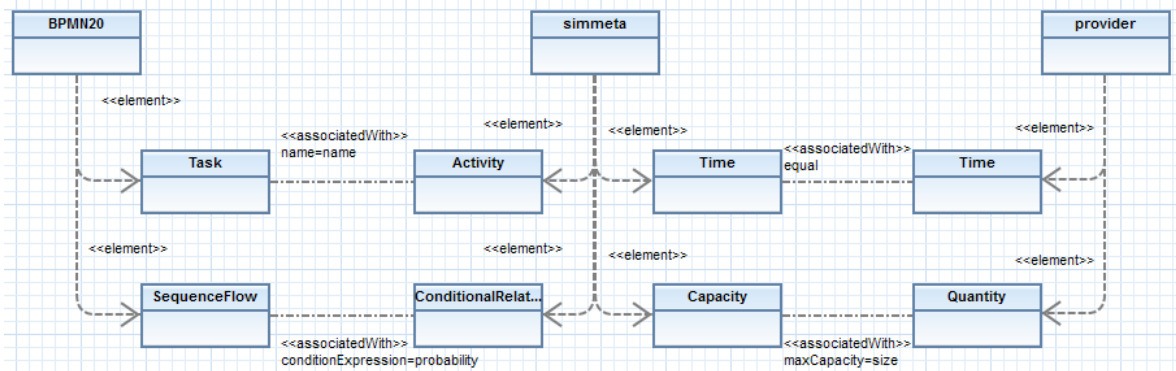


Abbildung 9-13: Modellierung von Beziehungen zwischen Modellen

Nachdem alle betroffenen Modelle und Modellelemente auf diese Art und Weise im Servicemetamodell hinterlegt sind, muss in einem letzten Schritt ein Service ausgewählt werden, der u.a. mit Hilfe der drei Modelltypen bereits beschrieben wurde. Konkret bedeutet dies, dass ein Service eine Art Container für Modelle repräsentiert. Sobald dieser ausgewählt ist, wird geprüft, ob die notwendigen Inputmodelle vorhanden sind und ob das Zielmodell bereits angelegt wurde. Falls nicht wird, wird über ein Template ein Modell erzeugt und die Attribute in den modellierten Beziehungen entsprechend befüllt.

9.4 Bewertung

Die abschließende Bewertung des Ansatzes wird über einen Vergleich mit den in Kapitel 4 dargestellten, verwandten Ansätzen durchgeführt und die Vergleichstabelle dementsprechend um eine Spalte erweitert (s. Tabelle 9-2). Nachfolgend wird dazu explizit auf die einzelnen Eigenschaften eingegangen und die Bewertung erläutert.

Kategorie / Eigenschaft	Service Modeling Framework	(Nardi, de Almeida Falbo et al. 2013)	(Berre, Lew et al. 2013)	(Steen, Iacop et al. 2012)	(Kett, Voigt et al. 2008; Scheithauer, Voigt et al. 2009)	(Quartel, Steen et al. 2007)	(Störrle und Glock 2007)	(Kunau, Junginger et al. 2005)
Modellierung								
Modellierungsart	M	O	M	M	M	M	M	M
Zahl vorhandener Perspektiven	beliebig	4	2	6	4	5	3	3
Einsatz standardisierter Modelle	ja/nein	ja	ja	Archi-Mate	ja	ja	ja	SeeMe
Erweiterbarkeit (Modelle)	ja	k.a.	ja	nein	nein	nein	ja	ja
Informationstransparenz	ja	ja	ja	ja	ja	ja	ja	nein
Abstraktionsniveau	gering	hoch	gering	gering	gering	mittel	gering	mittel
Rollenmodell								
Nutzergruppen	2	2	5	3	6	2	0	3
Rollenmodell	nein	nein	ja	ja	ja	ja	nein	ja
Zielgruppen	1	2	3	4	2	2	3	4
Anwendung								
Anwendungsgebiet	0)	1)	2)	3)	4)	5)	6)	7)
Anwendbarkeit	ja	nein	ja	ja	ja	ja	ja	ja
Anpassbarkeit	ja	ja	ja	ja	ja	ja	ja	ja

0) Servicebeschreibung, 1) Service Science and Services Computing, 2) Service Innovation and Realisation, 3) Agile Serviceentwicklung, 4) Service Marktplätze / Internet of Services, 5) Webdienstleistungen, 6) Business-driven SOA, 7) Service Engineering

Tabelle 9-2: Bewertung SMF

Die Kategorie Modellierung umfasst Eigenschaften des internen Aufbaus eines Ansatzes. Das SMF ist ein auf Modellen bzw. Metamodellen basierender Ansatz, der keinerlei Einschränkungen bzgl. der verwendeten Zahl an Perspektiven macht. Allerdings ergeben sich Einschränkungen aufgrund der Modellart bzw. des verwendeten Modellierungsparadigmas (vgl. Kapitel 2), da eine Integration von Modellen mit zu starken Unterschieden in der Art der Modellierung oder des Inhalts (z. B. Optimierungsmodelle, mathematische Modelle) keine sinnvollen Ergebnisse liefern. In der Regel sind deskriptive bzw. präskriptive Modelle am sinnvollsten. Bedingt durch die Wahlfreiheit der Perspektiven und den spezifischen Anforderungen und Einschränkungen ist der Einsatz standardisierter Modelle nicht eindeutig zu beantworten. Das SMF unterstützt ausdrücklich deren Verwendung und bejaht die Wiederverwendung bereits existierender Modelltypen. Prinzipiell ist es jedoch möglich

sämtliche durch das gewählte Metametamodell darstellbare Modelltypen zu verwenden, so dass auch nicht standardisierte Modelle zum Einsatz kommen können. Dies trifft auch auf einige bei der Validierung zum Einsatz kommenden Modelle zu. Die freie Wahl der Perspektiven führt im Weiteren auch zu einer Erweiterbarkeit des Ansatzes im Sinne des Hinzufügens neuer Modelle. Die Informationstransparenz ist sogar expliziter Bestandteil der Anforderungen des SMF, da die Integration von Modellen letztendlich zu einer durchgängigen Modellierung führt, bei der Informationen zwischen Modellen ausgetauscht werden sollen. Das Abstraktionsniveau ist überwiegend an die verwendeten Modelle gekoppelt. Das Basismetamodell selbst, mit dem eine Verbindung erreicht wird, hat aufgrund der wenigen, konkreten Konzepte ein niedriges Abstraktionsniveau.

Eigenschaften der Kategorie Rollenmodell beschreiben die Zahl und Art der unterschiedlichen Akteure bei der Nutzung des Ansatzes bzw. der Ergebnisse. Das SMF beinhaltet kein explizites Rollenmodell. Es wird jedoch davon ausgegangen, dass zumindest fachliche, spezialisierte Modellierer einzelne Teilmodelle zu einer integrierten Sicht beitragen und Domänenarchitekten die gewünschte Art der Integration über Instanzen des Basismetamodells herbeiführen. Spezifische Zielgruppen werden im SMF ebenfalls nicht definiert. Der Ansatz soll jedoch eine ganzheitliche Sicht auf Services ermöglichen und dadurch das operative Personal bei der Serviceentwicklung und -verwaltung unterstützen. Ob eine darüber hinausgehende Nutzung, etwa im Managementbereich, sinnvoll ist, kann derzeit nicht überprüft werden. Notwendig wäre dazu zunächst die Entwicklung geeigneter Modelle, mit deren Hilfe eine Unterstützung auf Managementebene geleistet werden kann.

Die Anwendung als letzte Kategorie umfasst schließlich Eigenschaften, mit deren Hilfe eine konkrete Anwendbarkeit bewertet werden kann. Das Anwendungsgebiet des SMF ist die allgemeine Servicebeschreibung mit Hilfe eines multiperspektivischen Ansatzes. Dadurch ergeben sich keinerlei Einschränkungen bzgl. der möglichen Anwendungsszenarien und letzten Endes, wie auch bereits dargelegt, keine Einschränkungen der Perspektiven. Die nahezu freie Wahl des Anwendungsszenarios und das geringe Abstraktionsniveau bzgl. der Modellierung ergeben somit auch eine sehr gute Anwendbarkeit des Ansatzes im operativen Bereich, die durch eine gute Anpassbarkeit des SMF noch weiter unterstützt wird. Letztere ist einerseits durch die zum Einsatz kommenden Modelle und andererseits durch die Integrationsweise gegeben. Die Verknüpfung von Modellen wird selbst als Modell abgebildet und kann jederzeit neuinterpretiert und somit aktualisiert werden.

10 Schlussbetrachtung und Diskussion

Dieses Kapitel widmet sich der abschließenden Betrachtung und Bewertung, des mit dieser Arbeit präsentierten Ansatzes zur Servicemodellierung. Nach einer Zusammenfassung der Inhalte (10.1) wird zunächst auf den Beitrag zur Forschung 10.2 mit Hilfe der gestellten Forschungsfragen eingegangen. In einer Schlussbetrachtung (10.3) wird schließlich ein Resümee der erzielten Resultate gezogen und eine kritische Würdigung deren vorgestellt. Insbesondere werden noch ungelöste Problemstellungen aufgezeigt, die auch im Hinblick auf weitere Forschungsmöglichkeiten zu anschlussfähigen Lösungsansätzen führen können.

10.1 Zusammenfassung

Die vorliegende Arbeit befasst sich mit der modellgetriebenen Servicebeschreibung, einem Bottom-up-Ansatz zur ganzheitlichen Beschreibung bzw. Modellierung von Services. Im Vergleich zu bestehenden Ansätzen (vgl. Kapitel 3.2 und 4), die im ersten Teil „Grundlagen“ enthalten sind, wird mit diesem Ansatz eine Integration bestehender Servicemodelle angestrebt, anstatt ein auf spezifische Zwecke ausgerichtetes Servicemodell zu entwickeln. Neben dem bereits schwer greifbaren Servicebegriff setzt sich die Arbeit mit spezifischen Modellierungsaspekten (vgl. Kapitel 2) und insbesondere mit der Zusammenführung verschiedener Modelle auseinander. Der „Service Modeling Framework“ genannte Hauptteil der Arbeit widmet sich unterschiedlichen Fragestellungen der multiperspektivischen Modellierung und Modellintegration, die in den Hypothesen und Forschungsfragen in Kapitel 1 verankert sind. Das Hauptaugenmerk liegt in einem transparenten Informationsübergang zwischen Modellen, so dass entweder Zusammenhänge bereits vorhandener Modelle besser zu Tage treten oder dass nicht vorhandene oder unvollständige Modelle durch Bestehende erstellt bzw. ergänzt werden können. Kapitel 5 zeigt hierzu die Anforderungen und Restriktionen an den Gesamtansatz. Das SMF wurde so konzipiert, dass Letztere möglichst gering ausfallen und so eine breite Anwendbarkeit gegeben ist. Allerdings sind teilweise bestehende Restriktionen wie die fehlende Einbeziehung eines Domänenmodells anderen Gründen geschuldet (s. 10.3). Kapitel 6 beinhaltet die Kernkonzepte des SMF. Zentral ist die Stellung des Basismetamodell genannten Servicemetamodells, mit dessen Hilfe unterschiedliche Servicemodelle integriert werden können. Der Fokus auf die Modellierung legt dabei nahe, dass auch eine Anwendbarkeit außerhalb der Servicemodellierung gegeben ist (vgl. auch 10.3). Kapitel 7 präsentiert drei wesentliche Komponenten, die für die Umsetzung des SMF und die zugrunde liegenden Konzepte unerlässlich sind: Persistenz, Modellierungssprache und Integration von Modellen. Analog zu Kapitel 5 werden ebenfalls An-

forderungen aufgezeigt. Sie beziehen sich jedoch ausschließlich auf die Umsetzung der Komponenten und nicht auf die Konzepte im Allgemeinen. Der dritte und letzte Teil „Anwendung und Validierung“ präsentiert zunächst eine Einführung in die Logistikdomäne (Kapitel 8), anhand derer die Anwendbarkeit der SMF-Konzepte und -Komponenten gezeigt wird. Darauf aufbauend beschreibt Kapitel 9 schließlich die Validierung. Diese wird zum einen über die erfolgten Publikationen, zum anderen über Fallbeispiele erbracht. Abschließend erfolgt eine Bewertung des SMF auf Basis der in Kapitel 4 bereits erhobenen Kriterien, wodurch auch eine Vergleichbarkeit hinsichtlich der verwandten Ansätze gegeben ist.

Die Anwendung des SMF-Ansatzes wird durch einen Prototyp gezeigt, die bereits in der Konzeptionsphase eingesetzt wurden, um die zu entwickelnden Konzepte zu erforschen. Diese Konzepte sowie Methoden und Werkzeuge erlauben es, eine Reihe von Ansätzen zur Beschreibung von Services in einem ganzheitlichen Ansatz zusammenzuführen und stellen dadurch eine effektivere Gestaltungsmöglichkeit dar.

10.2 Forschungsbeitrag

Der Beitrag zur Forschung ergibt sich, wenn man die Ergebnisse der Arbeit mit der Zielsetzung in Kapitel 1 vergleicht. Betrachtet man zunächst die Zielsetzung der Definition 1-1, die Konstruktion eines Rahmenwerks und dessen Umsetzung mittels Modellen, Editoren und Transformationsansätzen, so sind die relevanten Artefakte im Zuge der Erarbeitung geschaffen worden. Die Operationalisierung der Basishypothesen (s. Definition 1-2) zur Zielstellung erfolgte mit Hilfe von vier zentralen Forschungsfragen. Während auf die ersten beiden Fragen bereits hinreichend am Ende von Kapitel 4 eingegangen wurde, sollen die den Hauptteil betreffenden Forschungsfragen 3 & 4 (vgl. Definition 1-5 und Definition 1-6) abschließend erörtert werden.

Eine wichtige Erkenntnis stellt die Tatsache dar, dass Modelle im Allgemeinen nicht nur beitragen wichtige Konzepte einer Domäne zueinander in Beziehung zu setzen, sondern darüber hinaus auch Informationen mittels dieser Konzepte transportiert werden können, z. B. im Sinne einer Instanziierung von Metamodellen. Dadurch wird die Integration von Modellen zu einem wichtigen Forschungsgegenstand. Modelle beinhalten je nach Art und Zweck unterschiedliche Informationen, wobei der Informationsgehalt insgesamt entweder vollständig disjunkt oder aber zumindest in Teilen überlappend sein kann. Eine Integration führt dann dazu, dass die Überlappung von Modellen ausgenutzt und Informationen zwischen diesen übertragen werden, ohne jedoch eine vollständige Transformation auszuführen, bei der der Informationsgehalt durch das Austauschen des zugrundeliegenden Modell-

typs insgesamt gleichbleibt. Eine Integration von Modellen führt dagegen entweder zu einem steigenden Informationsgehalt in mindestens einem Modell oder aber zur expliziten Darstellung der Abhängigkeiten zwischen Modellen bei gleichbleibendem Informationsgehalt der beteiligten Modelle. Letzteres bedeutet, dass ein Modellierer spezifiziert, welche Informationen in unterschiedlichen Modellen enthalten sind und ersteres, dass die Spezifikation ausgenutzt wird, um Informationen zwischen Modellen auszutauschen.

Die Überführung von Modellen ineinander, d.h. die Übertragung von Informationen zwischen Modellen, kann mittels bereits existierender Transformationsansätzen erfolgen, wenn im Voraus alle Modelle bzw. deren Metamodelle bekannt sind. Die Anforderung an das SMF keine Einschränkung bzgl. der verwendeten Modellierungssprachen zu erheben, schränkt die Verwendbarkeit existierender Ansätze jedoch ein. Umso mehr, als das keine vollständige Transformation in der Regel benötigt wird, sondern nur gezielt Informationen zwischen Modellen ausgetauscht werden müssen. Eine Erkenntnis dieser Arbeit ist, dass eine solche Transformation nur gelingen kann, wenn deren Beschreibung nicht auf einem syntaktischen Vergleich, sondern auf einer fachlichen Beschreibung basiert. Auf syntaktischer Ebene ist eine Bewertung der darin enthaltenen Informationen nicht durchführbar. Vielmehr wird durch Transformationsregeln implizit festgelegt, welche Informationen enthalten sind. Je nachdem welche Modelle ineinander überführt werden sollen, variieren der Kontext und damit die Information. Eine fachliche Beschreibung bietet den Vorteil, dass unterschiedliche Modelle auch transitiv miteinander verbunden sein können, indem der semantische Zusammenhang zwischen Modellelementen explizit beschrieben wird. Eine weitere Verbesserung bzw. Präzisierung kann darüber hinaus erreicht werden, wenn Elemente der Anwendungsdomäne bei der Beschreibung der Beziehungen berücksichtigt werden.

Eng mit diesen Ergebnissen verknüpft sind auch die Erkenntnisse der letzten Forschungsfrage bzgl. der Ausgestaltung notwendiger Komponenten. Eine durchgängige Informationsmodellierung im Sinne einer transparenten Informationsübertragung zwischen Modellen kann nur dann gelingen, wenn sämtliche dafür verwendete Komponenten modellbasiert entwickelt und umgesetzt werden. Zum einen bedeutet dies, dass alle verwendeten Editoren, mit deren Hilfe Modelle erstellt werden, ein oder mehrere valide Metamodelle unterstützen müssen und zum anderen, dass ein alle Artefakte integrierendes Verfahren existieren muss, welches alle notwendigen Schritte mit Hilfe von modellbasierten Aktivitäten umsetzt. Ein solches Verfahren muss dann integraler Bestandteil einer notwendigen Modellierungsmethode sein, die ein Metamodell beinhaltet, das eine Verknüpfung von Modellen ermöglicht. Eine hinreichend präzise Servicebeschreibung wird dann als eine über die-

ses Metamodell verbundene Menge von Teilmodellen realisiert, bei der jedes Teilmodell eine Perspektive auf einen Service beschreibt und durch eine auf fachlicher Basis beschriebene Verbindung zwischen den Perspektiven ein Gesamtbild entsteht. Letztlich kann eine modellbasierte, ganzheitliche Servicebeschreibung nur mit Hilfe eines modellgetriebenen Ansatzes erfolgen, der alle Vorteile einer zumindest semistrukturierten Beschreibung nutzt und die Strukturinformationen für einen teilautomatisierten Informationsübergang verwendet.

Abschließend soll ebenfalls die Erreichung der Erkenntnis- und Gestaltungsziele aus Tabelle 1-1 bewertet werden. Im Sinne der modellgetriebenen Servicebeschreibung als Rahmenwerk zur Arbeit mit Servicemodellen zu deren Operationalisierung kann festgestellt werden, dass die Realisierung umfänglich erfolgt ist. Der enthaltene Servicemodellierungsansatz ist nicht auf eine bestimmte Anwendungsdomäne beschränkt, sondern ermöglicht eine freie Modellierung von Services aus unterschiedlichen Perspektiven und verknüpft diese zu einem Gesamtbild (Erkenntnisziel). Die Umsetzung des Modellierungsansatzes erfolgt modellbasiert und nutzt einen Prototyp, der mit Hilfe modellgetriebener Verfahren zumindest teilweise automatisiert generiert werden konnte (Erkenntnisziel). Das Gestaltungsziel, die Anwendung der Konzepte und des Prototyps in der Logistik, ist nur bedingt erreicht worden, da die Anwendung im Wesentlichen nur mit Hilfe von Beispielen und Szenarien erprobt werden konnte. Allerdings ist darauf aufbauend davon auszugehen, dass einer grundsätzlichen Anwendbarkeit in einem größeren Rahmen nichts entgegensteht. Die prototypische Realisierung verhindert zugleich jedoch eine solche Anwendung, da der bestehende Prototyp nur einen Proof-of-Concept Status besitzt und für eine intensivere Nutzung vor allem die Anwendungsfreundlichkeit verbessert sowie eine verbesserte Fehlerbehandlung implementiert werden muss.

10.3 Schlussbetrachtung

Die Gutachten zu im Laufe der Dissertationen eingereichten Konferenz- und Journalbeiträgen sowie die Zahl neuer, integrativer Ansätze auf diesem Gebiet belegen die Notwendigkeit eines integrativen Ansatzes bzw. das Fehlen wichtiger Schlüsselemente in bestehenden Ansätzen. Aktuelle Arbeiten mit integrativem Charakter, zumeist im Kontext der Unternehmensmodellierung, verwenden einen dafür typischen, Matrix-haften Aufbau, über den Elemente unterschiedlicher Dimensionen miteinander kombiniert werden können. Für jede Zelle einer solchen Matrix werden unterschiedliche Anforderungen oder Profile abgeleitet, die als Lösungsmuster verwendet werden sollen. Lösungsmuster im Sinne einer Empfehlung der zu modellierenden Aspekte bzw. Vorschläge für wiederkehrende Prob-

lemstellungen sind aus Sicht des Autors durchweg positiv zu bewerten und zugleich ein Kritikpunkt der eigenen Arbeit. Allerdings ist die verfügbare Bandbreite an Problemstellungen nicht umfangreich genug gewesen, um stabile Muster ableiten zu können. Eine Angabe rein auf Basis des Anwendungsbeispiels Logistik würde die generelle Anwendbarkeit tendenziell einschränken, denn verbessern. Die bei den verwandten Ansätzen darüber hinaus gehenden Empfehlungen oder Forderungen für den Einsatz spezifischer Modellierungssprachen werden ebenfalls kritisch bewertet. Dies führt zu einer zu starken Einschränkung der Nutzer bzw. in manchen Fällen zu einem sehr hohen Aufwand bei der Modellierung und anschließenden Transformation, falls etwa zu jeder Matrixzelle eine Sprache sowie Transformationen zu umliegenden Zellen im Ansatz definiert werden.

Eine Abwägung zwischen Strukturiertheit und Anwendbarkeit sollte bei der Gestaltung eines multiperspektivischen Ansatzes zur Servicemodellierung unbedingt enthalten sein. Darin liegt die Stärke des hier beschriebenen Ansatzes: Möglichst leicht einzuhaltende Anforderungen, wenige Einschränkungen und die Nutzung des bereits beim Anwender vorhandenen Wissens zur Modellierung. Im Falle des Domänenwissens ist bereits vorhandenes Wissen der Anwender unabdingbar. Wie bereits in 5.3 beschrieben besitzt das SMF keine direkte Verbindung zu einer Anwendungsdomäne bzw. enthält kein Domänenmodell, welches in die Modellierung des Servicemetamodells einbezogen werden könnte. Im Rahmen einer Erweiterung müsste dieser Punkt aufgegriffen und eine Methode entwickelt werden, wie ein unabhängiges Domänenmodell in die Modellierung direkt einbezogen werden könnte. Es muss ein Weg gefunden werden, wie Elemente eines Domänenmodells die Beziehungen zwischen Elementen unterschiedlicher Servicemodelle näher definieren, ohne dass das Domänenmodell selbst zum Bestandteil des Basismetamodells wird. Dann ist es auch möglich, das SMF im Kontext unterschiedlicher Anwendungsdomänen einzusetzen und die grundsätzliche Anwendbarkeit sicherzustellen. Ebenfalls sollte in einer Erweiterung ein spezifisches Nutzerkonzept bzw. Rollenmodell eingearbeitet werden, mit dessen Hilfe Verantwortlichkeiten bei der Modellierung definiert und die entstehenden Ergebnisse hinsichtlich des Nutzens besser bewertet werden können. Außerdem könnte mit Hilfe eines Rollenmodells eine Strukturierung der Abläufe erfolgen, ohne zusätzliche Einschränkungen definieren zu müssen. Dies könnte bspw. hilfreich sein, um eine gewisse Abfolge bei der Modellierung einzelner Perspektiven zu etablieren, die dafür sorgt, dass die zu integrierenden Modelle tatsächlich aufeinander aufbauen können.

Ein weiterer, interessanter Gesichtspunkt ergibt sich auch bei der Betrachtung der Gewichtung von Modellierung und Services im Kontext der Arbeit. Aufgrund des starken Fokus auf Modelle und Modelleigenschaften könnte geprüft werden, ob durch Austausch des Ba-

sismetamodells eine generelle Übertragbarkeit auf andere Problemstellungen möglich ist, etwa bei Transformationsansätzen. Dadurch würde sich die grundlegende Anwendbarkeit des hier vorgestellten Ansatzes noch weiter erhöhen. Dem Basismetamodell käme die Aufgabe zu, grundlegende Eigenheiten einer zu modellierenden Domäne zu strukturieren und bislang evtl. nur implizite Beobachtungen zu externalisieren.

Abschließend sollte gesagt werden, dass auch mit dem SMF keine endgültige Antwort auf die Frage nach der Natur eines Service gegeben werden kann. Die Servitisierung, d.h. die Umwandlung von Prozessen oder technischer Systeme sowie deren Angebot in Form von Services ist noch immer nicht abgeschlossen. Gerade neuartige Services wie location-based Services oder die Möglichkeiten, die sich aus dem „Internet der Dinge“ ergeben, werden die Art und Weise wie Services definiert werden noch einmal verändern. Die Wissenschaft, insbesondere die Wirtschaftsinformatik mit ihrer starken Verzahnung zur Praxis, muss Unternehmen mit effizienten, aber vor allem auch einfach einzusetzenden, Methoden und Werkzeugen für die weitere Entwicklung im Dienstleistungssektor unterstützen und auch eigenständig neue Wege der Serviceorientierung definieren und beschreiten.

Literaturverzeichnis

- Adam, D. (1997). Planung und Entscheidung Modelle - Ziele - Methoden ; mit Fallstudien und Lösungen. Wiesbaden, Gabler.
- Akkermans, H., Baida, Z., Gordijn, J., Pena, N., Altuna, A. und Laresgoiti, I. (2004). "Value Webs: Using Ontologies to Bundle Real-World Services." IEEE Intelligent Systems **19**(4): 57-66.
- Alarcon, R., Wilde, E. und Bellido, J. (2010). Hypermedia-Driven RESTful Service Composition. Service-Oriented Computing. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 111-120.
- Algermissen, L. Prozessorientierte Verwaltungsmodernisierung Gestaltung der Prozesslandschaft in öffentlichen Verwaltungen mit der PICTURE-Methode.
- Alter, S. (2008). "Service System fundamentals: Work system, value chain, and life cycle." IBM Systems Journal **47**(1): 71-85.
- Alter, S. (2010). "Integrating Sociotechnical and Technical Views of e-Services." e-Service Journal **7**(1): 15-42.
- Amrani, M., Dingel, J., Lambers, L., Lucio, L., Salay, R., Selim, G., Syriani, E. und Wimmer, M. (2012). Towards a model transformation intent catalog. Proceedings of the First Workshop on the Analysis of Model Transformations (AMT 2012). Innsbruck, Austria, ACM: 3-8.
- Arnold, D., Isermann, H., Kuhn, A. und Tempelmeier, H. (2008). Handbuch Logistik. Berlin [u.a.], Springer.
- Arns, M., Fischer, M., Kemper, P. und Tepper, C. (2002). "Supply Chain Modelling and Its Analytical Evaluation." The Journal of the Operational Research Society **53**(8): 885-894.
- Augenstein, C. und Ludwig, A. (2013a). Interconnected service models - Emergence of a comprehensive logistics service model. 17th IEEE Enterprise Distributed Object Computing Conference Workshops (EDOCW 2013), Vancouver, IEEE.
- Augenstein, C. und Ludwig, A. (2013b). The Service Meta Modeling Editor - Bottom-up integration of service models. 8th International Conference DESRIST 2013 (DESRIST 2013), Helsinki, Finnland, Springer Verlag Berlin Heidelberg.
- Augenstein, C., Ludwig, A. und Franczyk, B. (2012). Integration of service models – preliminary results for consistent logistics service management. SRII Global Conference. San Jose, USA, IEEE: 100-109.
- Augenstein, C., Müller, H. und Franczyk, B. (2010). Developing a unified service model for collaborative modeling of logistics services. AITM. Wroclaw, Poland.

- Augenstein, C., Mutke, S. und Ludwig, A. (2013). Integration von Planungssystemen in der Logistik - Ansatz und Anwendung. 11th International Conference on Wirtschaftsinformatik (WI 2013), Leipzig.
- Baacke, L. M., Tobias; Rohner, Peter (2009). Component-Based Process Modelling in Health Care. 17th European Conference on Information Systems (ECIS), Verona, Italy.
- Baida, Z., Gordijn, J., Akkermans, H., Morch, A. Z. und Saele, H. (2004a). Ontology-based analysis of e-service bundles for networked enterprises. 17th Bled eCommerce Conference (Bled 2004). Bled, Slovenia.
- Baida, Z., Gordijn, J. und Omelayenko, B. (2004b). A Shared Service Terminology for Online Service Provisioning. ICEC'04, Sixth International Conference on Electronic Commerce. ACM.
- Bamford, C. (2010). The changing supply of logistics services - a UK perspective. Global logistics : new directions in supply chain management. Waters, C. D. J. London, Philadelphia, Kogan Page.
- Barakat, L., Miles, S. und Luck, M. (2012). Reactive Service Selection in Dynamic Service Environments. Service-Oriented and Cloud Computing. Paoli, F. D., Pimentel, E. und Zavattaro, G. E. Bertinoro, Springer Verlag: 17-31.
- Bartsch, C. (2010). Modellierung und Simulation von IT-Dienstleistungsprozessen. Fakultät für Wirtschaftswissenschaften. Karlsruhe, Universität Karlsruhe. **Doktor:** 351.
- Baumann, S. (2008). Projektfinanzierung in Supply Chain Netzwerken : eine neo-institutionenökonomische Analyse am Beispiel der Automobilindustrie. Wiesbaden, Gabler.
- Becker, J. und Delfmann, P. (2004). Referenzmodellierung Grundlagen, Techniken und domänenbezogene Anwendung. Heidelberg ;s.l., Physica-Verlag HD.
- Becker, J., Delfmann, P., Knackstedt, R. und Kuropa, D. (2002). Konfigurative Referenzmodellierung. Wissensmanagement mit Referenzmodellen. Becker, J. Berlin, Physica Verlag: 25-144.
- Becker, J., Holten, R., Knackstedt, R. und Niehaves, B. (2003). Forschungsmethodische Positionierung in der Wirtschaftsinformatik – epistemologische, ontologische und linguistische Leitfragen – Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik.
- Becker, J., Knackstedt, R., Holten, R., Hansmann, H. und Neumann, S. (2001). Konstruktion von Methodiken: Vorschläge für eine begriffliche Grundlegung und domänenspezifische Anwendungsbeispiele. Arbeitsberichte des Instituts für Wirtschaftsinformatik. Münster, Westfälische Wilhelms-Universität Münster. **77**.
- Becker, J., Krömer, H. und Niehaves, B. (2009). Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Heidelberg, Physica-Verl.

- Becker, J., Niehaves, B. und Knackstedt, R. (2004). Bezugsrahmen zur epistemologischen Positionierung der Referenzmodellierung. Referenzmodellierung Grundlagen, Techniken und domänenbezogene Anwendung. Becker, J. und Delfmann, P. Heidelberg, Physica-Verlag: 1-17.
- Becker, J., Niehaves, B., Pöppelbuß, J., Ortbach, K., Plattfaut, R., Voigt, M. und Malsbender, A. (2015). Service Design - Mit der Quadromo-Methode von der Idee zum Konzept. Berlin, Heidelberg, Gabler Verlag.
- Becker, J. und Pfeiffer, D. (2006). Beziehungen zwischen behavioristischer und konstruktionsorientierter Forschung in der Wirtschaftsinformatik. Fortschritt in den Wirtschaftswissenschaften : wissenschaftstheoretische Grundlagen und exemplarische Anwendungen. Zelewski, S. Wiesbaden, Dt. Univ.-Verl.
- Becker, J., Pfeiffer, D. und Räckers, M. (2007). Domain Specific Process Modelling in Public Administrations – The PICTURE-Approach. Electronic Government. Wimmer, M., Scholl, J. und Grönlund, Å., Springer Berlin Heidelberg. **4656**: 68-79.
- Becker, J., Probandt, W. und Vering, O. (2012). Grundsätze ordnungsmäßiger Modellierung Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement. Berlin [u.a.], Springer Gabler.
- Becker, J., Rosemann, M. und Schütte, R. (1995). "Grundsätze ordnungsmäßiger Modellierung." Wirtschaftsinformatik **37**(5): 435-445.
- Becker, J. und Schütte, R. (2004). Handelsinformationssysteme : domänenorientierte Einführung in die Wirtschaftsinformatik. Frankfurt am Main, Redline Wirtschaft.
- Becker, J. K., Ralf; Pfeiffer, Daniel; Janiesch, Christian (2007). Configurative method engineering - on the applicability of reference modeling mechanisms in method engineering. Americas Conference on Information Systems (AMCIS), Keystone.
- Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M. und Mecella, M. (2003). A Foundational Vision of e-Services. Proc. of the CAiSE 2003 Workshop on Web Services, e-Business, and the Semantic Web (WES 2003).
- Berre, A. J., Lew, Y., Elvesaeter, B. und de Man, H. (2013). Service Innovation and Service Realisation with VDML and ServiceML. 17th IEEE International Enterprise Distributed Object Conference Workshops (EDOCW 2013), Vancouver, IEEE.
- Bézivin, J. (2004). "In search of a basic principle for model-driven engineering." European Journal for the Informatics Professional **5**(2): 21-24.
- Bézivin, J. (2006). Model driven engineering: an emerging technical space. Lecture Notes in Computer Science (GTTSE 2005). Lämmel, R., Saraiva, J. und Visser, J. Berlin, Heidelberg, Springer. **4143**: 36-64.
- Bianchini, D., De Antonellis, V., Pernici, B. und Plebani, P. (2006). "Ontology-based methodology for e-service discovery." Information Systems **31**: 361-380.

- Bitner, M. J., Brown, S. W. und Meuter, M. L. (2000). "Technology Infusion in Service Encounters." Journal of the Academy of Marketing Science **28**(1): 138-149.
- Bitsaki, M., Danylevych, O., van den Heuvel, W.-J., Koutras, G., Leymann, F., Manciacchi, M., Nikolaou, C. und Papazoglou, M. (2009). Model Transformations to Leverage Service Networks. LNCS (Service-Oriented Computing – ICSOC 2008 Workshops). Feuerlicht, G. und Lamersdorf, W. Berlin / Heidelberg, Springer **5472**: 103-117.
- Björkqvist, M., Spicuglia, S., Chen, L. und Binder, W. (2013). QoS-Aware Service VM Provisioning in Clouds: Experiences, Models, and Cost Analysis. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 69-83.
- Blake, M. B. (2007). "Decomposing Composition: Service-Oriented Software Engineers." IEEE Software **24**(6): 68-77.
- Blanchard, B. S. (2008). Logistics as an Integrating System's Function. Logistics Engineering Handbook. Taylor, D. Boca Raton, CRC Press.
- Blessing, D. und Fleisch, E. (2000). Metamodelle. Arbeitsbericht des Institut für Wirtschaftsinformatik der Universität St. Gallen. St. Gallen, Institut für Wirtschaftsinformatik, Universität St. Gallen.
- Böhmman, T., Winkler, T., Fogl, F. und Krcmar, H. (2004). Servicedatenmanagement für IT-Dienstleistungen: Ansatzpunkte für ein fachkonzeptionelles Referenzmodell. Referenzmodellierung - Grundlagen, Techniken und domänenbezogene Anwendung. Becker, J. und Delfmann, P., Physika-Verlag: 99-123.
- Boppert, J. und Rinza, T. (2007). Logistik im Zeichen zunehmender Entropie. Neue Wege in der Autommobillistik. Günther, W. A. Berlin, Springer: 17-28.
- Bossert, J. M. und Willems, S. P. (2007). "A Periodic-Review Modeling Approach for Guaranteed Service Supply Chains." Interfaces **37**(5): 420-435.
- Böttcher, M. (2009). Architektur integrierter Dienstleistungssysteme - Konzeption, Metamodell und technikraumspezifische Konkretisierung. Fakultät für Mathematik und Informatik. Leipzig, Universität Leipzig. **Doktor**: 207.
- Bretzke, W.-R. (2008). Logistische Netzwerke. Berlin [u.a.], Springer.
- Brocke, H. U., Falk; Brenner, Walter (2010). Reuse-Mechanisms for Mass Customizing IT-Service Agreements. 16th Americas Conference on Information Systems (AMCIS), Lima.
- Bruhn, M. und Stauss, B. (2010). Serviceorientierung im Unternehmen Forum Dienstleistungsmanagement. Wiesbaden, Gabler Verlag / GWV Fachverlage GmbH, Wiesbaden.
- Buchwald, S., Bauer, T. und Reichert, M. (2010). Durchgängige Modellierung von Geschäftsprozessen in einer Service-orientierten Architektur. GI Tagung Modellierung - 2010, Klagenfurt, GI.

- Buchwald, S., Tiedeken, J., Bauer, T. und Reichert, M. (2010). Anforderungen an ein Metamodell für SOA-Repositories. 2nd Central-European Workshop on Services and their Composition (ZEUS'10). Berlin: 25-32.
- Bullinger, H.-J., Fähnrich, K.-P. und Meiren, T. (2003). "Service Engineering - methodical development of new service products." International Journal of Production Economics **85**: 275-287.
- Bullinger, H.-J. und Scheer, A.-W., Eds. (2006). Service engineering : Entwicklung und Gestaltung innovativer Dienstleistungen ; mit 24 Tabellen. Berlin [u.a.], Springer.
- Calegari, D. und Szasz, N. (2013). "Verification of Model Transformations: A Survey of the State-of-the-Art." Electronic Notes in Theoretical Computer Science **292**(0): 5-25.
- Cardoso, J., Barros, A., May, N. und Kylau, U. (2010). Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments. 2010 IEEE International Conference on Services Computing. Miami, Florida. **0**: 602-609.
- Cardoso, J., Pedrinaci, C., Leidig, T., Rupino, P. und Leenheer, P. D. (2012). Open semantic service networks. Proceedings of the 4th International Symposium on Services Science Leipzig (Germany). Meyer, K. und Abdelkafi, N. Leipzig.
- Cardoso, J., Voigt, K. und Winkler, M. (2009). Service Engineering for the Internet of Services. ICEIS 2008, LNBIP 19. Filipe, J. und Cordeiro, J.: 15-27.
- Chesbrough, H. und Spohrer, J. (2006). "A research Manifesto for Service Science." Communications of the ACM **49**(7): 33-40.
- Chituc, C.-M. (2010a). Introduction to the First International Workshop on Performance Assessment and Auditing in Service Computing (PAASC 2010). Service-Oriented Computing. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 1-3.
- Chituc, C.-M. (2010b). Towards Assessing Performance in Service Computing. Service-Oriented Computing. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 51-61.
- Coenen, C. (2001). Serviceorientierung und Servicekompetenz von Kundenkontaktmitarbeitern. Jahrbuch Dienstleistungsmanagement. Bruhn, M. und Stauss, B. Wiesbaden, Gabler: 341-374.
- Cunningham, L. F., Young, C. E., Ulaga, W. und Lee, M. (2004). "Consumer views of service classifications in the USA and France." Journal of Services Marketing **18**(6): 421-432.
- Czarnecki, K. und Helsen, S. (2006). "Feature-based survey of model transformation approaches." IBM Systems Journal **45**(3): 621-645.

- Dasgupta, G. B., Shrinivasan, Y., Nayak, T. K. und Nallacherry, J. (2013). Optimal Strategy for Proactive Service Delivery Management Using Inter-KPI Influence Relationships. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 131-145.
- Del Fabro, M. D., Bezivin, J., Jouault, F. und Valduriez, P. (2005). Applying Generic Model Management to Data Mapping. 21èmes Journées Bases de Données Avancées (BDA 2005), Saint Malo, Frankreich.
- Delfmann, P. (2006). Adaptive Referenzmodellierung: Methodische Konzepte zur Konstruktion und Anwendung wiederverwendungsorientierter Informationsmodelle. Berlin, Logos-Verl.
- Demchenko, Y., Ngo, C., Martínez-Julia, P., Torroglosa, E., Grammatikou, M., Jofre, J., Gheorghiu, S., Garcia-Espin, J. A., Perez-Morales, A. D. und Laat, C. d. (2012). GEMBus Based Services Composition Platform for Cloud PaaS. Service-Oriented and Cloud Computing. Paoli, F. D., Pimentel, E. und Zavattaro, G. E. Bertinoro, Springer Verlag: 32-47.
- Demirkan, H., Kauffman, R. J., Vayghan, J. A., Fill, H.-G., Karagiannis, D. und Maglio, P. P. (2008). "Service-oriented technology and management: Perspectives on research and practice for the coming decade." Journal of Electronic Commerce Research and Applications(7): 356-376.
- Dijkstra, E. W. (1976). A Discipline of Programming. Englewood Cliffs, NJ, Prentice-Hall.
- Domschke, W., Scholl, A. und Domschke, S. (2003). Grundlagen der Betriebswirtschaftslehre : eine Einführung aus entscheidungsorientierter Sicht ; mit 80 Tabellen. Berlin [u.a.], Springer.
- Dorne, R., Lesaint, D., Owusu, G. und Voudouris, C. (2008). Service Chain Management: Technology Innovation for the Service Business. Berlin, Heidelberg, Springer-Verlag.
- Dumas, M., O'Sullivan, J., Heravizadeh, M., Edmond, D. und ter Hofstede, A. (2003). Towards a Semantic Framework for Service Description. Semantic Issues in E-Commerce Systems. Meersman, R., Aberer, K. und Dillon, T., Springer US. **111**: 277-291.
- Edvardsson, B., Gustafsson, A. und Roos, I. (2005). "Service portraits in service research: a critical review." International Journal of Service Industry Management **16**(1): 107-121.
- Erl, T. (2005). Service-oriented architecture : concepts, technology, and design. Upper Saddle River, NJ, Prentice Hall Professional Technical Reference.
- Erl, T. (2008). SOA: Principles of Service Design, Prentice Hall PTR.
- Erradi, A., Anand, S. und Kulkarni, N. (2006a). SOAF: An Architectural Framework for Service Definition and Realization. IEEE International Conference on Services Computing (SCC'06).

- Erradi, A., Anand, S. und Kulkarni, N. (2006b). SOAF: An Architectural Framework, for Service Definition and Realization. IEEE International Conference on Services Computing (SCC 06). Chicago, Illinois, USA, IEEE Computer Society: 151-158.
- Esswein, W. L., Sina; Stark, Jeannette (2010). The Potential of Reference Modeling for Simulating Mobile Construction Machinery. Lecture Notes in Business Information Processing. **43**: 683-694.
- Fährlich, K.-P. und Opitz, M. (2006). Service Engineering - Entwicklungspfad und Bild einer jungen Disziplin. Service engineering : Entwicklung und Gestaltung innovativer Dienstleistungen ; mit 24 Tabellen. Bullinger, H.-J. und Scheer, A.-W. Berlin, Springer: 85-112.
- Faravelon, A. e., Chollet, S., Verdier, C. und Front, A. (2012). Configuring Private Data Management as Access Restrictions: From Design to Enforcement. Service-Oriented Computing. Liu, C., Ludwig, H., Toumani, F. und Yu, Q. E. Shanghai, Springer Verlag: 344-358.
- Favre, J.-M. und Nguyen, T. (2005). "Towards a Megamodel to Model Software Evolution Through Transformations." Electronic Notes in Theoretical Computer Science **127**(3): 59-74.
- Fensel, D. (2003). Ontologies : a silver bullet for knowledge management and electronic commerce. Berlin ; New York, Springer-Verlag.
- Fernández, H., Tedeschi, C. und Priol, T. (2012). Decentralized Workflow Coordination through Molecular Composition. Service-Oriented Computing – ICSOC 2011 Workshops. Pallis, G., Jmaiel, M., Graupner, S. et al. Paphos, Springer Verlag: 22-32.
- Ferrario, R., Guarino, N., Janiesch, C., Kiemens, T., Oberle, D. und Probst, F. (2010). Towards an Ontological Foundation of Services Science: The General Service Model. 10th International Conference on Wirtschaftsinformatik (WI 2010). Zurich, Switzerland: 675-684.
- Ferstl, O. K. und Sinz, E. J. (2001). Grundlagen der Wirtschaftsinformatik Bd. 1 [...]. München [u.a.], Oldenbourg.
- Fettke, P. und Loos, P. (2002). Methoden zur Wiederverwendung von Referenzmodellen - Übersicht und Taxonomie. Referenzmodellierung. Becker, J. K., Ralf (Hrsg.). Münster, Universität Münster: 9-34.
- Fiege, R. und Stelzer, D. (2007). Analyse des Beitrages von Axiomatic Design zum Entwurf Serviceorientierter Architekturen. 8. Internationale Tagung Wirtschaftsinformatik 2007, Karlsruhe, Universitätsverlag Karlsruhe.
- Fischer, J., Dangelmaier, W., Nastansky, L. und Suhl, L. (2012). Bausteine der Wirtschaftsinformatik Grundlagen und Anwendungen. Berlin, Schmidt.
- Fitzsimmons, J. A. und Fitzsimmons, M. J. (2001). Service Management: Operations, Strategy, and Information Technology. New York, McGraw-Hill.

- Fortmann, K.-M. und Kallweit, A. (2007). Logistik. Stuttgart, Kohlhammer.
- Frank, U., Klein, S., Krcmar, H. und Teubner, A. (1998). Aktionsforschung in der WI - Einsatzpotentiale und -probleme. Wirtschaftsinformatik und Wissenschaftstheorie. Grundpositionen und Theoriekerne - Arbeitsbereiche des Instituts für Produktion und industrielles Informationsmanagement Nr. 4. Schütte, R., Siedentopf, R. und Zelewski, S. Essen, Institut für Produktion und industrielles Informationsmanagement. **4**: 71-90.
- Frank, U. und van Laak, B. L. (2003). Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsberichte des Instituts für Wirtschaftsinformatik. Koblenz, Instituts für Wirtschaftsinformatik, Universität Koblenz / Landau. **34**.
- Garschhammer, M., Hauck, R., Hegering, H.-G., Langer, M., Nerb, M., Kempter, B., Radisic, I., Rölle, H. und Schmidt, H. (2001). Towards generic service management concepts - A service model based approach. 7th IFIP/IEEE International Symposium on Integrated Network Management. Seattle, Washington, USA, IEEE Computer Society: 719-732.
- Gehlert, A. (2007). Migration fachkonzeptueller Modelle. Berlin, Logos-Verl.
- Ghezzi, C., Pezzé, M. und Tamburrelli, G. (2013). Improving Interaction with Services via Probabilistic Piggybacking. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 39-53.
- Glöckner, M., Augenstein, C. und Ludwig, A. (2014). Metamodel of a Logistics Service Map. 17th International Conference on Business Information Systems (BIS 2014), Larnaca, Cyprus, Springer.
- Glushko, R. J. (2008). "Designing a service science discipline with discipline." IBM Systems Journal **47**(1): 15-27.
- Goeken, M. (2003). Grundlagen und Ansätze einer Referenzmodellierung für Führungsinformationssysteme, Institut für Wirtschaftsinformatik.
- Gómez-Pérez, A., Fernández-López, M. und Corcho, O. (2004). Ontological engineering : with examples from the areas of knowledge management, e-commerce and the Semantic Web / Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. London ; New York, Springer.
- Göpfert, I. (2009). Logistik der Zukunft = Logistics for the future. Wiesbaden, Gabler.
- Greiffenberg, S. (2004). Methodenentwicklung in Wirtschaft und Verwaltung. Hamburg, Kovač.
- Gronau, N., Bahrs, J., Hake, M., Heinze, P., Lembcke, R., Scharff, C. und Vladova, G. (2010). Wissensorientierte Modellierung im Lebenszyklus von Dienstleistungen. Dienstleistungsmodellierung 2010. Thomas, O. und Nüttgens, M. Berlin, Physica-Verl.: 3-23.

- Grönroos, C. (2000). Service Management and Marketing. A Customer Relationship Management Approach. Chichester u.a., John Wiley & Sons.
- Gruhn, V., Pieper, D. und Röttgers, C. (2006). MDA Effektives Software-Engineering mit UML 2 und Eclipse. Berlin, Heidelberg, Springer-Verlag Berlin Heidelberg.
- Gu, Q. und Lago, P. (2007). A stakeholder-driven service life cycle model for SOA. 2nd international workshop on Service oriented software engineering: in conjunction with the 6th ESEC/FSE joint meeting. Dubrovnik, Croatia, ACM: 1-7.
- Gudehus, T. (2010). Logistik : Grundlagen - Strategien - Anwendungen. Berlin [u.a.], Springer.
- Gustafsson, A. und Johnson, M. (2003). Competing in the Service Economy. San Francisco, CA, Jossey-Bass.
- Hallerbach, A., Bauer, T. und Reichert, M. (2008). "Anforderungen an die Modellierung und Ausführung von Prozessvarianten." Datenbank Spektrum 24: 48-58.
- Hallerbach, A. B., Thomas; Reichert, Manfred (2010). Configuration and Management of Process Variants. Handbook on Business Process Management. vom Brocke, J. R., Michael. Heidelberg, New York, Dordrecht, London, Springer Verlag: 237-255.
- Hars, A. (1994). Referenzdatenmodelle: Grundlagen effizienter Datenmodellierung. Institut für Wirtschaftsinformatik. Wiesbaden, Universität Saarbrücken. **Doktor**.
- Hastings, H. und Saperstein, J. (2014). Service Thinking. New York, Business Expert Press.
- Herrmann, H.-J. (1992). Modellgestützte Planung in Unternehmen Entwicklung eines Rahmenkonzepts. Wiesbaden ;s.l., Gabler Verlag.
- Hesse, W. und Krzensk, B. (2004). Ontologien in der Softwaretechnik. Modellierung 2004. Marburg, GI.
- Hesse, W. und von Braun, H. (2001). Wo kommen die Objekte her? Ontologisch-erkenntnistheoretische Zugänge zum Objektbegriff. Informatik 2001 (Jahrestagung der Gesellschaft für Informatik e.V.)
- Heutschi, R. (2007). Serviceorientierte Architektur : Architekturprinzipien und Umsetzung in die Praxis ; mit ... 52 Tabellen. Berlin [u.a.], Springer.
- Hill, T. P. (1977). "On Goods and Services." Review of Income and Wealth 23(4): 315-338.
- Hofreiter, B. H., Christian; Kappel, Gerti; Mayrhofer, Dieter; vom Brocke, Jan (2012). Inter-organizational Reference Models - May Inter-organizational Systems Profit from Reference Modeling? Lecture Notes in Computer Science. Berlin, Springer. **7350**: 32-47.
- Hradilak, K. P. (2011). Führen von IT-Service-Unternehmen. Wiesbaden, Vieweg +Teubner Verlag.

- Hull, R. (2013). Data-Centricity and Services Interoperation. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 1-8.
- Ivanović, D., Carro, M. und Hermenegildo, M. V. (2012). A Constraint-Based Approach to Quality Assurance in Service Choreographies. Service-Oriented Computing. Liu, C., Ludwig, H., Toumani, F. und Yu, Q. E. Shanghai, Springer Verlag: 252-267.
- Jetzke, S. (2007). Grundlagen der modernen Logistik : Methoden und Lösungen. München, Hanser.
- Jones, S. (2005). "Toward an Acceptable Definition of Service." IEEE Software **22**(3): 87-93.
- Jongmans, S.-S. T. Q., Santini, F., Sargolzaei, M., Arbab, F. und Afsarmanesh, H. (2012). Automatic Code Generation for the Orchestration of Web Services with Reo. Service-Oriented and Cloud Computing. Paoli, F. D., Pimentel, E. und Zavattaro, G. E. Bertinoro, Springer Verlag: 1-16.
- Jouault, F. und Kurtev, I. (2007). "On the interoperability of model-to-model transformation languages." Science of Computer Programming **68**(3): 114-137.
- Juszczak, L., Schall, D., Mietzner, R., Dustdar, S. und Leymann, F. (2010). CAGE: Customizable Large-Scale SOA Testbeds in the Cloud*. Service-Oriented Computing. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 76-87.
- Karagiannis, D. und Kühn, H. (2002). Metamodelling Platforms. E-Commerce and Web Technologies. Bauknecht, K., Tjoa, A. und Quirchmayr, G. Berlin, Springer Berlin / Heidelberg. **2455**: 451-464.
- Karni, R. und Kaner, M. (2006). An engineering tool for the conceptual design of service systems. Advances in Services Innovations. Spath, D. und Fähnrich, K.-P. Berlin, Springer: 65-84.
- Karow, M. P., Daniel; Räckers, Michael (2008). Empirical-Based Construction of Reference Models in Public Administrations. Multikonferenz Wirtschaftsinformatik, München, GITO-Verlag.
- Kattepur, A., Georgantas, N. und Issarny, V. (2013). QoS Analysis in Heterogeneous Choreography Interactions. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 23-38.
- Kent, S. (2002). Model Driven Engineering. Integrated Formal Methods. Butler, M., Petre, L. und Sere, K. Berlin, Springer. **2335**: 286-298.
- Kern, H., Böttcher, M., Kühne, S. und Meyer, K. (2009). Ansatz zur ganzheitlichen Erstellung und Verarbeitung von Dienstleistungsmodellen. Dienstleistungsmodellierung – Methoden, Werkzeuge und Branchenlösungen. Thomas, O. und Nüttgens, M. Heidelberg, Physica Verlag: 3-16.

- Kett, H., Voigt, K., Scheithauer, G. und Cardoso, J. (2008). Service Engineering in Business Ecosystems. XVIII International RESER Conference, Stuttgart, Fraunhofer IRB.
- Klatt, B., Brosch, F., Durdik, Z. und Rathfelder, C. (2012). Quality Prediction in Service Composition Frameworks. Service-Oriented Computing – ICSOC 2011 Workshops. Pallis, G., Jmaiel, M., Graupner, S. et al. Paphos, Springer Verlag: 131-146.
- Kleine-Kleffmann, M. und Böber, S. (2006). Struktur des Logistik-Dienstleistungsmarktes, Fachbereich Betriebswirtschaft. Studiengang Wirtschaftsinformatik.
- Kleiner, F. (2013). IT Service Management. Wiesbaden, Springer Vieweg Fachmedien.
- Kleppe, A. G., Warmer, J. und Bast, W. (2003). MDA Explained: The Model Driven Architecture: Practice and Promise, Addison-Wesley Longman Publishing Co., Inc.
- Klischewski, R. (2000). Abstrakte Bedürfnisse und konkrete Beziehungen oder: Wie man Services (nicht) modelliert. Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik - Proceedings Modellierung 2000. Ebert, J. und Frank, U. H. St. Goar: 19-26.
- Koether, R. (2007). Technische Logistik. München, Hanser.
- Krafzig, D., Banke, K. und Slama, D. (2005a). Enterprise SOA : service-oriented architecture best practices. Indianapolis, IN, Prentice Hall Professional Technical Reference.
- Krafzig, D., Banke, K. und Slama, D. (2005b). Enterprise SOA. Service-oriented Architecture: Best Practices, Prentice Hall PTR.
- Kummer, S. (1992). Logistik im Mittelstand : Stand und Kontextfaktoren der Logistik in mittelständischen Unternehmen. Stuttgart, Poeschel.
- Kummer, S., Grün, O. und Jammernegg, W. (2006). Grundzüge der Beschaffung, Produktion und Logistik. München [u.a.], Pearson Studium.
- Kunau, G., Junginger, M., Herrmann, T. und Krcmar, H. (2005). Ein Referenzmodell für das Service Engineering mit multiperspektivischem Ansatz. Konzepte für das Service Engineering. Herrmann, T., Kleinbeck, U. und Krcmar, H. Heidelberg, Physica-Verlag: 187-216.
- Kunkel, R., Kerkhoff, H., Augenstein, C. und Franczyk, B. (2013). "Generierte Mensch-Maschine-Schnittstellen zur ad-hoc Integration von Logistikketten." HMD Praxis der Wirtschaftsinformatik **50**(249): 76-86.
- Kushida, K. E. und Zysman, J. (2009). "The Services Transformation and Network Policy: The new Logic of Value Creation." Review of Policy Research **26**(1-2): 173-194.
- Kutlu, S. (2007). Fourth Party Logistics: The future of supply chain outsourcing. Brentwood, Best Global Publishing Ltd.

- Lago, P. und Razavian, M. (2012). A Pragmatic Approach for Analysis and Design of Service Inventories. Service-Oriented Computing – ICSOC 2011 Workshops. Pallis, G., Jmaiel, M., Graupner, S. et al. Paphos, Springer Verlag: 44-53.
- Lang, K. (1997). Gestaltung von Geschäftsprozessen mit Referenzprozessbausteinen. Institut für Wirtschaftsinformatik. Wisbaden, Universität Erlangen-Nürnberg. **Dissertation**.
- Langley, J. C. (2009). 2009 third party logistics: The state of logistics outsourcing, Capgemini. **14**.
- Legner, C. und Heutschi, R. (2007). SOA Adoption in Practice - Findings from early SOA Implementations. Proceedings of the 15th European Conference on Information Systems (ECIS), St. Gallen, Switzerland.
- Legner, C. und Vogel, T. (2007). Design Principles for B2B Services – An Evaluation of Two Alternative Service Designs. 3rd IEEE International Conference on Services Computing, Salt Lake City, Utah, IEEE Computer Society.
- Lemke, C. (2012). Auszug aus der Modellierung des IT-Dienstleistungsmodells "proITS" am Beispiel der Struktur von Forschungseinrichtungen und deren IT- Service. Working Papers. Bruche, G., Dörrenbächer, C., Nagel, F. und Ripsas, S. Berlin, IMB Institute of Management Berlin. **65**: 29.
- Lindgreen, A. und Sempels, C. (2001). Competitive Marketing Strategies in the Era of IT-based Interactivity: The Marketing of Services on the Internet. 5th International Conference of the Academy of Business and Administrative Sciences. Brussels, Belgium.
- Lischke, M. (2008). Umsetzung von 4PL-Konzepten in Logistikunternehmen : am Beispiel der Supply Chain der Automobilindustrie. Hamburg, Diplomica Verl.: VII, 75 S.
- Löffler, C. (2011). Service-Externalisierung - Ein wettbewerbsstrategisches Konzept im Management IT-basierter Dienstleistungen. Wiesbaden, Gabler-Verlag.
- López-Sanz, M., Acuña, C. J., Cuesta, C. E. und Marcos, E. (2008). "Modelling of Service-Oriented Architectures with UML." Electronic Notes in Theoretical Computer Science **194**(4): 23-37.
- Lovelock, C. H. (1983). "Classifying Services to Gain Strategic Marketing Insights." Journal of Marketing **47**(3): 9-20.
- Lovelock, C. H. und Gummesson, E. (2004). "Whither Services Marketing." Journal of Service Research **7**(1): 20-41.
- Lusch, R. F., Vargo, S. L. und Wessels, G. (2008). "Toward a conceptual foundation for Service Science: Contributions from a service-dominant logic." IBM Systems Journal **47**(1): 5-14.
- Maglio, P. P., Srinivasan, S., Kreulen, J. T. und Spohrer, J. (2006). "Service systems, service scientists, SSME, and innovation." Communications of the ACM **49**(7): 81-85.

- Mansfeld, K. (2012). Integrierte Gestaltung und Bewertung für strategische Entscheidungen. Leipziger Beiträge zur Wirtschaftsinformatik. Franczyk, B. Leipzig, Institut für Wirtschaftsinformatik, Universität Leipzig. 9.
- March, S. T. und Smith, G. F. (1995). "Design and natural science research on information technology." Decision Support Systems 15(4): 251-266.
- Meiländer, D., Bucchiarone, A., Cappiello, C., Nitto, E. D. und Gorlatch, S. (2012). Using a Lifecycle Model for Developing and Executing Real-Time Online Applications on Clouds. Service-Oriented Computing – ICSOC 2011 Workshops. Pallis, G., Jmaiel, M., Graupner, S. et al. Paphos, Springer Verlag: 33-43.
- Melzer, I. (2010). Service-orientierte Architekturen mit Web Services. Heidelberg, Spektrum Akademischer Verlag.
- Mendling, J., Reijers, H. A. und van der Aalst, W. M. P. (2010). "Seven process modeling guidelines (7PMG)." Information and Software Technology 52(2): 127-136.
- Mens, T. und Van Gorp, P. (2006). "A Taxonomy of Model Transformation." Electronic Notes in Theoretical Computer Science 152(0): 125-142.
- Meyer, K. und Böttcher, M. (2012). Entwicklungspfad Service Engineering 2.0 - Neue Perspektiven für die Dienstleistungsentwicklung, Logos Verlag.
- Meyer, K., Thieme, M., Zinke, C. und Drechsler, N. (2012). Produkt-Dienstleistungs-Lebenszyklus: Methoden und Funktionen für die Entwicklung und das Management produktbegleitender Dienstleistungen. Proceedings of the 4th International Symposium on Services Science Leipzig (Germany), 2012, Leipzig.
- Min, H. und Zhou, G. (2002). "Supply chain modeling: past, present and future." Computers & Industrial Engineering 43(1-2): 231-249.
- Moha, N., Palma, F., Nayrolles, M., Conseil, B. J., Guéheneuc, Y.-G., Baudry, B. und Jézequel, J.-M. (2012). Specification and Detection of SOA Antipatterns. Service-Oriented Computing. Liu, C., Ludwig, H., Toumani, F. und Yu, Q. E. Shanghai, Springer Verlag: 1-16.
- Moustafa, A. und Zhang, M. (2013). Multi-Objective Service Composition Using Reinforcement Learning. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 298-312.
- Mutke, S., Augenstein, C. und Ludwig, A. (2013). Model-based integrated planning for logistics service contracts. 17th IEEE Enterprise Distributed Object Computing Conference Workshops (EDOCW 2013), Vancouver, IEEE.
- Mutke, S., Augenstein, C., Roth, M., Ludwig, A. und Franczyk, B. (2015). "Real-time information acquisition in a model-based integrated planning environment for logistics contracts." Journal of Object Technology 14(1): 2:1-25.
- Nardi, J. C., de Almeida Falbo, R., Almeida, J. P. A., Guizzardi, G., Pires, L. F., Van Sinderen, M. und Guarino, N. (2013). Towards a Commitment-based Reference

- Ontology for Services. 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2013), Vancouver, IEEE.
- Nayak, N. und Nigam, A. (2007). Modeling Business Services for Implementing on Global Business Services Delivery Platforms. 9th IEEE International Conference on E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007). Tokyo, Japan: 577-583.
- Nayak, N., Nigam, A., Sanz, J., Marston, D. und Flaxer, D. (2006). Concepts for Service-Oriented Business Thinking. IEEE International Conference on Services Computing.
- Neto, W. (2012). Adaptable UI for Web Service Composition: A Model-Driven Approach. Service-Oriented Computing - ICSOC 2011 Workshops. Pallis, G., Jmaiel, M., Charfi, A. et al, Springer Berlin Heidelberg. **7221**: 177-182.
- Nissen, V. und Bothe, M. (2002). "Fourth Party Logistics - Ein Überblick." Logistik Management(4): 16-25.
- Nissen, V., Petsch, M. und Schorcht, H. (2007). Service-orientierte Architekturen : Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Wiesbaden, Dt. Univ.-Verl.
- Nitin, N., Anil, N., Jorge, S., David, M. und David, F. (2006). Concepts for Service-Oriented Business Thinking. Proceedings of the IEEE International Conference on Services Computing, IEEE Computer Society: 357-364.
- Oster, Z. J., Ali, S. A., Santhanam, G. R., Basu, S. und Roop, P. S. (2012). A Service Composition Framework Based on Goal-Oriented Requirements Engineering, Model Checking, and Qualitative Preference Analysis. Service-Oriented Computing. Liu, C., Ludwig, H., Toumani, F. und Yu, Q. E. Shanghai, Springer Verlag: 283-297.
- Österle, H., Winter, R. und Brenner, W., Eds. (2010). Gestaltungsorientierte Wirtschaftsinformatik : ein Plädoyer für Rigor und Relevanz, Infowerk AG.
- Overhage, S. und Turowski, K. (2007). Serviceorientierte Architekturen – Konzept und methodische Herausforderungen. Service-orientierte Architekturen. Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Nissen, V., Petsch, M. und Schorcht, H. Wiesbaden, Deutscher Universitäts-Verlag: 3-18.
- Papazoglou, M. P. und Van Den Heuvel, W.-J. (2006). "Service-oriented design and development methodology." International Journal of Web Engineering and Technology **2**(4): 412-442.
- Parnas, D. L. (1972). "On the criteria to be used in decomposing systems into modules." Communications of the ACM **15**(12): 1053-1058.

- Patig, S. (2007). Eine Theorie der Evolution von Modellierungssprachen. Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik. Lehner, F. und Zelewski, S., GITO mbH Verlag: 18-33.
- Payne, A. F., Storbacka, K. und Penny, F. (2008). "Managing the co-creation of value." Journal of the Academy of Marketing Science **36**(1): 83-96.
- Petrasch, R. und Meimberg, O. (2006). Model Driven Architecture eine praxisorientierte Einführung in die MDA. Heidelberg, dpunkt.
- Petz, A., Duckwitz, S., Nielen, A. und M., S. C. (2014). Modellierung und Simulation von wissensintensiven Dienstleistungssystemen mit Design Structure Matrizen. Dienstleistungsmodellierung. Thomas, O. und Nüttgens, M. H. Wiesbaden, Springer Verlag: 33-52.
- Pfeiffer, D. (2007). Constructing comparable conceptual models with domain specific languages. 15th European Conference on Information Systems (ECIS 2007). St. Gallen, Switzerland: 876-888.
- Pfeiffer, D. (2008). Semantic Business Process Analysis - Building Block-based Construction of Automatically Analyzable Business Process Models. ERCIS. Münster, Westfälische Wilhelms-Universität Münster. **Doktor**: 311.
- Pfohl, H.-C. (2004). Logistikmanagement : Konzeption und Funktionen. Berlin [u.a.], Springer.
- Pfohl, H.-C. (2010). Logistiksysteme : betriebswirtschaftliche Grundlagen. Berlin [u.a.], Springer.
- Piccinelli, G., Emmerich, W. und Finkelstein, A. (2001). Mapping Service Components to EJB Business Objects. EDOC '01 Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing.
- Piccinelli, G., W., E., Williams, S. L. und Stearns, M. (2003). A Model-Driven Architecture for Electronic Service Management Systems. ICSSOC 2003, LNCS 2910.
- Preda, M. D., Gabrielli, M., Guidi, C., Mauro, J. und Montesi, F. (2012). Interface-Based Service Composition with Aggregation. Service-Oriented and Cloud Computing. Paoli, F. D., Pimentel, E. und Zavattaro, G. E. Bertinoro, Springer Verlag: 48-63.
- Quartel, D. C., Steen, M. A., Pokraev, S. und Sinderen, M. (2007). "COSMO: A conceptual framework for service modelling and refinement." Information Systems Frontiers **9**(2-3): 225-244.
- Ramaswamy, V. (2006). "Co-Creating Experiences of value with customers." Infosys **4**(1): 25-36.
- Ramaswamy, V. (2008). "Co-creating value through customers' experience: the Nike case." Strategy + Leadership **36**(5): 9-14.

- Remme, M. (1997). Konstruktion von Geschäftsprozessen Ein modellgestützter Ansatz durch Montage generischer Prozeßpartikel. Wiesbaden, Gabler Verlag.
- Romero, J. R., Jaén, J. I. und Vallecillo, A. (2009). Realizing Correspondences in Multi-viewpoint Specifications. 2009 IEEE International Enterprise Distributed Object Computing Conference (EDOC 2009), Auckland, New Zealand, IEEE.
- Rowley, J. (2006). "An analysis of the e-service literature: towards a research agenda." Internet Research **16**(3): 339-359.
- Roy, M., Suleiman, B. und Weber, I. (2010). Facilitating Enterprise Service Discovery for Non-technical Business Users. Service-Oriented Computing. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 100-110.
- Rust, R. T. und Kannan, P. K. (2002). e-Service: New Direction in Theory and Practice. Armonk New York, M.E. Sharpe.
- Rust, R. T. und Kannan, P. K. (2003). "E-Service: A New Paradigm for Business in the Electronic Environment." Communications of the ACM **46**(6).
- Saini, A., Nanchen, B. und Evequoz, F. (2013). Putting the Customer Back in the Center of SOA with Service Design and User-Centered Design. Service-Oriented and Cloud Computing. Lau, K.-K., Lamersdorf, W. und Pimentel, E. E. Malaga, Springer Verlag: 94-103.
- Sampson, S. E. (2001). Understanding Service Businesses: Applying Principles of Unified Systems Theory. New York, John Wiley & Sons.
- Savaskan, R. C., Bhattacharya, S. und Wassenhove, L. N. v. (2004). "Closed-Loop Supply Chain Models with Product Remanufacturing." Management Science **50**(2): 239-252.
- Scheer, A.-W. (1997). Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. Berlin et al., Springer Verlag.
- Scheer, A.-W., Grieble, O. und Klein, R. (2006). Modellbasiertes Dienstleistungsmanagement. Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen. Bullinger, H.-J. und Scheer, A.-W. H. Berlin, Heidelberg, New York, Springer Verlag: 19-51.
- Scheithauer, G., Voigt, K., Bicer, V., Heinrich, M., Strunk, A. und Winkler, M. (2009). Integrated service engineering workbench: service engineering for digital ecosystems. Proceedings of the International Conference on Management of Emergent Digital EcoSystems, France, ACM.
- Schlieter, H. J., Martin; Niggemann, Stephan (2010). The Challenge of Energy Management - Status-Quo and Perspectives for Reference Models. Pacific Asia Conference on Information Systems (PACIS), Taipei.

- Schmedes, T. (2007). Modellierung service-orientierter Architekturen in der Energieversorgung. Software Engineering Workshop 2007, Hamburg, Gesellschaft für Informatik e.V. (GI).
- Schmitt, A. (2006). 4PL-Providing-TM als strategische Option für Kontraktlogistikdienstleister : eine konzeptionell-empirische Betrachtung. Wiesbaden, Dt. Univ.-Verl.
- Schütte, R. (1997). Die neuen Grundsätze ordnungsmäßiger Modellierung. Forschungsforum 1997. Leipzig.
- Schütte, R. (1998). Grundsätze ordnungsmäßiger Referenzmodellierung Konstruktion konfigurations- und anpassungsorientierter Modelle. Wiesbaden ;s.l., Gabler Verlag.
- Selonen, P. und Kettunen, M. (2007). Metamodel-based inference of inter-model correspondence. 11th European Conference on Software Maintenance and Reengineering (CSMR '07), Amsterdam, the Netherlands IEEE.
- Shostack, G. L. (1984). "Designing services that deliver." Harvard Business Review **62**: 133-139.
- Shrinivasan, Y. B., Dasgupta, G. B., Desai, N. und Nallacherry, J. (2012). A Method for Assessing Influence Relationships among KPIs of Service Systems. Service-Oriented Computing. Liu, C., Ludwig, H., Toumani, F. und Yu, Q. E. Shanghai, Springer Verlag: 191-205.
- Simon, L., Bansal, A., Mallya, A. und Gupta, G. (2005). A Universal Service Semantics Description Language. 3rd IEEE European Conference on Web Services (ECOWS 2005), Växjö, Sweden, IEEE Computer Society.
- Smit, M. und Stroulia, E. (2010). Configuration Decision Making Using Simulation-Generated Data. Service-Oriented Computing. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 15-26.
- Spath, D. und Ganz, W., Eds. (2008). The Future of Services : Trends and Perspectives. München, Hanser.
- Spohrer, J. und Maglio, P. P. (2008). "The emergence of Service Science: Toward systematic service innovations to accelerate co-creation of value." Production and Operations Management **17**(3): 238-246.
- Spohrer, J., Maglio, P. P., Bailey, J. und Gruhl, D. (2007). "Steps toward a science of service systems." IEEE Computer(40): 71-77.
- Spohrer, J., Vargo, S. L., Caswell, N. und Maglio, P. P. (2008). The Service System is the basic abstraction of Service Science. HICSS, Hawai, IEEE.
- Staab, S. und Studer, R. (2009). Handbook on ontologies. Berlin [u.a.], Springer.
- Stachowiak, H. (1973). Allgemeine Modelltheorie. Wien [u.a.], Springer.

- Stahl, T., Völter, M., Efftinge, S. und Haase, A. (2007). Modellgetriebene Softwareentwicklung : Techniken, Engineering, Management. Heidelberg, dpunkt-Verl.
- Steen, M. W. A., Iacop, M. E., Lankhorst, M. M., Jonkers, H., Zoet, M., Engelsmann, W., Versendaal, J., Proper, H. A., Debije, L. und Gaaloul, K. (2012). Service Modeling. Agile Service Development - Combining Adaptive methods and flexible solutions. Lankhorst, M. Berlin Heidelberg, Springer Berlin Heidelberg: 59-94.
- Störrle, H. und Glock, W. (2007). Geschäftsprozessmodellierung für Service-Orientierte Architekturen. Service-orientierte Architekturen. Nissen, V., Petsch, M. und Schorcht, H. Wiesbaden, Deutscher Universitäts-Verlag: 77-94.
- Strahring, S. (1996). Metamodellierung als Instrument des Methodenvergleichs eine Evaluierung am Beispiel objektorientierter Analysemethoden. Aachen, Shaker.
- Strahring, S. (1998). Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. CEUR Workshop Proceedings zur Modellierung 1998, Münster, Gesellschaft für Informatik (GI).
- Tapscott, D. und Ticoll, D. (2003). The Naked Corporation: How the Age of Transparency Will Revolutionize Business, Free Press.
- Teleroute (2005). What it costs to manage collaborative logistics. Brussels, Belgium, Teleroute Integrated solutions.
- Terry, L. (2009). 2009 Third party logistics - The state of logistics outsourcing, Capgemini, the Georgia Institute of Technology, Panalpina and eyefortransport. **14**.
- Terry, L. (2010). 2010 Third-party logistics - The state of logistics outsourcing, Capgemini, the Georgia Institute of Technology, Panalpina and eyefortransport. **15**.
- Terry, L. (2012). 2012 Third-party logistics - The state of logistics outsourcing, Capgemini, the Georgia Institute of Technology, Panalpina and eyefortransport. **16**.
- Thomas, O., Leyking, K., Dreifus, F., Fellmann, M. und Loos, P. (2007). Serviceorientierte Architekturen : Gestaltung, Konfiguration und Ausführung von Geschäftsprozessen. Veröffentlichungen des Instituts für Wirtschaftsinformatik im Deutschen Forschungszentrum für Künstliche Intelligenz. Loos, P. Saarbrücken: 1-21.
- Thomas, O. und Nüttgens, M. (2010). Dienstleistungsmodellierung 2010 : interdisziplinäre Konzepte und Anwendungsszenarien. Berlin [u.a.], Physica-Verl.
- Thomas, O. und Scheer, A.-W. (2003). Referenzmodell-basiertes (Reverse-) Customizing von Dienstleistungsinformationssystemen (REBECA). Saarbrücken, Universität Saarbrücken.
- Upadhyaya, B., Zou, Y., Wang, S. und Ng, J. (2013). Automatically Composing Services by Mining Process Knowledge from the Web. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 267-282.

- Uribarren, A., Parra, J., Uribe, J. P., Makibar, K., Olalde, I. und Herrasti, N. (2006). "Service Oriented Pervasive Applications Based On Interoperable Middleware." 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI).
- Uschold, M. und Gruninger, M. (2004). "Ontologies and Semantics for Seamless Connectivity." SIGMOD **33**(4): 58-64.
- Vahrenkamp, R. und Siepermann, C. (2005). Logistik : Management und Strategien. München [u.a.], Oldenbourg.
- Vargo, S. L. und Lusch, R. F. (2004). "Evolving to a new dominant logic for marketing." Journal of Marketing **68**: 1-17.
- Vargo, S. L., Maglio, P. P. und Akaka, M. A. (2008). "On value and value co-creation: A service systems and service logic perspective." European Management Journal **26**(3): 145-152.
- Verhoef, T. F., ter Hofstede, A. H. M. und Wijers, G. M. (1991). Structuring modelling knowledge for CASE shells. Advanced Information Systems Engineering. Andersen, R., Bubenko, J., Jr. und Sølvberg, A., Springer Berlin Heidelberg. **498**: 502-524.
- Vogel, T. (2009). Serviceorientiertes Business Networking : Referenzarchitektur und Gestaltungsprinzipien.
- Vom Brocke, J. (2003). Referenzmodellierung - Festlegung und Verteilung von Konstruktionsprozessen. Berlin, Logos Verlag.
- vom Brocke, J. B., Christian (2006a). Reusable Conceptual Models. Requirements Based on the Design Science Research Paradigm. First International Conference on Design Science Research in Information Systems and Technology (DESRIST). Claremont, USA.
- vom Brocke, J. T., Oliver (2006b). Reference Modeling for Organizational Change. Applying Collaborative Techniques for Business Engineering. Americas Conference on Information Systems (AMCIS), Acapulco, Mexico.
- von der Gracht, H., Däneke, E., Micic, P., Darkow, I.-L. und Jahns, C. (2008). Zukunft der Logistik-Dienstleistungsbranche in Deutschland 2025, BVL, Bundesvereinigung Logistik.
- von Foerster, H. und Glasersfeld, E. v. (2007). Wie wir uns erfinden : eine Autobiographie des radikalen Konstruktivismus. Heidelberg, Carl-Auer-Verl.
- von Glasersfeld, E. (1997). Radikaler Konstruktivismus : Ideen, Ergebnisse, Probleme. Frankfurt am Main, Suhrkamp.
- Wagner, F., Ishikawa, F. und Honiden, S. (2012). Applying QoS-Aware Service Selection on Functionally Diverse Services. Service-Oriented Computing – ICSOC 2011

- Workshops. Pallis, G., Jmaiel, M., Graupner, S. et al. Paphos, Springer Verlag: 100-113.
- Walkerdine, J., Hutchinson, J., Sawyer, P., Dobson, G. und Onditi, V. (2007). A Faceted Approach to Service Specification. 2nd International Conference on Internet and Web Applications and Services (ICIW 07). Morne, Mauritius, IEEE Computer Society.
- Wang, L., Wang, H., Yu, Q., Sun, H. und Bouguettaya, A. (2013). Online Reliability Time Series Prediction for Service-Oriented System of Systems. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 421-428.
- Weigand, H., Johannesson, P., Andersson, B. und Berholtz, M. (2009). "Value-Based Service Modeling and Design: Toward a Unified View of Services." Advanced Information Systems Engineering **5565**: 410-424.
- Wilde, T. und Hess, T. (2006). Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung. München, Institut für Wirtschaftsinformatik und neue Medien. **Arbeitsbericht Nr. 2 / 2006**.
- Wilde, T. und Hess, T. (2007). Forschungsmethoden der Wirtschaftsinformatik: Eine empirische Untersuchung. Wirtschaftsinformatik. **49**: 280-287.
- Ye, Z., Bouguettaya, A. und Zhou, X. (2013). QoS-Aware Cloud Service Composition Using Time Series. Service-Oriented Computing. Basu, S., Pautasso, C., Zhang, L. und Xiang, F. E. Berlin, Springer Verlag: 9-22.
- Yu, J., Han, J., Gunarso, S. O. und Versteeg, S. (2013). A Business Protocol Unit Testing Framework for Web Service Composition. Advanced Information Systems Engineering. Salinesi, C., Norrie, M. C. und Pastor, O. s. E. Valencia, Springer Verlag: 17-34.
- Zachman, J. A. (1987). "A framework for information systems architecture." IBM Systems Journal **26**(3): 276-292.
- Zahn, E. und Stanik, M. (2006). Integrierte Entwicklung von Dienstleistungen und Netzwerken - Dienstleistungskooperationen als strategischer Erfolgsfaktor. Service engineering : Entwicklung und Gestaltung innovativer Dienstleistungen. Bullinger, H.-J. und Scheer, A.-W. Berlin, Springer: 299-320.
- Zelewski, S. (2006). Fortschritt in den Wirtschaftswissenschaften : wissenschaftstheoretische Grundlagen und exemplarische Anwendungen. Wiesbaden, Dt. Univ.-Verl.
- Zirpins, C., Feuerlicht, G., Lamersdorf, W. und Guadalupe, O. (2010). Engineering Service-Oriented Applications. Service-Oriented Computing - 6th International Workshop WESOA 2010. Maximilien, M. E., Rossi, G., Yuan, S.-T., Ludwig, H. und Fantinato, M. E. San Francisco, Springer Verlag: 62-64.

Zolnowski, A. und Böhmann, T. (2013). Grundlagen service-orientierter Geschäftsmodelle. Service-orientierte Geschäftsmodelle. Böhmann, T., Warg, M. und Weiß, P. H. Berlin, Heidelberg, Springer Verlag: 164.

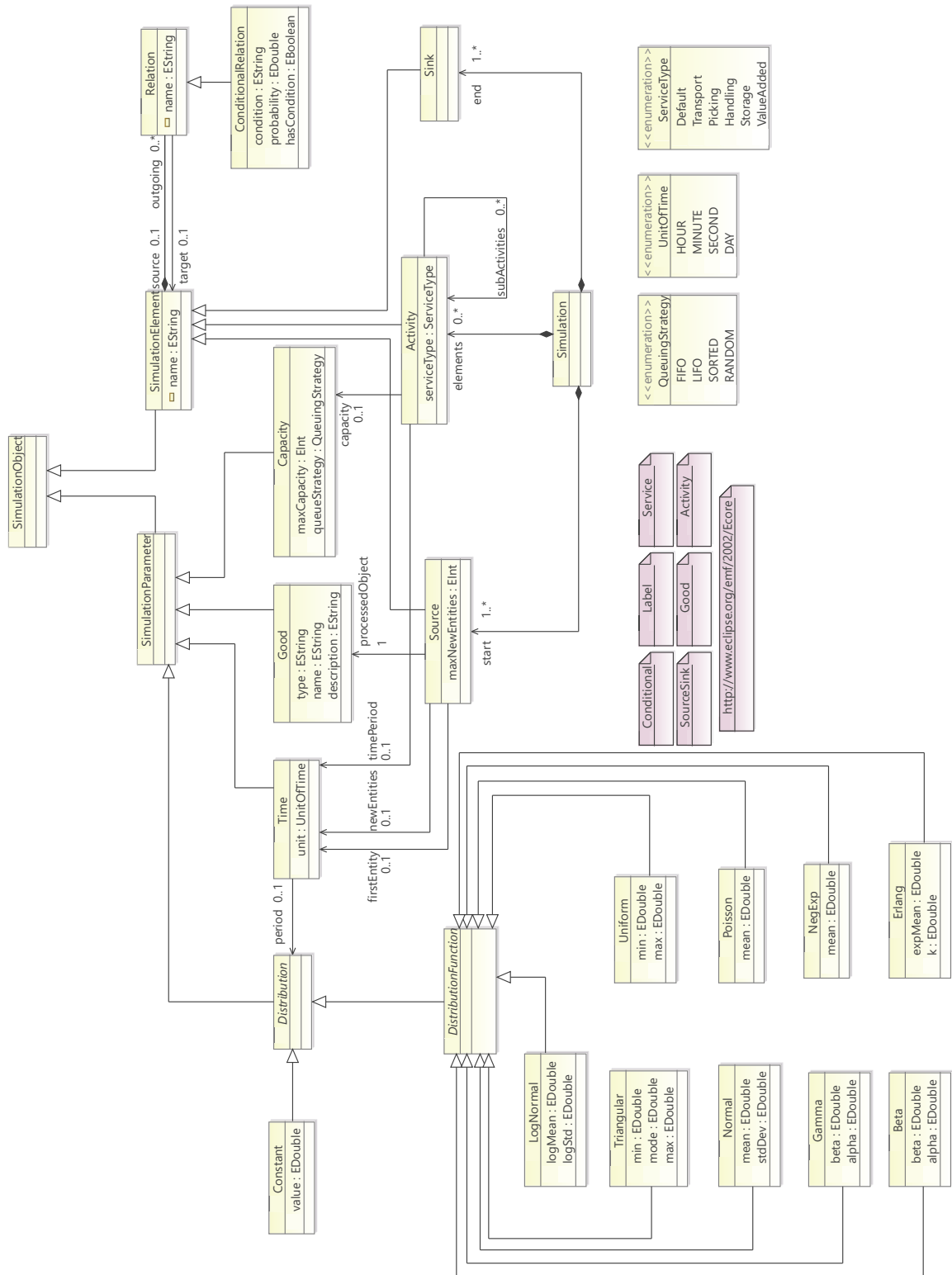
Anhang

Anhang 1 Validierungsmodelle

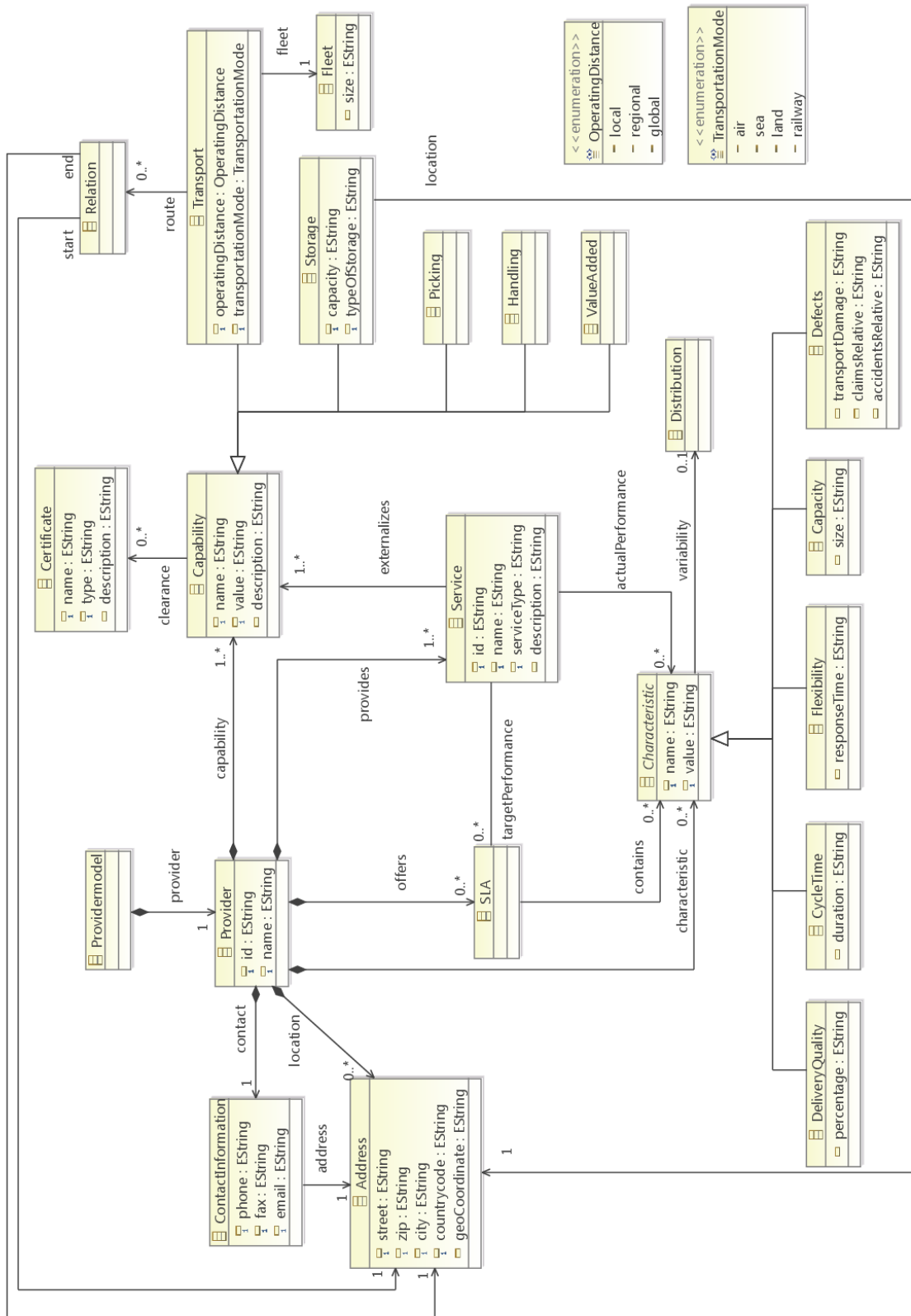
Der Anhang 1 enthält einen Überblick über die bei der prototypischen Validierung verwendeten Metamodelle. Es sind die entwickelten Simulations- (s. Anhang 1.1) und Dienstleistermodelle (s. Anhang 1.2) abgebildet. Das BPMN-Metamodell ist jedoch zu umfassend, um es vollständig abzubilden, so dass auf dessen Darstellung verzichtet wird. Da BPMN im Gegensatz zu den anderen Modellen jedoch standardisiert ist, kann auf die Spezifikation der OMG⁹ verwiesen werden.

⁹ <http://www.bpmn.org/>

Anhang 1.1 Simulationsmetamodell

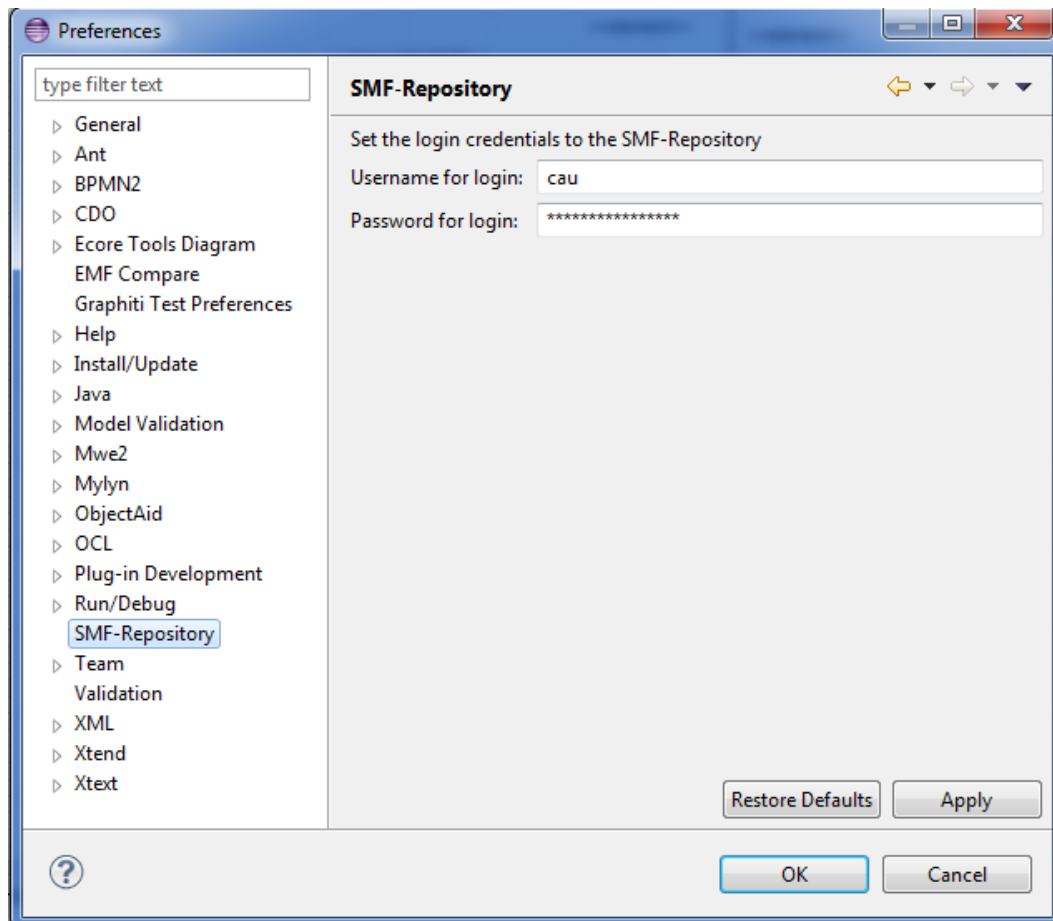


Anhang 1.2 Dienstleistermetamodell

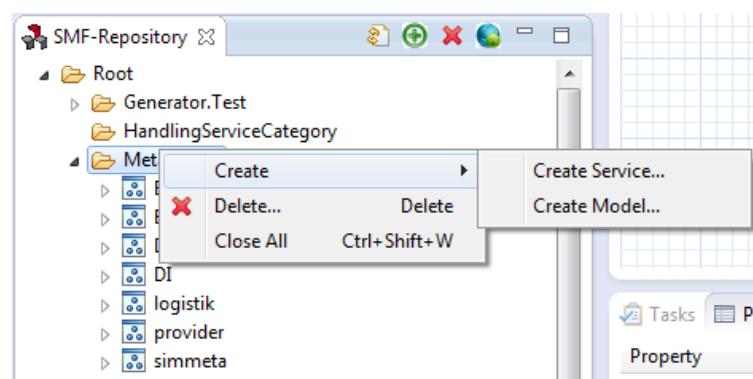


Anhang 2 Prototyp

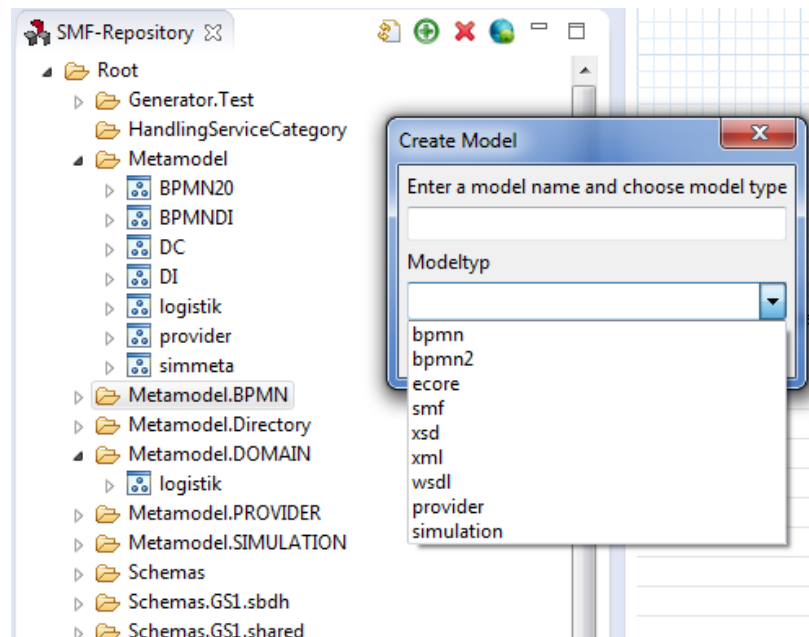
Der Anhang 2 enthält mehrere Screenshots des Prototyps, wie er für die Validierung verwendet wurde. Dargestellt werden die wichtigsten Schritte von der ersten Verwendung bis hin zur vollständigen Modellierung der Abhängigkeiten mit Hilfe des Service Editors.



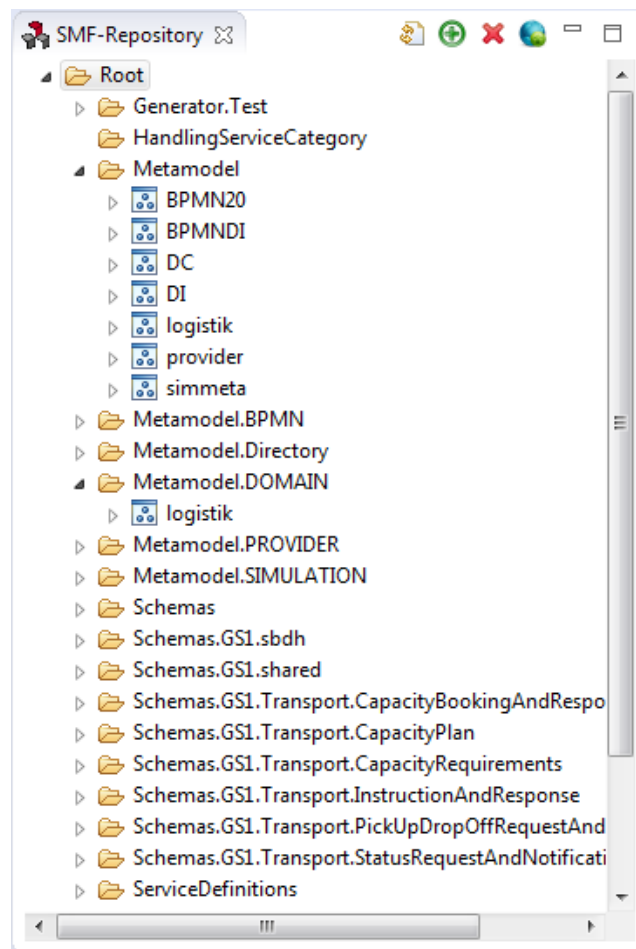
Vor der eigentlichen Arbeit mit dem Prototyp müssen zunächst Benutzer und Passwort hinterlegt werden, um Zugriff auf das Service Repository zu erlangen.



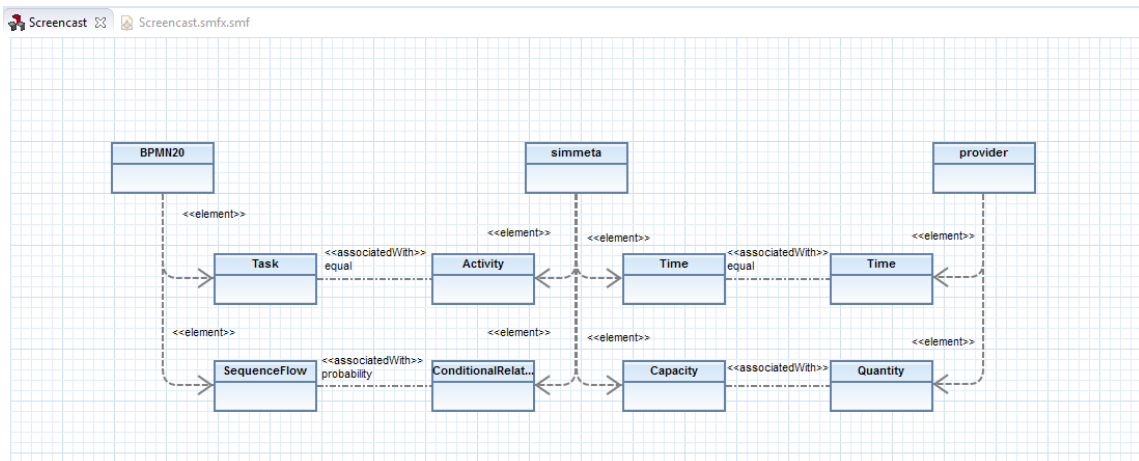
Sobald dies geschehen ist, steht der Inhalt des Repositorys zur Verfügung. Danach können Services und Modelle bspw. auch hinzugefügt werden.



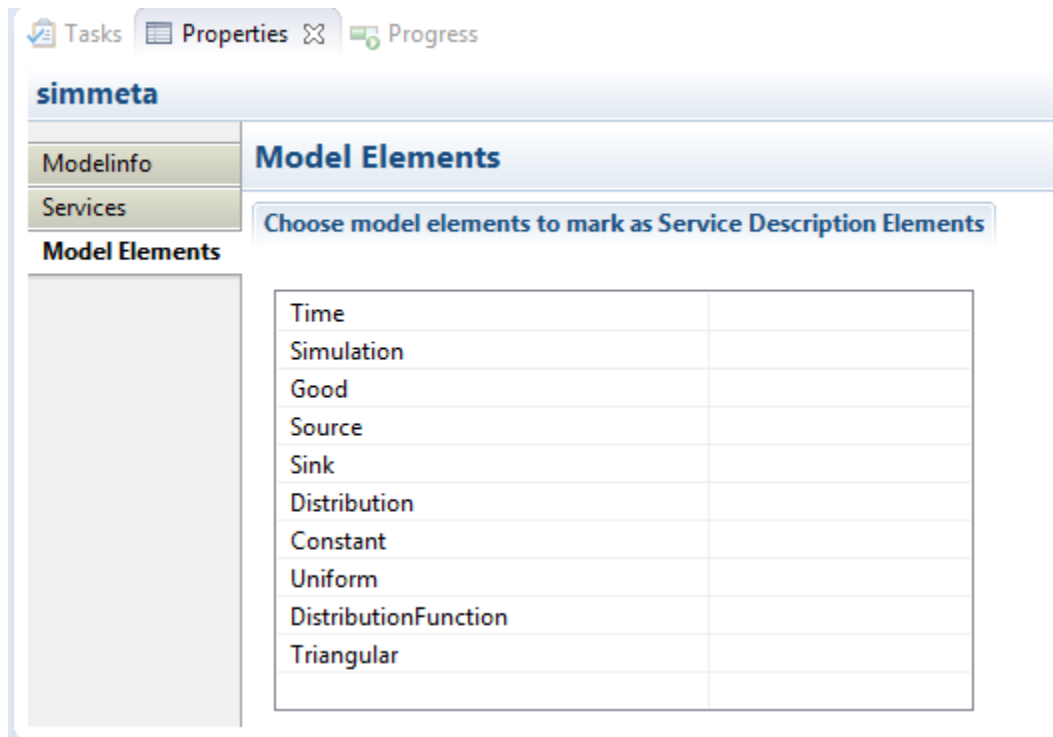
Dies kann einerseits über Drag&Drop geschehen, aber auch manuell.



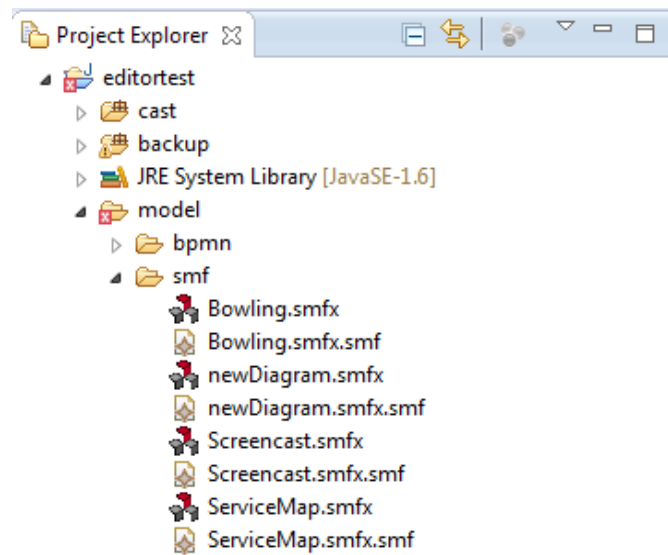
Im Repository werden Services, Modelle und auch Modellversionen gespeichert und können ebenfalls per Drag&Drop in den SMF Editor hingezogen werden.



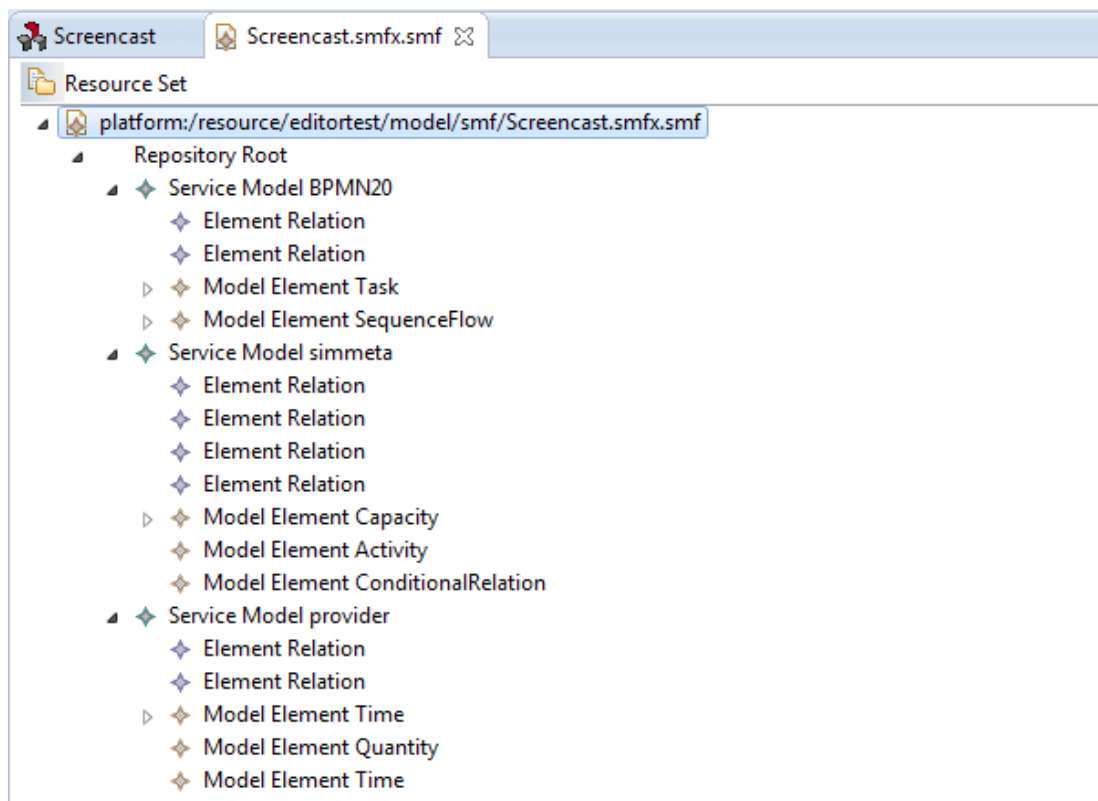
Die Elemente BPMN20, simmeta sowie provider repräsentieren Metamodelle, die aus dem Repository in den Editor gezogen wurden. Die Elemente darunter sind Modellelemente, die im entsprechenden Metamodell definiert sind.



Falls Metamodelle aus dem Repository geladen werden, stehen alle Modellelemente über den Reiter Model Elements in den Eigenschaften des Editors zur Verfügung. Das Bild zeigt alle Modellelemente des Simulationsmetamodells, die bislang noch nicht im Editor bei der Modellierung verwendet wurden. Sämtliche zur Verfügung stehenden Modellelemente können ebenfalls per Drag&Drop in das Editorfenster gezogen werden.



Soll mit Hilfe des Editors ein multiperspektivisches Servicemodell erstellt werden, muss zunächst ein neues Diagramm erzeugt werden. Die grafischen Elemente werden dabei in einer Datei mit der Endung `*.smfx` gespeichert. Sobald Beziehungen zwischen Modellen und Modellelementen beschrieben sind und abgespeichert werden sollen, wird automatisch eine Datei mit der Endung `*.smf` erzeugt, die denselben Namen wie das Diagramm trägt. In dieser Datei sind alle modellierten Zusammenhänge als eigenständiges Modell (multiperspektivisches Servicemetamodell) gespeichert. Das zugehörige Metamodell ist das Basismetamodell.

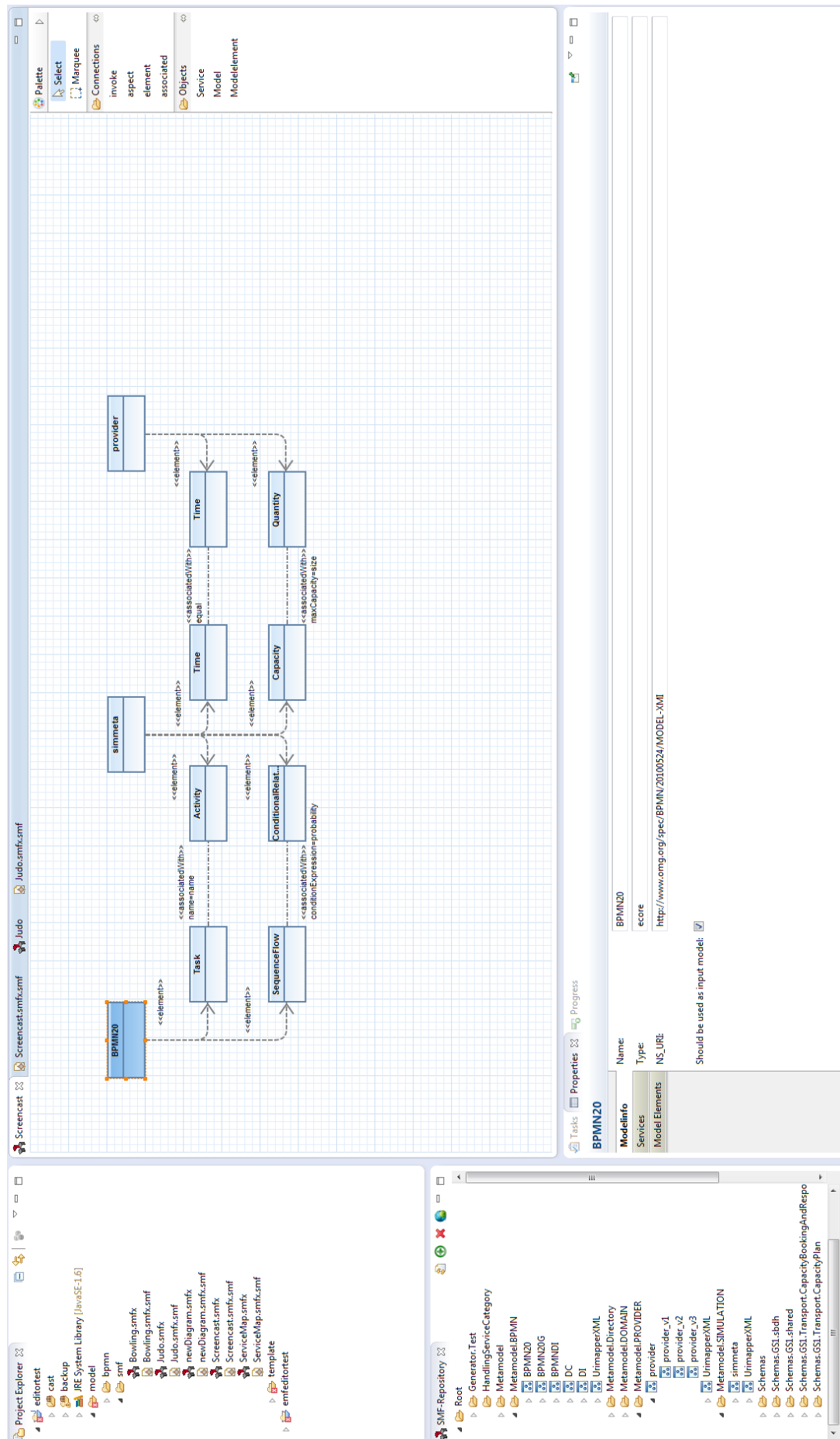


Beide Dateien, sowohl das Diagramm als auch das Servicemodell werden zunächst lokal gespeichert und müssen erst in das Repository geladen werden. Das Servicemodell umfasst alle Elemente und Beziehungen, die modelliert wurden und dient als Input für die anschließende Verarbeitung bzw. Transformation.

BPMN20		
Modelinfo	Name:	BPMN20
Services	Type:	ecore
Model Elements	NS_URI:	http://www.omg.org/spec/BPMN/20100524/MODEL-XMI

Should be used as input model:

Zur Festlegung der Quell- und Zielmodelle kann bei den Eigenschaften der im Editor dargestellten Modelle ein Haken für die Inputmodelle gesetzt werden, so dass bei der Transformation die korrekte Richtung der Informationsübertragung berücksichtigt werden kann.

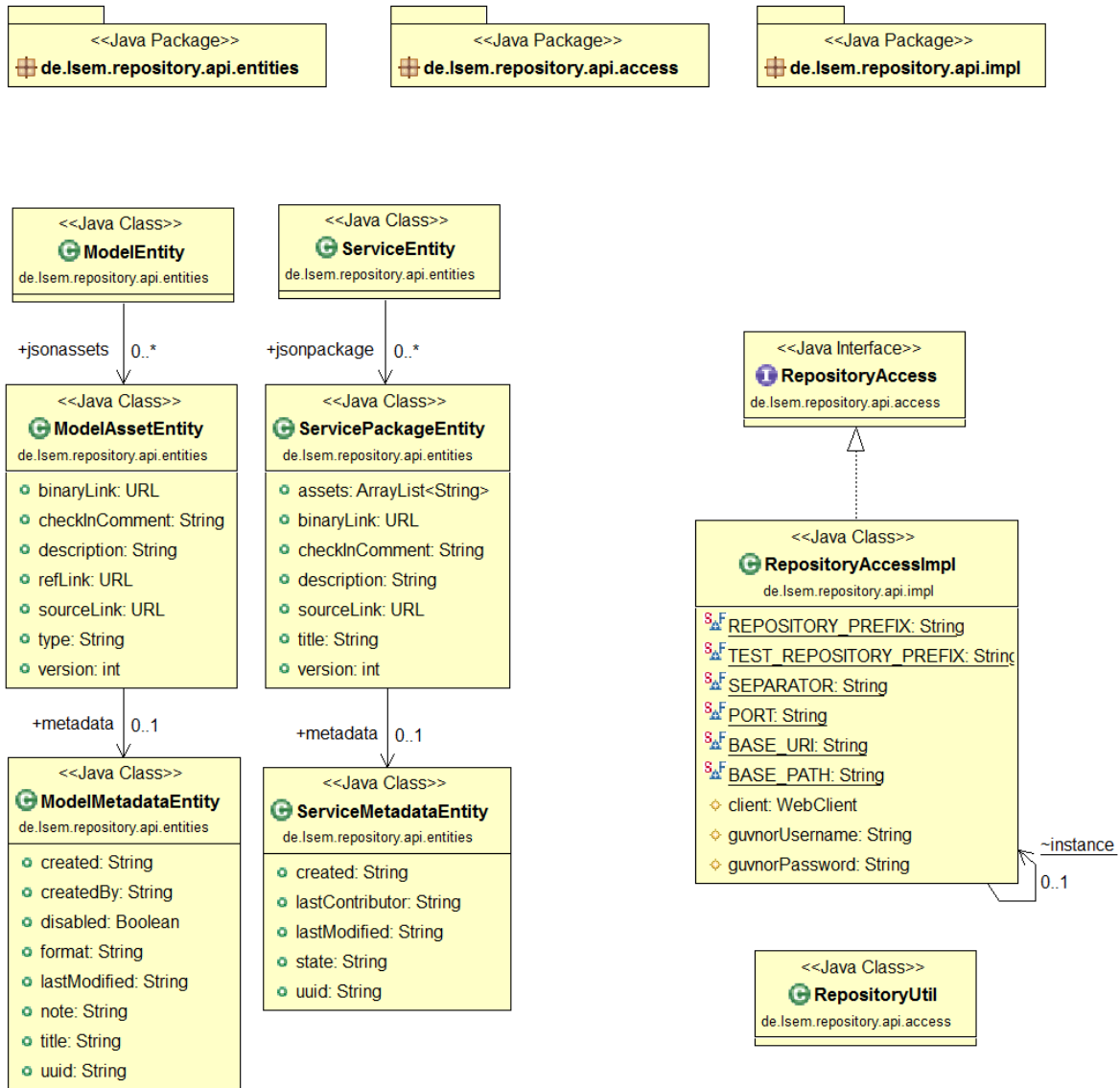


Alle Komponenten in einer integrierten Entwicklungsumgebung.

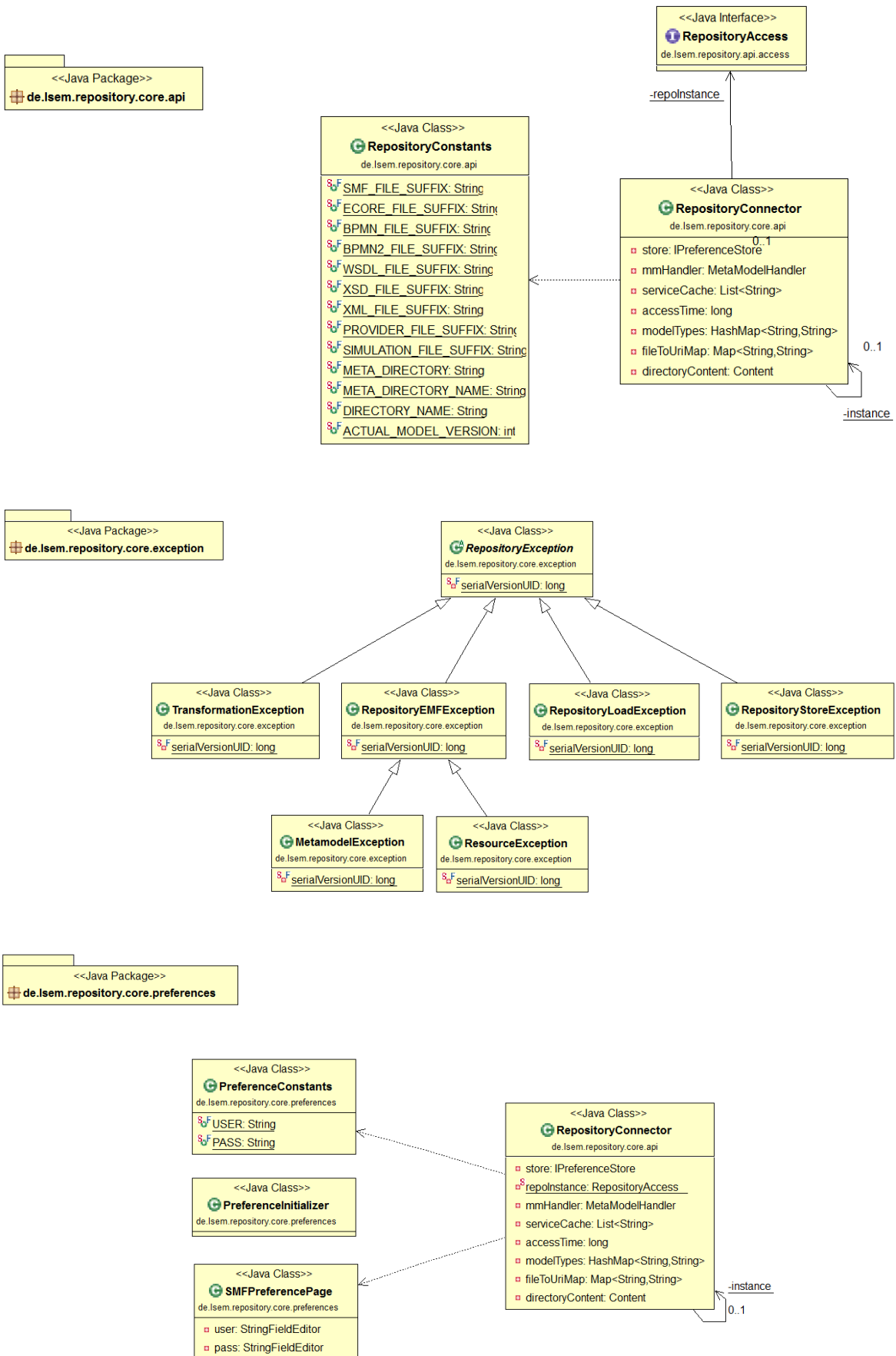
Anhang 3 UML

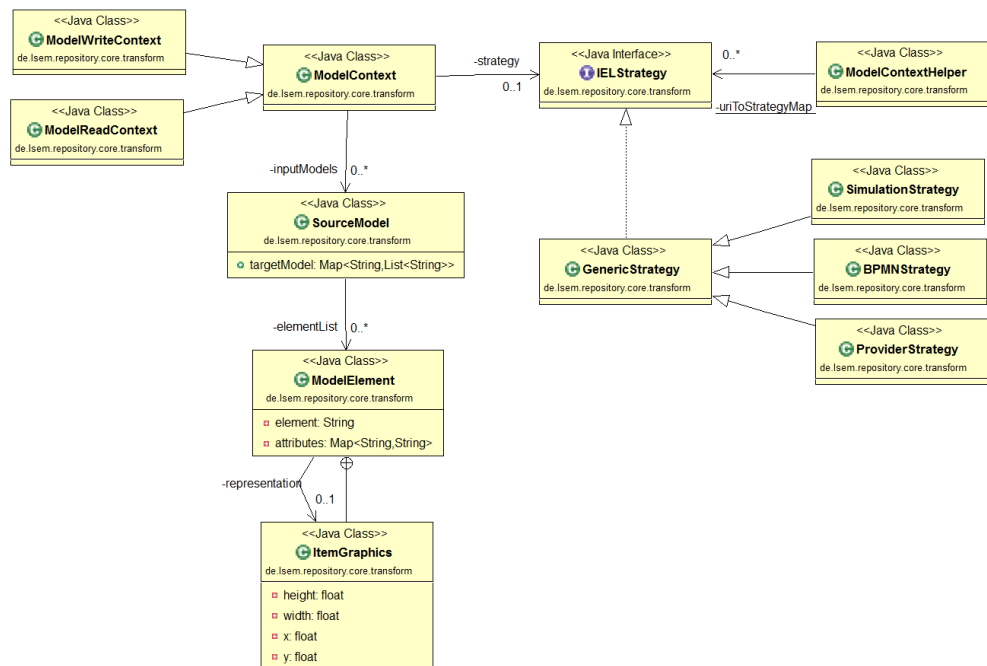
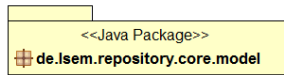
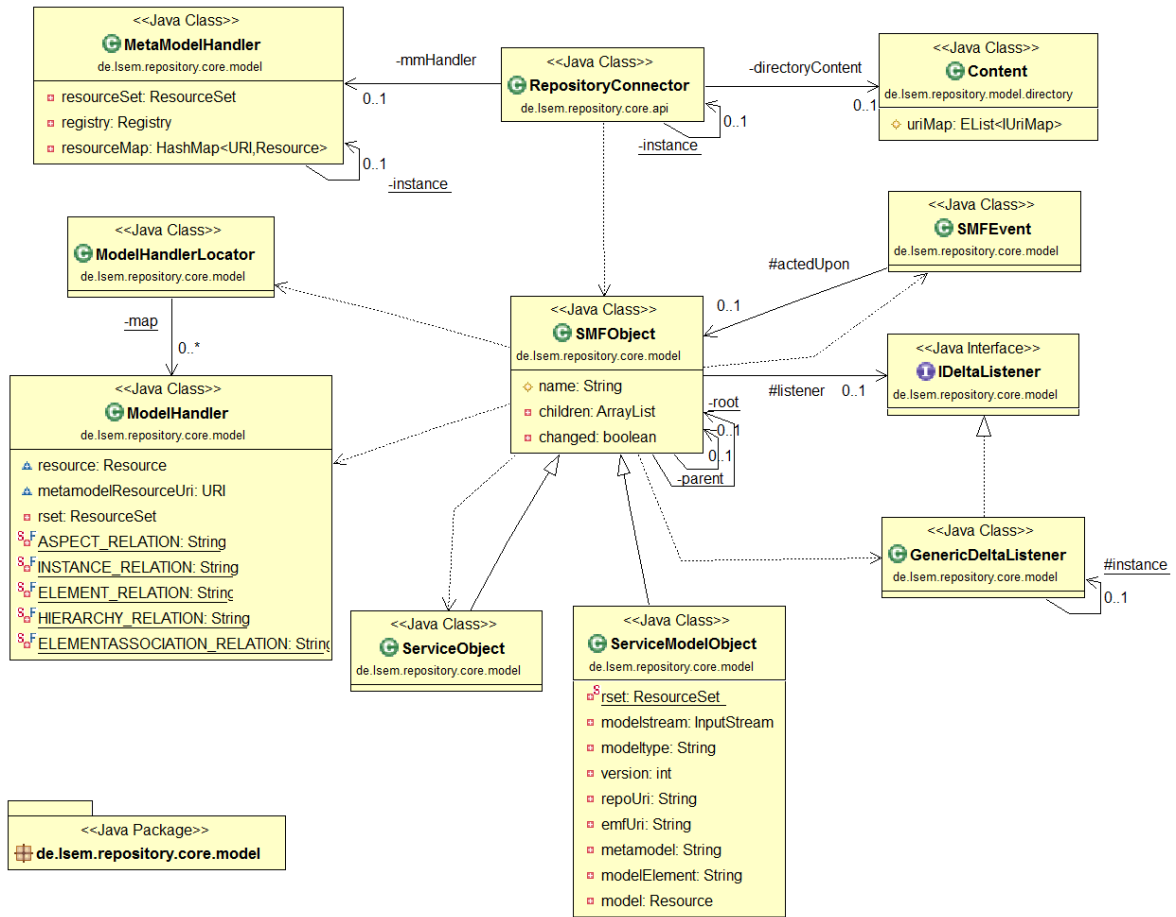
Der Anhang 3 enthält eine Darstellung der implementierten Klassen in UML-Notation. Für jede Komponente (Anhang 3.X) sind die enthaltenen Packages sowie die Klassen inkl. Attribute und deren gegenseitige Abhängigkeiten abgebildet. Auf die Methoden wurde aus Platzgründen verzichtet. Außerdem wurden externe Abhängigkeiten zu wichtigen Frameworks wie bspw. Eclipse oder Graphiti ggf. hinzugefügt.

Anhang 3.1 UML - SMF-API

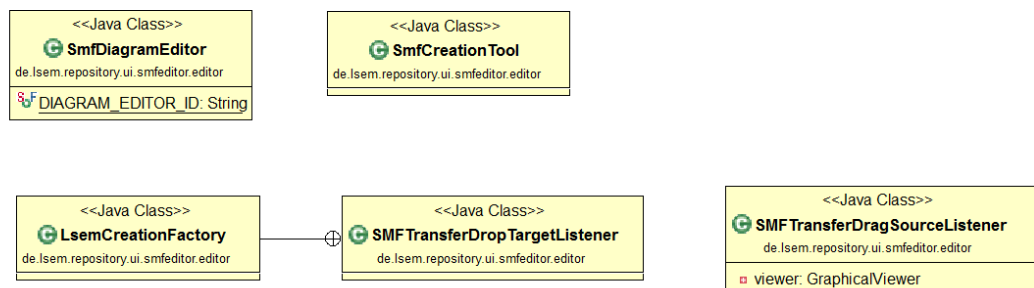
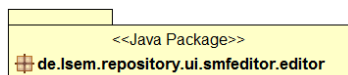
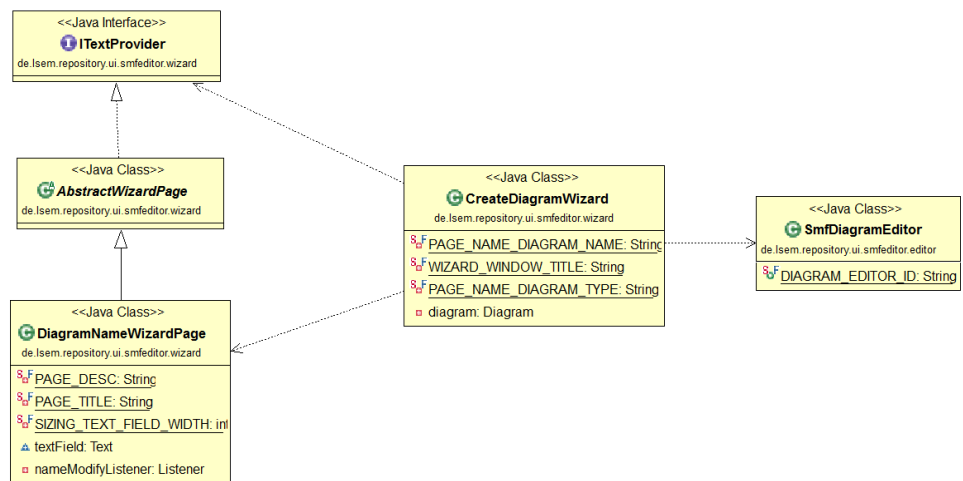
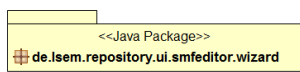
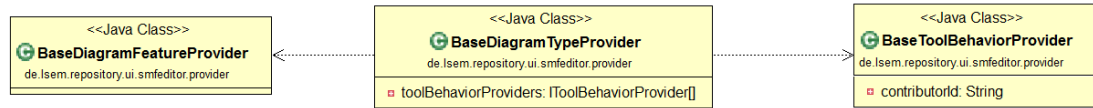
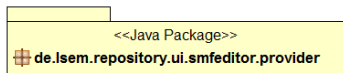


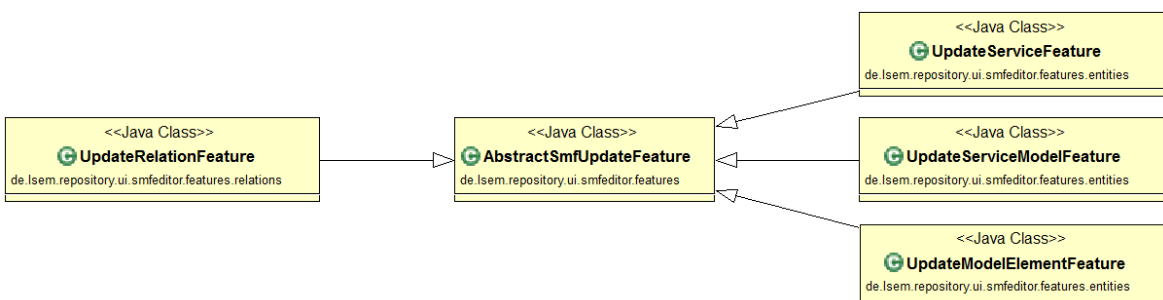
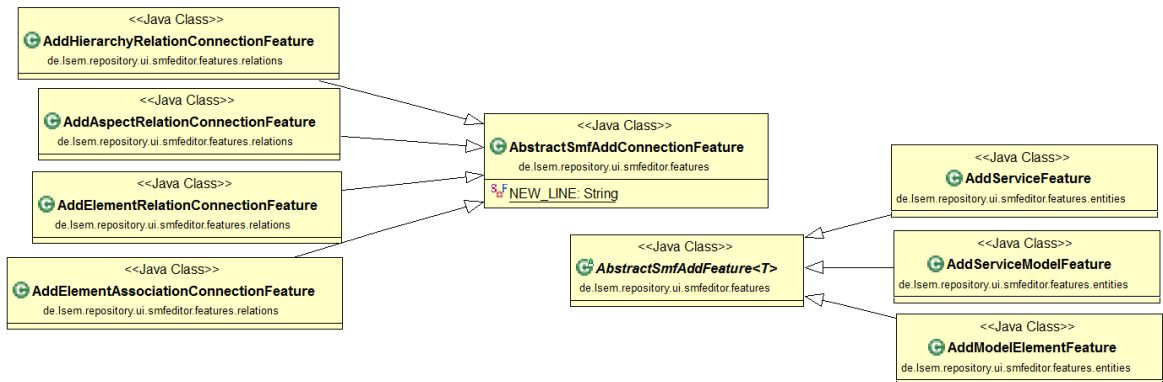
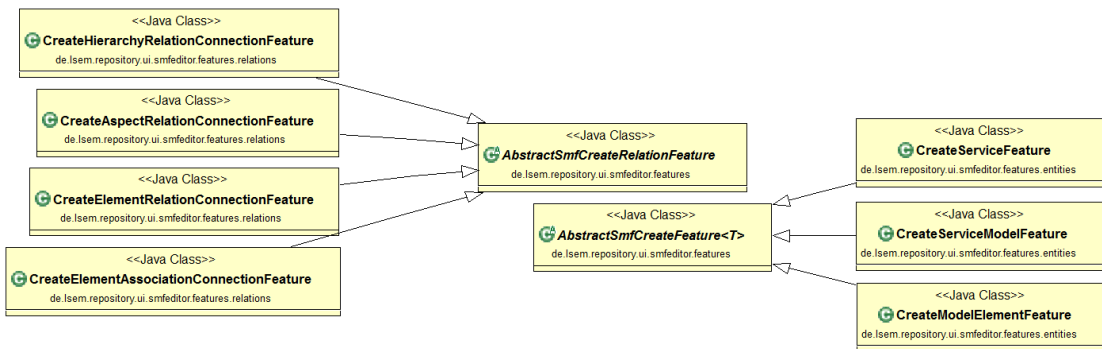
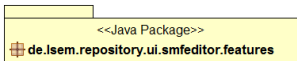
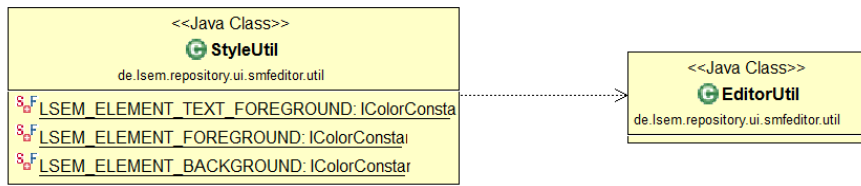
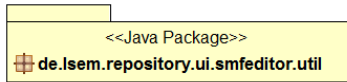
Anhang 3.2 UML - SMF-Core

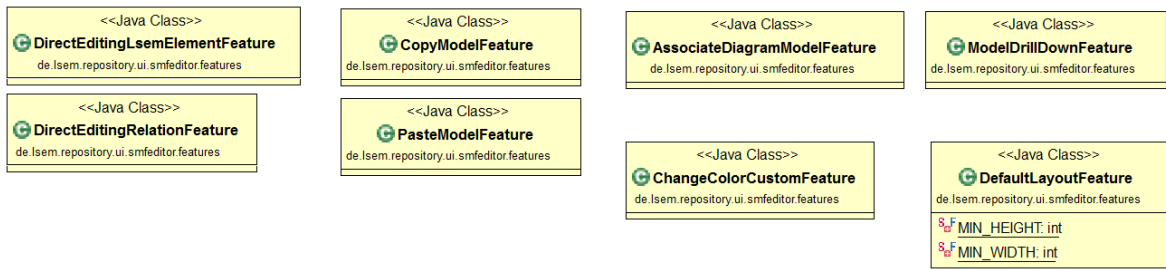




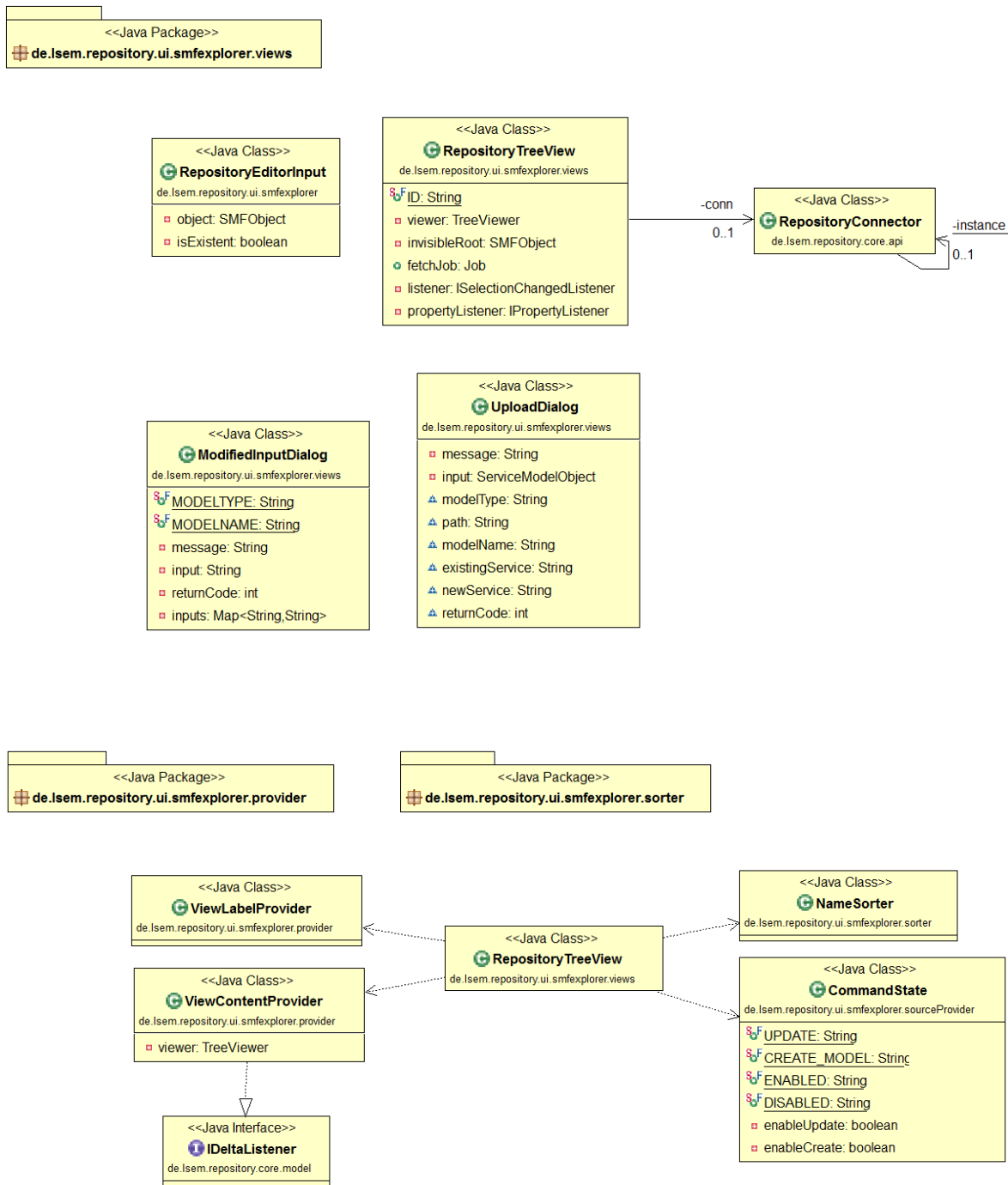
Anhang 3.3 UML - SMF-Editor

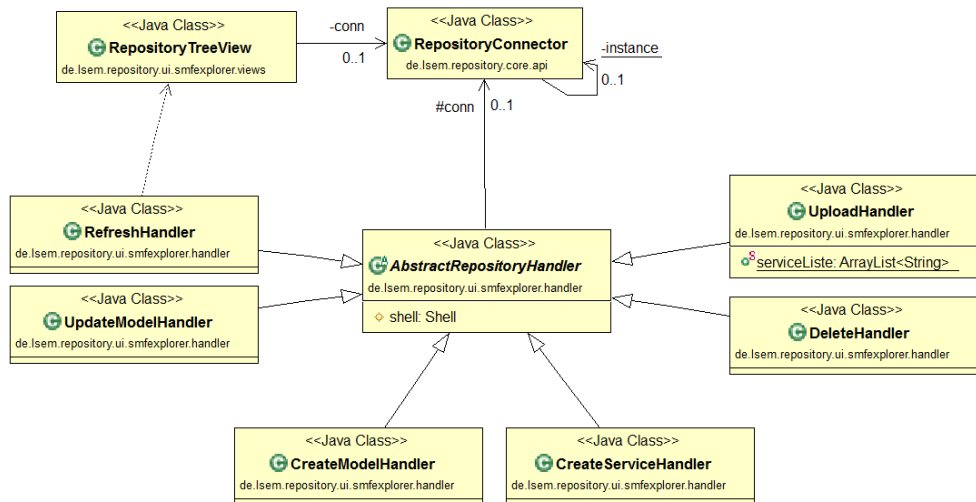
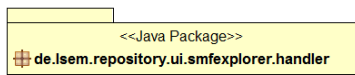
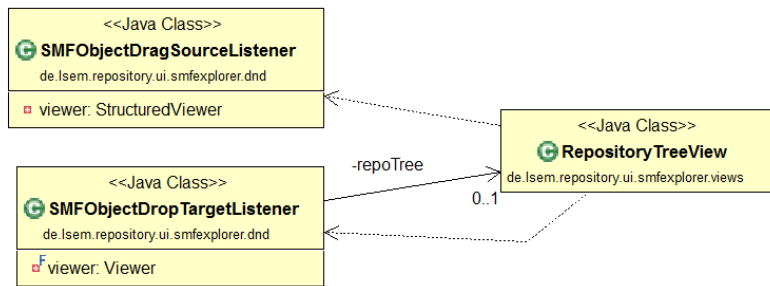
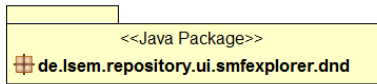




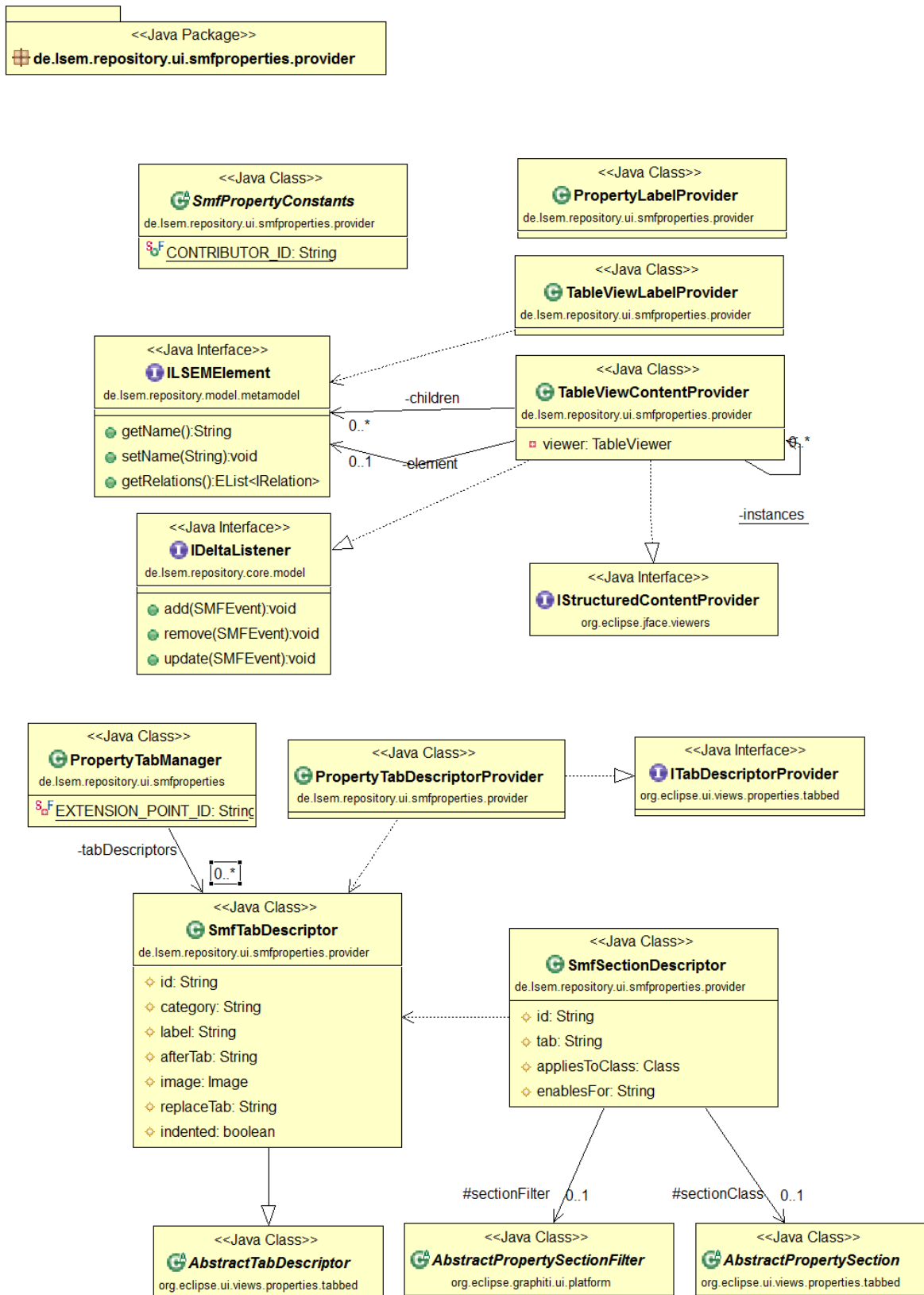


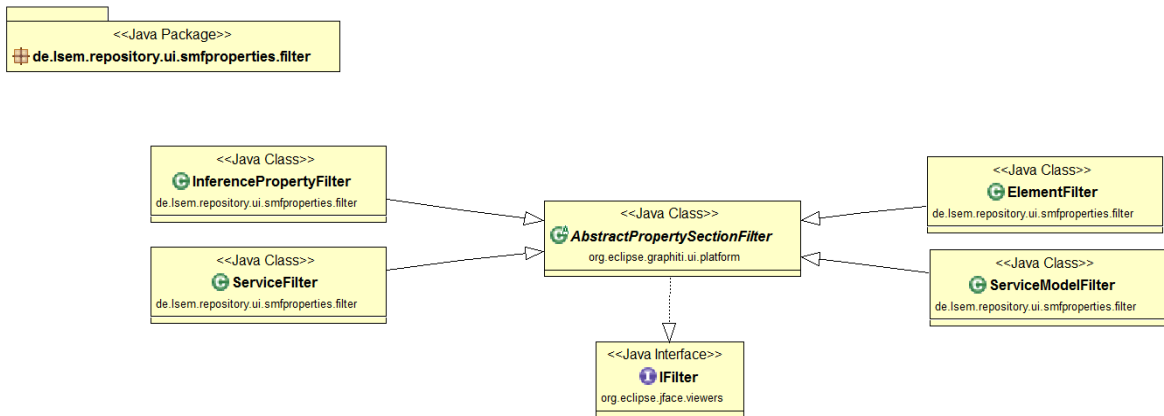
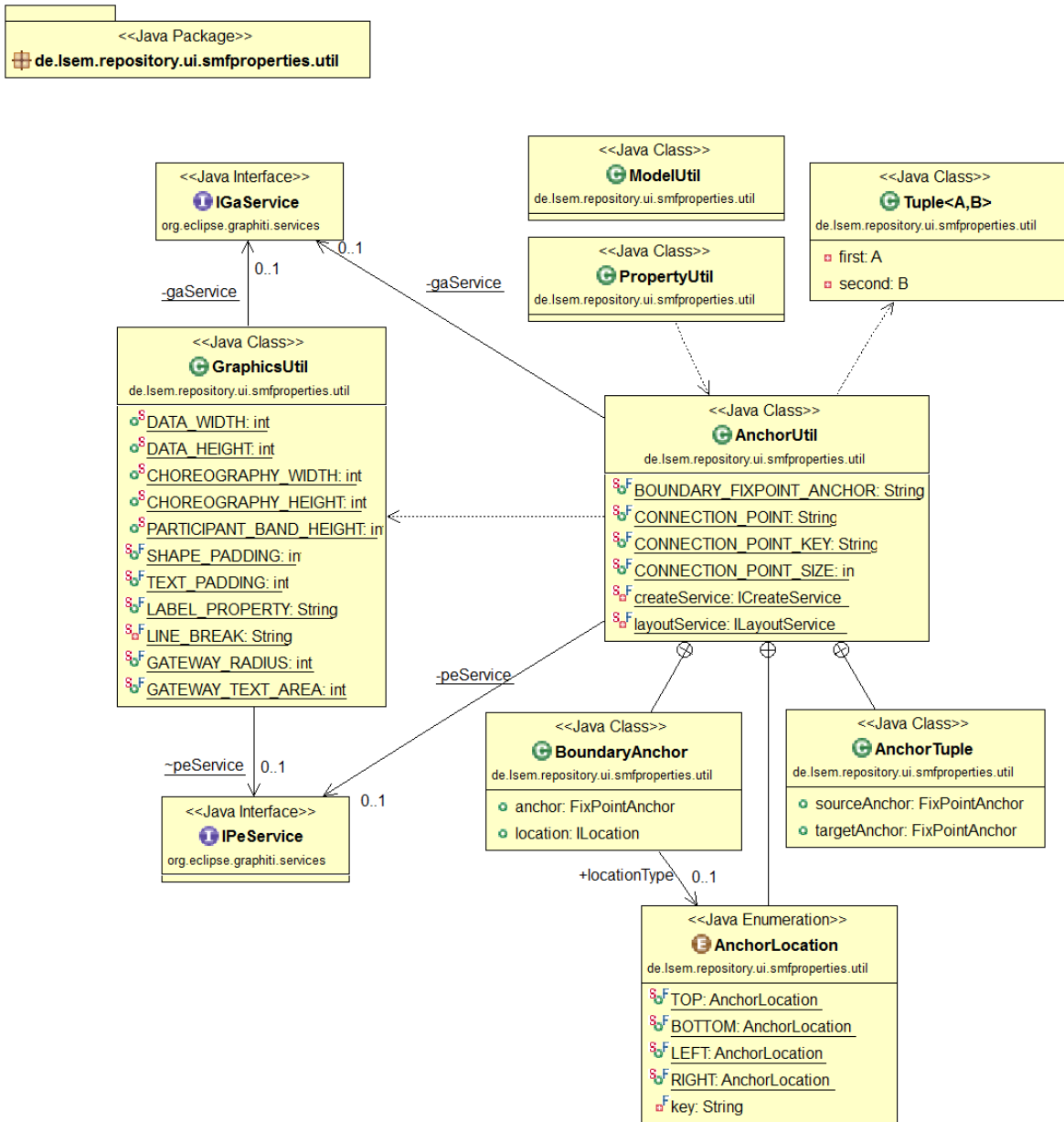
Anhang 3.4 UML - SMF-Explorer

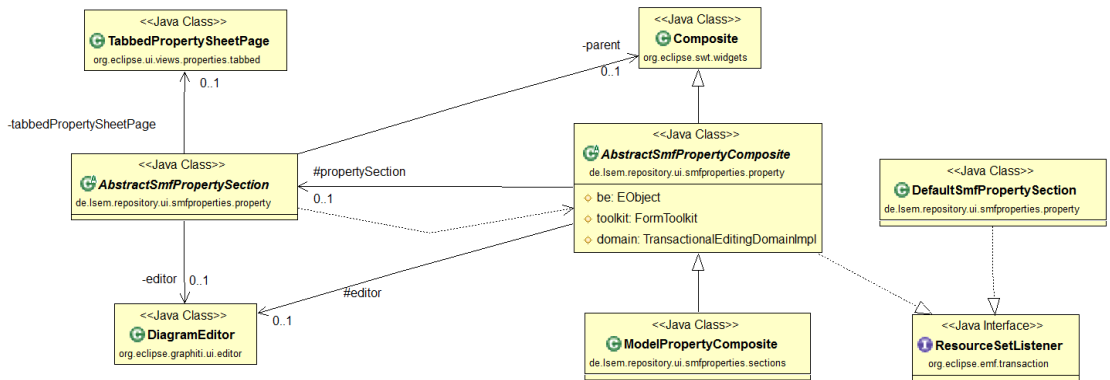
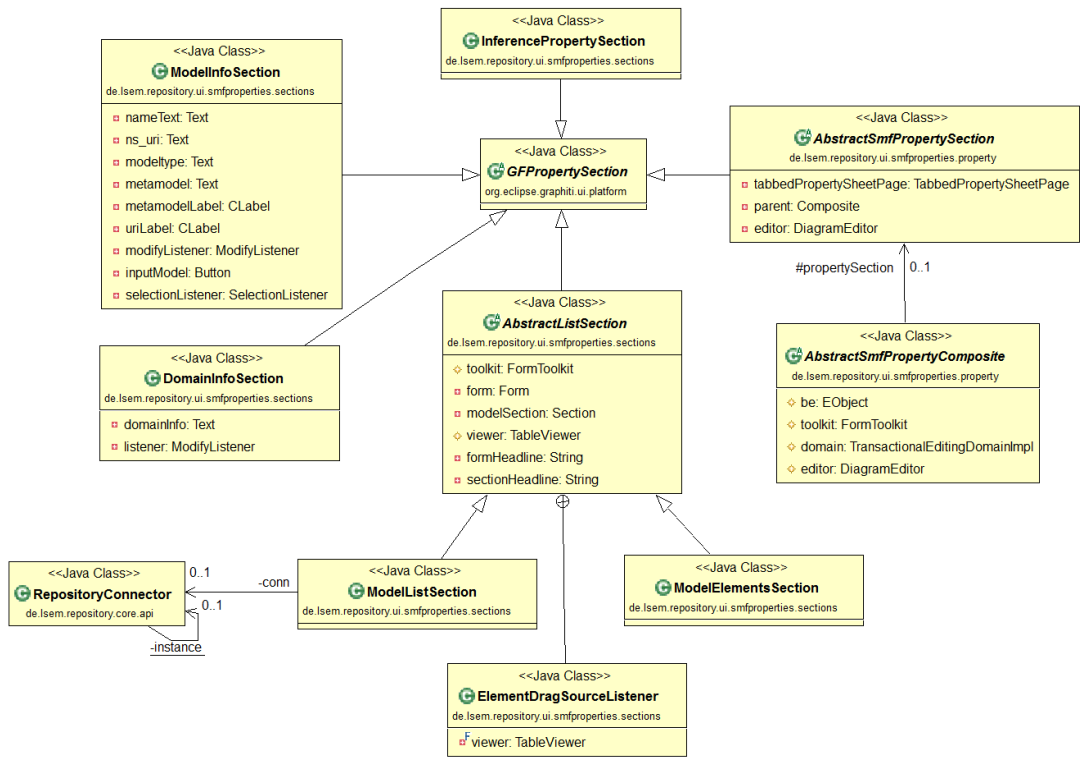




Anhang 3.5 UML - SMF-Properties







Selbständigkeitserklärung

Hiermit erkläre ich, dass

1. die vorgelegte Dissertation ohne unzulässige Hilfe, insbesondere ohne die Inanspruchnahme eines Promotionsberaters, und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde und dass die aus fremden Quellen direkt oder indirekt übernommenen Gedanken in der Arbeit als solche kenntlich gemacht worden sind und
2. die vorgelegte Dissertation weder im Inland noch im Ausland in gleicher oder in ähnlicher Form einer anderen Prüfungsbehörde zum Zwecke einer Promotion oder eines anderen Prüfungsverfahrens vorgelegt und insgesamt noch nicht veröffentlicht wurde.

Leipzig, den 6. Juni 2016

.....

(Unterschrift)