

**IT-gestütztes Service-Level-Management –
Anforderungen und Spezifikation einer
Managementarchitektur**

Dissertation

an der

**Fakultät für Mathematik, Informatik und Statistik
der
Ludwig-Maximilians-Universität München**

vorgelegt von

Thomas Schaaf

Tag der Einreichung: 17. November 2008

IT-gestütztes Service-Level-Management – Anforderungen und Spezifikation einer Managementarchitektur

Dissertation

an der

Fakultät für Mathematik, Informatik und Statistik
der
Ludwig-Maximilians-Universität München

vorgelegt von

Thomas Schaaf

Tag der Einreichung: 17. November 2008

Tag des Rigorosums: 17. Dezember 2008

1. Berichterstatter: **Prof. Dr. Heinz-Gerd Hegering**,
Ludwig-Maximilians-Universität München
2. Berichterstatter: **Prof. Dr. Johann Schlichter**,
Technische Universität München

Danksagung

Diese Dissertation ist das Ergebnis einer Reihe von Ideen und der Bereitschaft, nicht wenige davon nach kürzeren oder längeren Phasen der Reflexion wieder zu verwerfen. Dieser Prozess wurde begleitet, gefördert und unterstützt durch viele Menschen, deren Anteil an dieser Arbeit in diesen Zeilen Anerkennung finden soll.

Mein wichtigster und tief empfundener Dank geht an meinen Doktorvater, Professor Dr. Heinz-Gerd Hegering, für seine Anleitung zum wissenschaftlichen Arbeiten, die lebhaften Diskussionen und sein außergewöhnliches Engagement während der gesamten Zeit der Betreuung.

Auch meinem Zweitberichterstatter, Prof. Dr. Johann Schlichter, möchte ich für seine enorm schnellen Reaktionen und viele hilfreiche Anmerkungen sehr herzlich danken. Ich danke Prof. Dr. Alexander Knapp für viele praktische Hinweise zur Umsetzung wichtiger Teile dieser Arbeit sowie Prof. Dr. Christian Böhm und Prof. Dr. Dieter Kranzlmüller für ihre Unterstützung.

Meinen Kollegen möchte ich für die offene und herzliche Aufnahme in das MNM-Team danken. Die angenehme, vertrauensvolle und durch einen ganz besonderen Teamgeist geprägte Atmosphäre dieses Umfelds hat maßgeblich zum Gelingen dieser Arbeit beigetragen. Für nicht selten harte und kritische, aber stets konstruktive Diskussionen über Ideen und ihre Umsetzung möchte ich mich bei allen Kollegen und stellvertretend hierfür bei Michael Brenner („Hast du schon dein Zertifikat?“), Vitalian Danciu („Ist das eine Vektorgrafik?“) und Michael Schiffers („Das seh’ ich ganz leidenschaftslos.“) bedanken. Vielen Dank an Rob Kuhlig für kreative Diskussionen und zahlreiche Berichte aus der Praxis.

Im privaten Umfeld fordert der Entstehungsprozess einer Dissertation nicht selten Opfer von Familie und Freunden. Ich danke vor allem Katharina sowie meinen Eltern und Geschwistern für ihren immensen Rückhalt und ihr Verständnis.

München, im November 2008

Kurzfassung

Mit der weiter zunehmenden Komplexität vernetzter IT-Systeme und der Heterogenität der auf diesen Systemen realisierten IT-Dienste wachsen die Anforderungen an das Management moderner IT-Umgebungen stetig. Längst ist klar, dass diesen Managementtherausforderungen nicht allein mittels technischer Hilfsmittel auf Ressourcen- oder System-Ebene begegnet werden kann. Vielmehr müssen auch organisatorische Aspekte des dienstorientierten IT-Managements in ganzheitliche Lösungsansätze einfließen. Prozessorientierte Rahmenwerke, die in der Praxis zunehmend an Bedeutung gewinnen, ebnen den Weg dorthin. Auch ISO/IEC 20000 als internationaler Standard für *IT-Service-Management* gibt Mindestanforderungen an Prozesse zur Planung, Steuerung und Entstörung von IT-Diensten vor.

Ein praxis- wie forschungsrelevantes Problem aller prozessorientierten Management-Rahmenwerke stellt jedoch die Unterstützung der Prozesse und ihres Zusammenspiels durch geeignete IT-gestützte Werkzeuge (*Tools*) dar. Die meisten Ansätze liefern in diesem Punkt kaum Hilfestellung. Während in den eher operativen Bereichen wie dem Störungsmanagement aufgrund des eindeutigen Aufgabenfokus und klar definierter Prozessstrukturen bereits Lösungsansätze entwickelt werden konnten, bestehen in anderen Disziplinen noch große Lücken. Besonders ausgeprägt ist diese Lückenhaftigkeit in einer der Kerndisziplinen des IT-Service-Managements, dem *Service-Level-Management* (SLM). Mit dem Ziel, IT-Dienste in der vom Kunden beanspruchten Qualität zu liefern, umfasst Service-Level-Management vielfältige Aufgaben wie den Abschluss von Dienstvereinbarungen (*Service Level Agreements*, SLAs) zwischen Dienstanbieter und Dienstnehmer, die Spezifikation der zu erbringenden Dienstfunktionalität und -qualität sowie die Definition, Messung und Berichterstattung über Leistungsparameter, um nur die bekanntesten Beispiele zu nennen.

In diesen Kontext ordnet sich die vorliegende Arbeit ein. Es wird eine Architektur für IT-gestütztes Service-Level-Management vorgestellt, die das Ziel verfolgt, den Entwurf eines Werkzeug-orientierten SLM-Managementsystems zu unterstützen. Ein Ordnungsrahmen aus verschiedenen architekturellen Teilmodellen bestimmt den modularen Aufbau des Lösungsansatzes, der – analog zu allgemeinen Managementarchitekturen – aus einem Organisations-, einem Funktions-, einem Informations- und einem Kommunikationsmodell besteht. Jedes dieser Teilmodelle wird aus zwei unterschiedlichen Sichten, der Prozess- und der Systemsicht, beleuchtet. Als weiterer methodischer Baustein wird ein Entwicklungsrahmen zugrunde gelegt, der festlegt, wie auf Basis realer

IT-Szenarien funktionale Anforderungen an die SLM-Architektur identifiziert werden und wie diese Anforderungen in die einzelnen Teilmodelle einfließen.

Das Ergebnis der Arbeit kann als ein generischer, modellbasierter, wiederverwendbarer und erweiterbarer „Baukasten“ für den Entwurf eines IT-gestützten Managementsystems zur Unterstützung anfallender Aufgaben im Service-Level-Management angesehen werden. Dieser Baukasten besteht aus konkreten funktionalen Anforderungen, Anwendungsfällen für ein Managementsystem, Prozess-bezogenen Modellen, Systemmodellen mit Bezug auf Softwareentwicklungs- und Automatisierungsaspekte sowie praktischen Implementierungshinweisen.

Abstract

As a result of the increasing complexity of today's networked IT systems and the heterogeneity of IT services running on these systems, requirements on managing modern IT environments are becoming more and more sophisticated. It is generally well understood that facing these challenges by means of assistive technology on the resource and system layer only is not sufficient. A holistic and integrated approach rather has to consider organizational aspects of service-oriented IT management, too. Process-oriented frameworks that have gained soaring attention in practice recently pave the way towards this goal. As an international standard for *IT Service Management*, ISO/IEC 20000 states minimum requirements on processes in the areas of planning, control and fault clearance of IT services.

A problem which all process-oriented management frameworks have in common and that is relevant not only from a researcher's, but also from a practitioner's perspective, is given by the question of adequate computer-aided tool support for the processes enclosed. Most frameworks hardly provide any guidance with respect to this issue. While – due to unambiguous objectives and well-structured process workflows – various tools have already been developed in the more operational field of service management, including for example incident handling, other fields are still suffering from major gaps concerning tool support. This situation also applies to *Service Level Management (SLM)* which is one of the core disciplines of IT Service Management. Aiming at providing IT services to customers in the quality required by those customers, Service Level Management encompasses a multifaceted scope of functions including the closure of service level agreements (SLAs), the specification of the service functionality and quality as well as the definition, measurement and reporting of performance parameters.

In that context, this work presents an architecture for IT-supported Service Level Management, aiming at supporting the design of a tool-based management system for SLM. A static classification framework realizes a highly modular design of the solution. Analogous to generic management architectures, it consists of an organizational, a functional, an information and a communication model. Each of these partial models is examined from two different view points: the process view and the system view. As an additional methodical building block, a development framework describes how functional requirements on the SLM architecture can be identified and how these requirements lead to the architecture's core elements.

The outcome of this work can be regarded as a generic, model-based, reusable

and extensible „construction kit“ for the design of a computer-aided management system in support of incidental tasks in Service Level Management. This construction kit consists of concrete functional requirements, use cases for a management system, process-related models, system models with regard to software development and automation aspects as well as practical implementation guidance.

Inhaltsverzeichnis

1	Einleitung und Motivation	1
1.1	Service-Level-Management	3
1.2	Einordnung und Probleme bestehender Lösungen	4
1.3	Haupt- und Teilfragestellungen dieser Arbeit	7
1.4	Vorgehen	9
1.4.1	Methodische Herangehensweise	10
1.4.2	Vorgehensmodell	11
1.4.3	Schwerpunkte	11
1.4.4	Zusammenhänge der Teilergebnisse	15
1.5	Einordnung und Abgrenzung	16
1.5.1	Arbeiten aus dem MNM-Team	16
1.5.2	Forschungsprojekte	19
1.6	Angestrebte Ergebnisse dieser Arbeit	20
2	Methodisches Vorgehen und Begriffsbildung	23
2.1	Methodische Grundlagen	24
2.1.1	Modellgetriebene Architektur	25
2.1.2	Prozessorientierung	25
2.1.3	Systembasierter Ansatz	27
2.1.4	Zusammenfassung: Methodische Grundlagen	31
2.2	Systematik des Architekturentwurfs	32
2.2.1	Ordnungsrahmen	32
2.2.2	Entwicklungsrahmen	35
2.3	Begriffsbildung	38
2.3.1	Der Dienstbegriff	38
2.3.2	Begriffe der Providerdomäne	45
2.3.3	Begriffe der Kundendomäne	46
2.3.4	Begriffe auf der Schnittstelle zwischen Provider und Kunde	47

2.4	Zusammenfassung	48
3	Anforderungsanalyse	51
3.1	Szenarien	53
3.1.1	Morphologischer Kasten zur Einordnung der Szenarien .	54
3.1.2	Szenario 1: Transportdienstleister (ITG)	59
3.1.3	Szenario 2: Hochschulrechenzentrum (LRZ)	63
3.1.4	Szenario 3: Flughafen (FMG)	66
3.1.5	Vollständigkeit der Szenarien	70
3.2	Funktionale Anforderungen	72
3.2.1	SLM-Zielsetzung und Grundannahmen	74
3.2.2	Definition eines generischen Dienstlebenszyklus	76
3.2.3	Funktionale Anforderungen an Service-Level-Management	78
3.2.4	Auswertung der allgemeinen Funktionsanforderungen . .	96
3.2.5	Zusammenfassung und nächste Schritte	99
3.3	Primäranwendungsfälle	99
3.3.1	Definition der Anwendungsfälle	101
3.3.2	Zusammenhänge zwischen den Anwendungsfällen	117
3.4	Nicht-funktionale Anforderungen	127
3.5	Zusammenfassung und nächste Schritte	130
4	Umfeld und verwandte Arbeiten	131
4.1	Einführung, Überblick und Bewertungsschema	132
4.1.1	Überblick	132
4.1.2	Analyse- und Bewertungsschema	135
4.2	Grundlegende Arbeiten	135
4.2.1	Dienstmanagement-Rahmenwerk (Dreo)	135
4.2.2	SLM für Unternehmensnetze (Lewis)	139
4.3	Prozessrahmenwerke	141
4.3.1	IT Infrastructure Library (OGC)	141
4.3.2	ISO/IEC 20000	145
4.3.3	Enhanced Telecom Operations Map (TMF)	147
4.4	Modellbasierte Ansätze	148
4.4.1	Shared Information/Data Model (TMF)	148
4.4.2	Werkzeugunterstützung für ITIL-orientiertes Dienstma- nagement (Brenner)	154
4.4.3	Modellbasiertes SLM (Debusmann)	155
4.5	Sonstige Ansätze	156
4.5.1	NGOSS SLA-Management-Handbuch (TMF)	156
4.5.2	SLA-Entwurf auf Basis von Dienstprozessen (Schmidt) .	162
4.5.3	Weitere Veröffentlichungen im Kurzüberblick	164

4.6	Zusammenfassung und abschließende Bewertung	165
5	Prozess-Sicht der Managementarchitektur	167
5.1	Organisationsmodell	170
5.1.1	Organisationsdomänen	170
5.1.2	Rollen	171
5.1.3	Verantwortungsmodelle	174
5.1.4	Zusammenfassung	176
5.2	Informationsmodell	177
5.2.1	Grundlagen der Prozess-Informationsmodellierung . . .	177
5.2.2	Identifikation SLM-spezifischer Konzepte	181
5.2.3	Konzepterweiterungen	186
5.2.4	Ableitung und Klassifizierung von Informationsartefakten	192
5.2.5	Zusammenfassung	214
5.3	Funktionsmodell	216
5.3.1	Festlegung der Funktionsbereiche	216
5.3.2	Funktionsbereich I: Dienstinformationsmanagement . .	221
5.3.3	Funktionsbereich II: Stakeholder-Management	237
5.3.4	Funktionsbereich III: Vereinbarungsmanagement	242
5.3.5	Funktionsbereich IV: Messdaten- und Berichtsmanage- ment	256
5.3.6	Zusammenfassung	264
5.4	Kommunikationsmodell	266
5.4.1	Intraprozess-Kommunikationsmodell	267
5.4.2	Interprozess-Kommunikationsmodell	282
5.4.3	Zusammenfassung	294
5.5	Zusammenfassung und nächste Schritte	296
5.5.1	Ergebnisse der Prozess-Sicht	296
5.5.2	Abgrenzung	297
6	System-Sicht der Managementarchitektur	299
6.1	Organisationsmodell	301
6.1.1	Systembenutzerrollen	302
6.1.2	Nutzerschnittstellen und Systeminteraktivität	304
6.1.3	Zugriffskontrolle	308
6.1.4	Zusammenfassung	312
6.2	Informationsmodell	313
6.2.1	Auswahl einer Modellierungssprache	314
6.2.2	Methodisches Vorgehen	315
6.2.3	Datenmodellierung	319
6.2.4	Zusammenfassung	330

6.3	Funktionsmodell	332
6.3.1	Methodischer Ansatz	333
6.3.2	Automatisierung vom Managementfunktionen	338
6.3.3	Zusammenfassung	347
6.4	Kommunikationsmodell	348
6.4.1	Realisierung der Intraprozesskommunikation	349
6.4.2	Realisierung der Interprozesskommunikation	352
6.4.3	Zusammenfassung	356
7	Plattformentwicklung und Implementierungsaspekte	357
7.1	Realisierung einer Managementplattform	358
7.1.1	Komponenten einer Managementplattform	359
7.1.2	Schritte zum Plattformentwurf	361
7.2	Exemplarische Instanziierung des Informationsmodells	364
7.2.1	Umsetzung	364
7.2.2	Fazit	368
8	Zusammenfassung und Ausblick	369
8.1	Zusammenfassung	369
8.1.1	Ausgangspunkt der Arbeit	369
8.1.2	Methodisches Vorgehen und wesentliche Ergebnisse	371
8.1.3	Offene Punkte	373
8.2	Ausblick	374
A	SLM-Informationssystem und Datenmodelle	377
A.1	ServiceLevelManagementInformationSystem	377
A.2	CommonTypes	382
A.3	ServicePortfolioDatabase	382
A.4	CustomerDatabase	385
A.5	ServiceDeliveryPartyDatabase	386
A.6	SupportLevelProfileDatabase	388
A.7	ServiceLevelAgreementDatabase	390
A.8	SupportiveAgreementDatabase	392
A.9	MeasurementDatabase	394
A.10	ReportDatabase	395
A.11	ConflictDatabase	398
B	Exemplarische Instanziierung des SLM-Informationssystems	399
C	Literaturverzeichnis	405

Abbildungsverzeichnis

1.1	IT-Service-Management: Umfeld und Beziehungen	2
1.2	Einordnung bestehender Ansätze zur Realisierung von Managementlösungen in das Klassifizierungsschema nach Hegering et al. [1]	5
1.3	Die Teilmodelle einer Managementarchitektur	8
1.4	Vorgehensmodell	12
1.5	Zusammenhänge der Teilergebnisse	15
2.1	Drei Säulen des methodischen Vorgehens	24
2.2	Top-Down-Vorgehen dieser Arbeit im Kontext des MDA-Ansatzes	26
2.3	OSI-Management: Management durch MIB-Zugriff	33
2.4	Zielsetzungen der Teilmodelle für Architekturen im Dienstmanagement	34
2.5	Ordnungsrahmen der Architektur	35
2.6	Entwicklungsrahmen, Teil 1: Systematik der Anforderungsanalyse	37
2.7	Entwicklungsrahmen, Teil 2: Zusammenhänge der Teilmodelle .	39
2.8	Wert-Dimensionen eines IT-Dienstes nach ITIL [2]	41
2.9	Das MNM-Dienstmodell	42
3.1	Einordnung dieses Kapitels in den Ordnungsrahmen	52
3.2	Abgrenzung der funktionalen und nicht-funktionalen Anforderungen	53
3.3	Morphologischer Kasten zur Einordnung der Szenarien	54
3.4	Organigramm der Internationalen Transportgesellschaft (ITG) (Quelle: EXIN [3])	60
3.5	Morphologie des ITG-Szenarios	62
3.6	Morphologie des LRZ-Szenarios	65
3.7	Einordnung des FMG-Szenarios in den Morphologischen Kasten	69

3.8	Zusammenfassende morphologische Analyse aller Szenarien . . .	71
3.9	Typischer Dienstlebenszyklus nach Dreo [4]	77
3.10	Einordnung des ermittelten Funktionsspektrums	97
3.11	Das Managementsystem als <i>Black Box</i>	100
3.12	Überblick über die Primäranwendungsfälle 1 bis 29 und ihre wechselseitigen Abhängigkeiten	118
3.13	Überblick über die Primäranwendungsfälle 30 bis 38 und ihre wechselseitigen Abhängigkeiten	119
4.1	Kernbestandteile des Dienstmanagement-Rahmenwerks von Dreo	137
4.2	Anwendung von Dienst-Template-Modellen nach Dreo [4] . . .	138
5.1	Einordnung dieses Kapitels in den Ordnungsrahmen	168
5.2	Domänen- und Rollenmodell im Überblick	175
5.3	Abgrenzung zwischen Dienstmanagement und Prozessmanage- ment im Hinblick auf relevante Informationsartefakte	179
5.4	UML-Konzeptdiagramm für die Disziplin Service-Level- Management, Teil 1	182
5.5	UML-Konzeptdiagramm für die Disziplin Service-Level- Management, Teil 2	185
5.6	Konzeptionelle Erweiterung um Multi-Level-SLAs	188
5.7	Konzeptionelle Erweiterung um explizite Diensthierarchien . . .	190
5.8	Konzeptionelle Erweiterung um Dienstmodule	191
5.9	Partielles Informationsmodell für die Konzepte <i>Dienst</i> und <i>Dienstkatalog</i>	193
5.10	Partielles Informationsmodell für das Konzept <i>Kunde</i>	197
5.11	Partielles Informationsmodell für das Konzept <i>ServiceDelivery- Party</i>	199
5.12	Partielles Informationsmodell für das Konzept <i>Support-Level- Profil</i>	201
5.13	Partielles Informationsmodell für das Konzept <i>ServiceLevel- Agreement</i>	203
5.14	Partielles Informationsmodell für das Konzept <i>Unterstützende Vereinbarung</i>	206
5.15	Partielles Informationsmodell für das Konzept <i>Dienstmessung</i> .	208
5.16	Partielles Informationsmodell für das Konzept <i>Dienstbericht</i> , Teil 1	210
5.17	Partielles Informationsmodell für das Konzept <i>Dienstbericht</i> , Teil 2	212
5.18	Partielles Informationsmodell für das Konzept <i>Benachrichti- gung über eine SLA-Verletzung</i>	213

5.19	Partielles Informationsmodell für das Konzept <i>Vereinbarungskonflikt</i>	214
5.20	Grundlegende Funktionsbereiche des Service-Level-Managements und ihre Einordnung in das Funktionsspektrum .	218
5.21	Darstellung der Funktionsbereiche und ihrer Teilbereiche sowie schematische Darstellung des weiteren Top-Down-Vorgehens im Rahmen des Funktionsmodells	219
5.22	Referenz-Workflow 1 (<i>Create new service</i>)	222
5.23	Referenz-Workflow 2 (<i>Change service</i>)	226
5.24	Referenz-Workflow 3 (<i>Delete service</i>)	228
5.25	Referenz-Workflow 7 (<i>Initialize service catalog</i>)	230
5.26	Referenz-Workflow 8 (<i>Assign catalog to customer</i>)	231
5.27	Referenz-Workflow 9 (<i>Dissociate catalog from customer</i>)	232
5.28	Referenz-Workflow 10 (<i>Delete service catalog</i>)	233
5.29	Referenz-Workflow 11 (<i>Add service to catalog</i>)	233
5.30	Referenz-Workflow 12 (<i>Remove service from catalog</i>)	234
5.31	Referenz-Workflow 13 (<i>Create support level profile</i>)	235
5.32	Referenz-Workflow 14 (<i>Change support level profile</i>)	236
5.33	Referenz-Workflow 15 (<i>Delete support level profile</i>)	237
5.34	Referenz-Workflow 4 (<i>Create new customer</i>)	238
5.35	Referenz-Workflow 5 (<i>Change customer</i>)	239
5.36	Referenz-Workflow 6 (<i>Delete customer</i>)	239
5.37	Referenz-Workflow 16 (<i>Create delivery party</i>)	241
5.38	Referenz-Workflow 17 (<i>Change delivery party data</i>)	241
5.39	Referenz-Workflow 18 (<i>Delete delivery party</i>)	242
5.40	Referenz-Workflow 19 (<i>Close SLA</i>)	244
5.41	Referenz-Workflow 20 (<i>Revise SLA</i>)	247
5.42	Referenz-Workflow 21 (<i>Resign SLA</i>)	248
5.43	Referenz-Workflow 22 (<i>Close OLA</i>)	249
5.44	Referenz-Workflow 23 (<i>Revise OLA</i>)	250
5.45	Referenz-Workflow 24 (<i>Resign OLA</i>)	251
5.46	Referenz-Workflow 25 (<i>Close UC</i>)	252
5.47	Referenz-Workflow 26 (<i>Revise UC</i>)	253
5.48	Referenz-Workflow 27 (<i>Resign UC</i>)	254
5.49	Referenz-Workflow 28 (<i>Analyze agreements</i>)	255
5.50	Referenz-Workflow 29 (<i>Measure and record</i>)	258
5.51	Referenz-Workflow 30 (<i>Create internal report</i>)	260
5.52	Referenz-Workflow 31 (<i>Create external report</i>)	262
5.53	Referenz-Workflow 32 (<i>Create SLA violation notification</i>)	263
5.54	Abgrenzung von Prozess- und System-Sicht der Managementarchitektur	297

6.1	Einordnung dieses Kapitels in den Ordnungsrahmen	300
6.2	Visualisierung des Datenmodells zum ServiceLevelManagement-InformationSystem (Auszug)	321
6.3	Visualisierung des Datenmodells zum ServiceLevelManagement-InformationSystem (Auszug)	322
6.4	Visualisierung des Datenmodells zum Informationsartefakt ServicePortfolio (Auszug)	324
6.5	Visualisierung des Datenmodells zum Informationsartefakt Customer (Auszug)	325
6.6	Visualisierung der Typdefinition zum Informationsartefakt ServiceCatalog (Auszug)	326
6.7	Visualisierung der Typdefinition zum Informationsartefakt ServiceLevelAgreement (Auszug)	327
6.8	Visualisierung der Typdefinition zum Informationsartefakt MeasurementSpecification (Auszug)	328
6.9	Visualisierung der Typdefinition zum Informationsartefakt ExternalReportSpecification (Auszug)	329
6.10	Visualisierung der Typdefinition zum Informationsartefakt ExternalReport (Auszug)	331
6.11	Stufenschema zur Festlegung des Automatisierungsgrads	335
6.12	Sub-Workflow für die Aktivität <i>Create service catalog</i>	345
6.13	Schematische Darstellung für das Kommunikationsmuster <i>Signalisierung</i>	349
6.14	Schematische Darstellung für das Kommunikationsmuster <i>Informationsübermittlung</i>	350
6.15	Schematische Darstellung für das Kommunikationsmuster <i>Anfrage-Antwort-Kommunikation</i>	351
6.16	Schematische Darstellung für das Kommunikationsmuster <i>Standardisierter Dialog</i>	352
6.17	Ansatz für die Kopplung von Managementsystemen unter Kommunikations Gesichtspunkten	353
6.18	Analyse von IMM, CIM und SID auf ihre Eignung als Dienstmanagement-Informationsmodell	355
7.1	Einordnung dieses Kapitels in den Ordnungsrahmen	358
7.2	Aufbau von Managementplattformen nach Hegering et al. [1]	360
7.3	Schritte zum Entwurf einer Plattform für Service-Level-Management	362
7.4	Visualisierung der exemplarischen Instanz des Informationsmodells (Auszug, Teil 1)	365

7.5	Visualisierung der exemplarischen Instanz des Informationsmodells (Auszug, Teil 2)	366
7.6	Visualisierung der exemplarischen Instanz des Informationsmodells (Auszug, Teil 3)	367

Tabellenverzeichnis

2.1	Methodische Konzepte und Begriffe im Überblick	31
3.1	Zuordnung der funktionalen Anforderungen zu den Szenarien .	98
3.2	Zuordnung zwischen funktionalen Anforderungen und Primäranwendungsfällen	101
3.3	Beschreibung der Abhängigkeiten zwischen den Anwendungsfällen	127
4.1	Überblick der in diesem Kapitel vorgestellten Arbeiten und ihren Bezug zu dieser Dissertation	134
4.2	Schema und Symbolik zur Analyse und Bewertung des Beitrags zur Architekturentwicklung durch einen Lösungsansatz	136
4.3	Bewertung des Lösungspotenzials des Dienstmanagement-Rahmenwerks von Dreo im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	139
4.4	Bewertung des Lösungspotenzials der SLM-Methodik und -Architektur von Lewis im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	141
4.5	Überblick über die zentralen Bestandteile von ITIL Version 2 .	143
4.6	Überblick über die zentralen Bestandteile von ITIL Version 3 .	144
4.7	Bewertung des Lösungspotenzials von ITIL Version 3 im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	146
4.8	Bestandteile des Standards ISO/IEC 20000	146
4.9	Überblick über die zentralen Dokumente der eTOM	148
4.10	Bewertung des Lösungspotenzials von eTOM im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	149
4.11	Überblick über die Domänen-Addenda des SID	153
4.12	Bewertung des Lösungspotenzials von SID im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	154

4.13	Bewertung des Lösungspotenzials des SLA-Management-Handbuchs im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	163
4.14	Bewertung des Lösungspotenzials des Ansatzes von Schmidt im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit	164
5.1	Informationsanforderungen an das Artefakt Service	194
5.2	Informationsanforderungen an das Artefakt ServiceCatalog . . .	197
5.3	Informationsanforderungen an das Artefakt Customer	198
5.4	Informationsanforderungen an das Artefakt ServiceDeliveryParty	200
5.5	Informationsanforderungen an das Artefakt SupportLevelProfile	202
5.6	Informationsanforderungen an das Artefakt ServiceLevelAgreement	204
5.7	Informationsanforderungen an das Artefakt SupportiveAgreement	206
5.8	Informationsanforderungen an das Artefakt ServiceMeasurementSpecification	208
5.9	Informationsanforderungen an das Artefakt ServiceMeasurementResults	209
5.10	Informationsanforderungen an das Artefakt ReportSpecification	211
5.11	Informationsanforderungen an das Artefakt ServiceReport . . .	211
5.12	Informationsanforderungen an das Artefakt SLAViolationNotification	213
5.13	Informationsanforderungen an das Artefakt AgreementConflict .	214
5.14	Festlegung von Kommunikationsmustern mit Hilfe der Differenzierungskriterien	269
5.15	Übersicht über alle Intraprozess-Kommunikationsbeziehungen .	281
5.16	Grundlegendes Referenz-Prozessrahmenwerk in Anlehnung an ISO/IEC 20000	284
5.17	Abgrenzung: Prozess-Sicht vs. System-Sicht der Architektur . .	298
6.1	Interaktivitätsanalyse	307
6.2	Überblick über alle SLM-spezifischen Informationsartefakte . .	330
6.3	Auszug aus der MABA-MABA-Liste nach Fitts [5]	337
6.4	Automatisierbarkeitsanalyse aller Funktionsbereiche	343

Kapitel 1

Einleitung und Motivation

Inhalt dieses Kapitels

1.1	Service-Level-Management	3
1.2	Einordnung und Probleme bestehender Lösungen	4
1.3	Haupt- und Teilfragestellungen dieser Arbeit .	7
1.4	Vorgehen	9
1.4.1	Methodische Herangehensweise	10
1.4.2	Vorgehensmodell	11
1.4.3	Schwerpunkte	11
1.4.4	Zusammenhänge der Teilergebnisse	15
1.5	Einordnung und Abgrenzung	16
1.5.1	Arbeiten aus dem MNM-Team	16
1.5.2	Forschungsprojekte	19
1.6	Angestrebte Ergebnisse dieser Arbeit	20

Aufgrund der nach wie vor zunehmenden Komplexität moderner IT-Systeme spielen Managementfragestellungen in der Informatik eine immer bedeutendere Rolle. Obwohl diese Entwicklung nicht neu ist, bringt sie stets neue Konzepte unterschiedlicher Ausrichtungen hervor. So gewinnt das Paradigma des dienstorientierten IT-Managements seit Anfang der 1990er-Jahre immer mehr an Bedeutung, dessen Grundidee darin besteht, nunmehr ganze Dienste statt einzelner Komponenten zu managen. Dieses Prinzip führte zu einer deutlich weniger technologiezentrierten Sichtweise auf das Management verteilter Anwendungen, Systeme und Netze als bisherige Ansätze und begründete den Trend des *IT-Service-Managements* (ITSM) .

Obwohl es nicht explizit durch diesen Begriff beschrieben wird, wird IT-Service-Management vielfach auch mit der Ausrichtung von IT-Management-

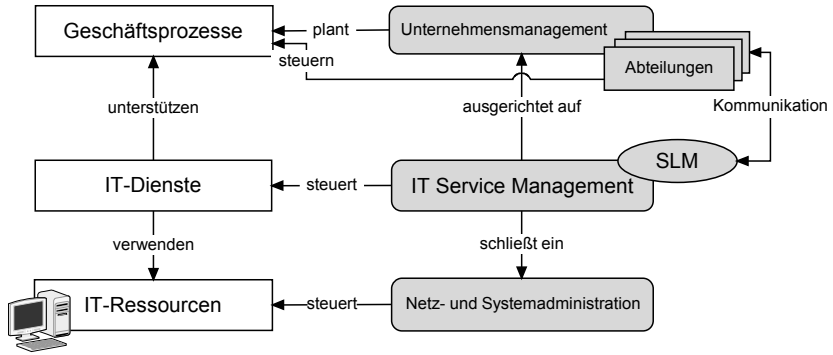


Abbildung 1.1: IT-Service-Management: Umfeld und Beziehungen

Aktivitäten an den Geschäftszielen eines Unternehmens (im Englischen: *Business IT Alignment* oder auch *Business-driven IT Management*) in Verbindung gebracht. Darunter versteht man ganz allgemein Maßnahmen, die dazu beitragen, Ziele, Leistungen und Prozesse zwischen einem Anbieter von IT-Diensten (IT-Organisation, Provider) und direkt an der Wertschöpfung beteiligten Parteien aufeinander abzustimmen. Hintergrund ist die enge Verbindung zwischen den Wertschöpfungsprozessen (Geschäftsprozessen) eines Unternehmens und den IT-Diensten, die eben diese Wertschöpfung unterstützen sollen.

Der Gedanke des *Business IT Alignment* bedingt damit auch ein Umdenken innerhalb der IT-Provider, die sich zunehmend in der Rolle von kundenorientierten Dienstleistern sehen. Im Falle einer internen IT-Organisation sind die Kunden des IT-Providers in der Regel die Fachabteilungen des Unternehmens. Abbildung 1.1 illustriert diese Zusammenhänge. Hier wird auch das Spannungsfeld deutlich, in dem sich *IT-Service-Management* befindet: als Disziplin zwischen dem klassischen technischen Netz- und Systemmanagement und dem betriebswirtschaftlichen Unternehmensmanagement (*Enterprise Management*), fokussiert auf das Ziel, IT-Dienste in einer Weise zu steuern, dass die verfügbaren Technologien in effizienter Weise zur Unterstützung der wertschöpfenden Geschäftsprozesse eingesetzt werden können.

Aus diesem neuen Verständnis der Positionierung von IT innerhalb eines Unternehmens ist schließlich auch die Disziplin *Service-Level-Management* (SLM) entstanden, die im Fokus der vorliegenden Arbeit steht und im folgenden Abschnitt einführend beschrieben wird. Anschließend wird in diesem Kapitel herausgearbeitet, welche Probleme die derzeitigen Ansätze zur Unterstützung der

Aktivitäten im Service-Level-Management mit sich bringen und welcher Lösungsansatz mit dieser Arbeit verfolgt wird.

1.1 Service-Level-Management

Service-Level-Management ist eine wichtige Teildisziplin im dienstorientierten IT-Management, deren grundlegende Ziele, Prinzipien und Herausforderungen von Lewis et al. [6, 7] erstmals umfassend beschrieben wurden. Im Allgemeinen fasst man unter dem Begriff des Service-Level-Managements alle Managementaktivitäten in einer IT-Organisation zusammen, die ein IT-Provider unternimmt, um seine Dienste gemäß ihrer funktionalen Spezifikation zu einem gegebenenfalls kundenspezifisch festgelegten Qualitätsniveau anbieten zu können. In geschäftsmäßigen Provider-Kunden-Beziehungen, wie zum Beispiel im *Business-to-Business*-Bereich, schließt ein vereinbartes Dienstniveau für einen oder mehrere Dienste oftmals Qualitätsgarantien ein, deren Nichteinhaltung im schlimmsten Fall zu erheblichen finanziellen Verlusten (Strafzahlungen) seitens des Dienstansbieters führen können. Auf der anderen Seite versetzt ein effektiv und kundenorientiert betriebenes Service-Level-Management einen Provider in die Lage, sich zusätzlich zur Ausgestaltung seiner Dienste und Produkte auch über die buchbaren Dienstniveaus mit den verbundenen Qualitätsgarantien von seinen Wettbewerbern am Markt abzuheben.

Dieser kurze Überblick macht bereits deutlich, welche entscheidende Rolle das Service-Level-Management gerade in einem kunden- und geschäftsorientierten Dienstmanagement-Umfeld einnimmt. Dass Service-Level-Management dabei genau auf der Schnittstelle zwischen dem Kunden mit seinen IT-unterstützten geschäftlichen Aktivitäten und der Provider-Domäne liegt, macht es erforderlich, dass einerseits das oben erwähnte Prinzip der IT-Ausrichtung an den Geschäftszielen Anwendung findet und dass ein ausgeprägtes Verständnis der Kundenbedürfnisse vorhanden ist. Andererseits darf die technische Perspektive nicht aus dem Auge verloren werden – ein Trend, der heute in vielen Ansätzen des IT-Service-Managements zu beobachten ist. Denn die zur Realisierung von IT-Diensten verfügbaren Technologien bestimmen, welches Dienstniveau überhaupt zu welchen Kosten bereitgestellt werden kann. Somit steht und fällt die Belastbarkeit von Qualitätsgarantien mit dem Verständnis der Technologien, die hierfür die Grundlage bilden.

Im engeren Sinne geht es im Zusammenhang mit der Disziplin des Service-Level-Managements primär um den Prozess der Aushandlung sowie um alle direkten Maßnahmen zur Überwachung und Erfüllung von *Dienstvereinbarungen* (*Service Level Agreements*, SLAs) zwischen einem Dienstanbieter und seinen

Kunden. Das beinhaltet die Identifikation der Kundenbedürfnisse hinsichtlich Dienstfunktionalität, Dienstqualität, Bestimmung messbarer Leistungsparameter, Feststellung der realisierbaren Dienstgüte, Definition von Weiterverrechnungsschlüsseln, Formulierung der Verträge sowie Überwachung (*Monitoring*) und Berichtswesen (*Reporting*) über die tatsächlich erreichte Dienstqualität. Weiterhin umfasst Service-Level-Management auch die Spezifikation der den Vereinbarungen zugrunde liegenden Dienste im Rahmen des Aufbaus eines *Dienstportfolios*, welches zum Ziel hat, alle angebotenen Dienste des Providers formal und mit allen managementrelevanten Parametern zu beschreiben. *Dienstkataloge* realisieren in diesem Zusammenhang eine Kundensicht auf das Dienstportfolio.

Obwohl allgemein als eine der wichtigsten und kritischsten Disziplinen des dienstorientierten IT-Managements erkannt und bewertet, existieren derzeit nur wenige Werkzeuge (Tools) zur effektiven, rechnergestützten Durchführung der anfallenden Aufgaben und Aktivitäten im Service-Level-Management. Zurückzuführen ist dieser Umstand nicht zuletzt auf die unklaren funktionalen Detailanforderungen und resultierenden Anwendungsfälle für ein entsprechendes *Managementsystem*.

Die vorliegende Arbeit beschäftigt sich daher mit Bausteinen zur Entwicklung IT-gestützter Managementlösungen im Bereich Service-Level-Management. Dazu wird im nächsten Abschnitt zunächst ein Überblick über einige generelle Fragestellungen aus dem Bereich des Service-Level-Managements aus Sicht der Informatik und des Dienstmanagements gegeben und erläutert, inwiefern die bestehenden Ansätze in diesem Zusammenhang Beiträge liefern und wo sie noch Lücken offen lassen. Dies bildet die Motivation für den in dieser Arbeit zu entwickelnden integrierten und werkzeugorientierten Ansatz eines Management-Rahmenwerks im Stil einer Managementarchitektur.

1.2 Einordnung und Probleme bestehender Lösungen

In der Disziplin des Service-Level-Managements existieren heute einige Ansätze, die versuchen, anfallende Aufgaben zu unterstützen. Dies kann prinzipiell einerseits auf einer eher technischen Ebene durch geeignete Managementwerkzeuge geschehen oder aber auf organisatorischer Ebene durch Rahmenwerke, die Empfehlungen zur Ausgestaltung des Service-Level-Managements beschreiben. Zumeist sind solche Empfehlungen eingebettet in einen oder mehrere Prozesse, weswegen die maßgeblichen Rahmenwerke in diesem Zusammenhang auch unter dem Paradigma der *Prozessorientierung* auftreten. Kapitel 4 gibt

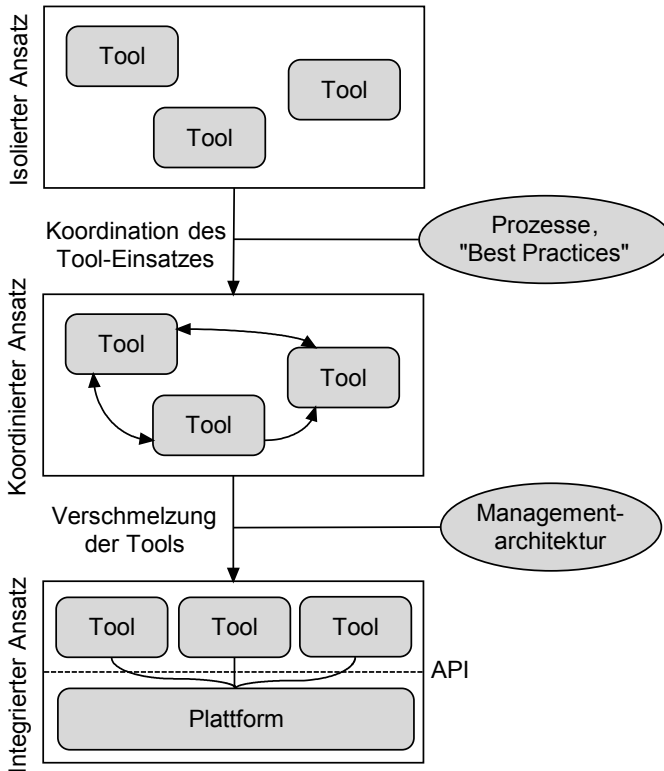


Abbildung 1.2: Einordnung bestehender Ansätze zur Realisierung von Managementlösungen in das Klassifizierungsschema nach Hegering et al. [1]

einen Überblick über die wichtigsten und am weitesten verbreiteten Ansätze aus dieser Kategorie.

Versucht man eine Einordnung der bestehenden Werkzeuge (*Tools*) und Rahmenwerke (*Frameworks*) in die Klassen von Ansätzen zur Realisierung von Managementlösungen gemäß Hegering et al. [1] vorzunehmen, so ergibt sich die in Abbildung 1.2 dargestellte Situation. Werkzeuge allein realisieren einen isolierten Ansatz. Das heißt, dass zu jedem konkreten Managementproblem ein eigenes, isoliertes Werkzeug geschaffen wird. Diese Werkzeuge arbeiten un-

abhängig voneinander – unter Umständen sogar auf Basis unterschiedlicher Datenbestände und ohne wechselseitige Schnittstellen.

Schwächen isolierter Ansätze: Hier sind die Probleme offensichtlich: Durch den Einsatz unterschiedlichster Werkzeuge für die verschiedenen Aufgabenbereiche des Service-Level-Managements – etwa ein Tool zum Dokumentenmanagement, ein zweites für die Generierung von Berichten (Reports) und ein drittes für die Verwaltung von Dienstinformationen – entsteht ein regelrechter „Zoo“ an heterogenen Tools. Das führt unvermeidbar zu Redundanzen zum Beispiel in der Datenhaltung und birgt zudem die Gefahr von Inkonsistenzen und Fehlern.

Entstehung koordinierter Ansätze: Rahmenwerke wie die *IT Infrastructure Library* (ITIL), die *Enhanced Telecom Operations Map* (eTOM) oder auch das IT-Governance-Framework *Control Objectives for Information and Related Technology* (COBIT) können als koordinierende Ansätze verstanden werden, wobei anzumerken ist, dass zumindest ITIL und COBIT ein generell wenig ausgeprägtes Tool-Verständnis mit sich bringen (vgl. Brenner [8]). Prinzipiell lässt sich aber sagen, dass Prozessrahmenwerke einen signifikanten Beitrag leisten, durch die Definition geeigneter Prozesse und ihrer Schnittstellen den koordinierten Einsatz von Ressourcen wie Technologien (also auch Werkzeugen) und Personen zu forcieren. Koordinierte Ansätze zeichnen sich dadurch aus, dass Werkzeuge zwar weiterhin in ihrer Funktionserbringung isoliert arbeiten, im Einsatz aber koordiniert werden und sich somit gegenseitig ergänzen – beispielsweise indem Resultate (*Outputs*) eines Tools als Eingaben (*Inputs*) für ein oder mehrere andere Tools dienen, wobei unter Umständen Adaptoren eingesetzt werden müssen, die in der Lage sind, unterschiedliche Datenformate ineinander zu überführen. Trotz der Möglichkeit eines einheitlichen Bediensystems sind die verfügbaren Informationen und Funktionen der Einzelanwendungen damit jedoch noch nicht vereinheitlicht.

Schwächen koordinierter Ansätze: Vor allem aber bieten koordinierte Ansätze keine belastbare Grundlage für die Erstellung neuer Werkzeuge, also für die Erweiterung des Managementbaukastens um neue Funktionen. In diesem Fall besteht die einzige Möglichkeit darin, sich an bestehenden Tools zu orientieren, um zu gewährleisten, dass das Kooperationsparadigma erhalten bzw. fortgeführt wird. Das wird umso schwieriger, je umfangreicher und heterogener das Gesamtsystem wird. Insbesondere basieren koordinierte Ansätze nicht – wie integrierte Managementlösungen –

auf einer gemeinsamen Plattform als Basiswerkzeug, auf der verschiedene Managementanwendungen durch die gebotene Programmierschnittstelle (*Application Programming Interface*, API) direkt aufsetzen können.

Zwischenfazit: Wünschenswert wäre im Bereich Service-Level-Management die Entwicklung in Richtung eines vollständig integrierten Managementansatzes. Nach Hegering et al. [1] ist ein integrierter Ansatz erst gegeben, wenn „*Informationen über wohldefinierte Schnittstellen zugänglich sind*“ und eine Managementplattform zur Verfügung steht, die ein „*offenes Trägersystem für Managementanwendungen*“ bereitstellt. Die Entwicklung einer vollständigen Managementarchitektur ist Voraussetzung dafür und daher Gegenstand der vorliegenden Arbeit. Im folgenden Abschnitt wird erläutert, welche Teilfragestellungen sich hieraus ergeben und wie sich aus diesen Teilfragestellungen die Vorgehensmethodik für diese Arbeit ableitet.

1.3 Haupt- und Teilfragestellungen dieser Arbeit

Die Hauptfragestellung, die dieser Arbeit zugrunde liegt, lautet also:

Wie müsste eine Managementarchitektur aussehen, auf deren Basis sich ein Managementsystem zum IT-gestützten, vollständig integrierten Service-Level-Management praktisch entwickeln lässt?

Die Entwicklung einer Architektur für ein integriertes Service-Level-Management als Voraussetzung für den Entwurf von Managementsystemen in heterogener Umgebung erfolgt idealerweise durch Definition der vier Teilmodelle, die sich in jeder vollständigen Managementarchitektur implizit oder explizit wiederfinden. Diese Teilmodelle sind das Informationsmodell, das Organisationsmodell, das Kommunikationsmodell und das Funktionsmodell, wie in Abbildung 1.3 überblicksartig dargestellt.

Während das Funktionsmodell die Managementfunktionen im Service-Level-Management beschreibt, die von den im Organisationsmodell identifizierten Rollen und Akteuren ausgeführt werden, beschreibt das Informationsmodell Methoden zur Modellierung der eigentlichen Managementobjekte (*Managed Objects*) – etwa Dienstvereinbarungen, Dienstkataloge, Messungen oder Berichte. Das Kommunikationsmodell stellt Konzepte zum Austausch von Managementinformation bereit und ist damit vor allem für den integrativen Aspekt maßgeblich. Aus Dienstmanagement-Sicht ergeben sich auf dieser Grundlage die folgenden zu untersuchenden Teilfragestellungen:

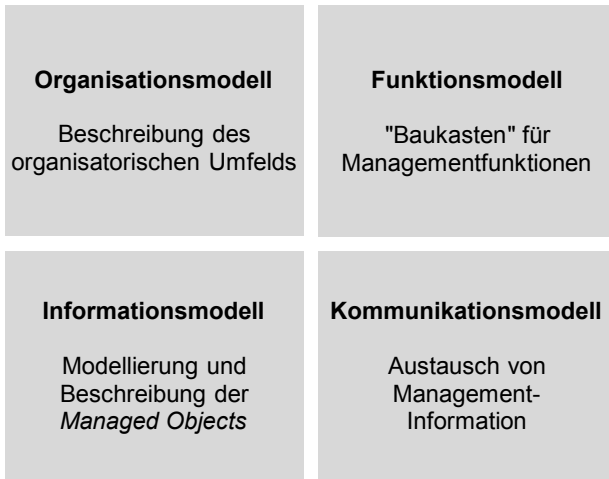


Abbildung 1.3: Die Teilmodelle einer Managementarchitektur

- Welche funktionalen Anforderungen können anhand realistischer Szenarien für die Disziplin Service-Level-Management identifiziert werden? Welche Anwendungsfälle (*Use Cases*) für ein SLM-unterstützendes Managementsystem ergeben sich aus diesen Anforderungen?
→ Szenariobasierte Anforderungsanalyse
- Welche Rollen und Akteure können im Service-Level-Management identifiziert werden, und wie hängen sie zusammen?
→ Entwurf eines Organisationsmodells
- Welche funktionalen Teilbereiche können im Service-Level-Management identifiziert werden, und wie ordnen sich die identifizierten Anwendungsfälle in diese Funktionsbereiche ein? Welche Kontroll- und Objektflüsse sind in den funktionalen Teilbereichen durch ein IT-gestütztes Managementsystem zu realisieren?
→ Entwurf eines Funktionsmodells
- Welche Informationen, Daten, Dokumente und Pläne werden im Service-Level-Management benötigt bzw. verwaltet? Wie sind diese Informationen zusammengesetzt? Welche Zusammenhänge bestehen zwischen den

einzelnen Informationsartefakten?

→ Entwurf eines Informationsmodells

- Welche Kommunikationsbeziehungen erfordern die Anwendungsfälle zwischen den Akteuren im Service-Level-Management? Wie kann die Kommunikation durch das Managementsystem effektiv unterstützt werden? Wie gliedert sich Service-Level-Management in ein (prozessorientiertes) Managementumfeld ein? Welche Zusammenhänge bestehen zu anderen Dienstmanagementdisziplinen und -prozessen?
→ Entwurf eines Kommunikationsmodells
- Wie kann eine Plattform für ein SLM-unterstützendes, IT-gestütztes und vollständig integriertes Managementsystem auf Basis der vier Architekturteilmodelle realisiert werden?
→ Bereitstellung eines Implementierungsleitfadens
- Wie lässt sich mit Hilfe des Anforderungskatalogs eine Bewertung (*Assessment*, *Benchmarking*) der so entwickelten Architektur bzw. beliebiger anderer Architekturen und Lösungsansätze für die Disziplin Service-Level-Management durchführen?
→ Entwurf eines Bewertungskatalogs und -verfahrens

Mit diesen Teilfragestellungen wurde bereits ein Teil der Vorgehensmethodik vorweg genommen, die der Arbeit zugrunde liegt. Der nächste Abschnitt geht hierauf noch etwas genauer ein. Weitere konkrete Einzelfragestellungen in Ergänzung zu den oben genannten ergeben sich im Verlauf der Arbeit und werden insbesondere zu Beginn des Entwurfs jedes Architekturteilmodells vorgestellt.

1.4 Vorgehen

Dieser Abschnitt beschreibt in erster Instanz die groben Schritte auf dem Weg zu einer Managementarchitektur für Service-Level-Management und schafft damit einen vorläufigen Rahmen für diese Arbeit. Der Abschnitt gliedert sich in vier Unterabschnitte: Zunächst wird gezeigt, wie der teilmodellbasierte Ansatz methodisch verfeinert wird (Abschnitt 1.4.1). Anschließend wird auf dieser Basis ein Vorgehensmodell gezeichnet, das die Struktur der Dissertation reflektiert (Abschnitt 1.4.2). Danach werden einige inhaltliche Schwerpunkte der Arbeit aufgezeigt (Abschnitt 1.4.3) und abschließend dargestellt, wie die verschiedenen Teilergebnisse dieser Arbeit zusammenhängen (Abschnitt 1.4.4).

1.4.1 Methodische Herangehensweise

Betrachtet man die Gruppen von Teilfragestellungen, die im vorherigen Abschnitt aufgeführt wurden, genauer, so erkennt man, dass einige Fragen im Kontext der Architekturteilmodelle darauf abzielen, ein besseres Verständnis über Aspekte und Zusammenhänge im Service-Level-Management aufzubauen; zum Beispiel: „*Welche Informationen, Daten, Dokumente und Pläne werden im Service-Level-Management benötigt bzw. verwaltet?*“. Andere hingegen adressieren konkrete Aspekte einer Automatisierung (IT-Support) im Service-Level-Management; beispielsweise: „*Welche Kontroll- und Objektflüsse sind in den funktionalen Teilbereichen durch ein IT-gestütztes Managementsystem zu realisieren?*“

Aus diesem Umstand ergibt sich eine der wichtigsten methodischen Weichenstellungen für diese Arbeit, nämlich die Realisierung zweier verschiedener Sichten auf jedes der vier Teilmodelle:

Prozess-Sicht: Die Modelle der Prozess-Sicht beschäftigen sich mit Aspekten der Realisierung eines SLM-unterstützenden Managementsystems *ohne eine spezifische Ausrichtung auf Automatisierungsaspekte*. Maßgebliche Teilaspekte sind in diesem Zusammenhang unter anderem Prozess- und Verfahrensabläufe sowie deren Steuerung, Personen und ihre Kommunikation untereinander, Dokumente, Vereinbarungen, Berichte und sonstige Informationsartefakte im SLM und ihre Zusammenhänge.

System-Sicht: Die Modelle der System-Sicht bauen auf den Modellen der Prozess-Sicht auf und erweitern bzw. ergänzen diese in einer Weise, dass die Modelle in der Gesamtheit einen Baukasten für ein *IT-gestütztes Managementsystem* bilden. Der Automatisierungsaspekt ist hier also vorrangig. Als wichtige Teilaspekte sind in diesem Kontext unter anderem Informationsobjektflüsse, Komponentenarchitekturen und konkrete, formale Datenmodelle zu nennen.

Dieses Vorgehen reflektiert die grundlegende methodische Herangehensweise, die in dieser Arbeit verfolgt wird. Eine detaillierte Auseinandersetzung mit methodischen Einzelaspekten und insbesondere mit dem Aufbau eines detaillierten Ordnungs- und Entwicklungsrahmens liefert Kapitel 2 (Methodik und Begriffsbildung).

1.4.2 Vorgehensmodell

Abbildung 1.4 illustriert das Vorgehensmodell, das den Rahmen dieser Arbeit bildet. Dabei stellt dieses Kapitel mit Motivation, Problembeschreibung, Identifikation von Teilfragestellungen und Vorgehensbeschreibung den ersten Schritt dar. Ihm folgt im zweiten Kapitel eine Beschreibung der *methodischen Grundlagen* sowie eine grundlegende, *strukturierte Begriffsbildung*. Kapitel 3 liefert eine *detaillierte Anforderungsanalyse*, deren Ziel es ist, Anforderungen an die zu entwickelnde Managementarchitektur zu identifizieren, einheitlich und aussagekräftig zu beschreiben und geeignet einzuordnen.

Im vierten Kapitel werden *themenverwandte Arbeiten* analysiert. Den Schwerpunkt bilden hier Prozessrahmenwerke und andere modellbasierte Ansätze im Bereich des Service-Level-Managements. Es wird jeweils untersucht, wie diese zur Begriffsbildung, Methodik oder zu den Lösungsansätzen dieser Arbeit beitragen. Andererseits wird dargestellt, wie sie sich inhaltlich untereinander und natürlich im Hinblick auf diese Dissertation abgrenzen.

Es folgen die beiden Kernkapitel der Arbeit: die Entwicklung der eigentlichen *Managementarchitektur und ihrer Teilmodelle* in Kapitel 5 und 6, sowie die Adressierung von *Implementierungsaspekten* in Kapitel 7 zur Transformation der plattformunabhängigen Architekturmodelle in plattformspezifische Modelle und Code. Ziel von Kapitel 5 und 6 ist vor allem eine konsistente, einheitliche und hinreichend formale Spezifikation der einzelnen Teilmodelle und ihrer Bestandteile. Kapitel 5 deckt dabei die Prozess-Sicht, Kapitel 6 die System-Sicht ab. Daneben werden Querbeziehungen zwischen den Teilmodellen hergestellt und ebenfalls spezifiziert. Ziel von Kapitel 7 ist es aufzuzeigen, wie diese Modelle in einem konkreten Plattformentwicklungsprojekt instrumentiert werden können.

Kapitel 8 enthält eine Zusammenfassung der erzielten Ergebnisse sowie einen Ausblick auf mögliche weiterführende Arbeiten.

1.4.3 Schwerpunkte

Eine Managementarchitektur kann nur dann maximal hilfreich bei der Realisierung eines IT-gestützten Managementsystems sein, wenn alle ihre Teilmodelle im Sinne der funktionalen Anforderungen vollständig beschrieben sind. Dennoch kann die Detailliertheit und Granularität der einzelnen Teilmodelle und ihrer Elemente variieren. Daher sollen an dieser Stelle bereits einige Schwerpunkte der vorliegenden Arbeit beschrieben werden. Hierbei handelt es sich um einen überblicksartigen Vorgriff auf die späteren Kernkapitel.

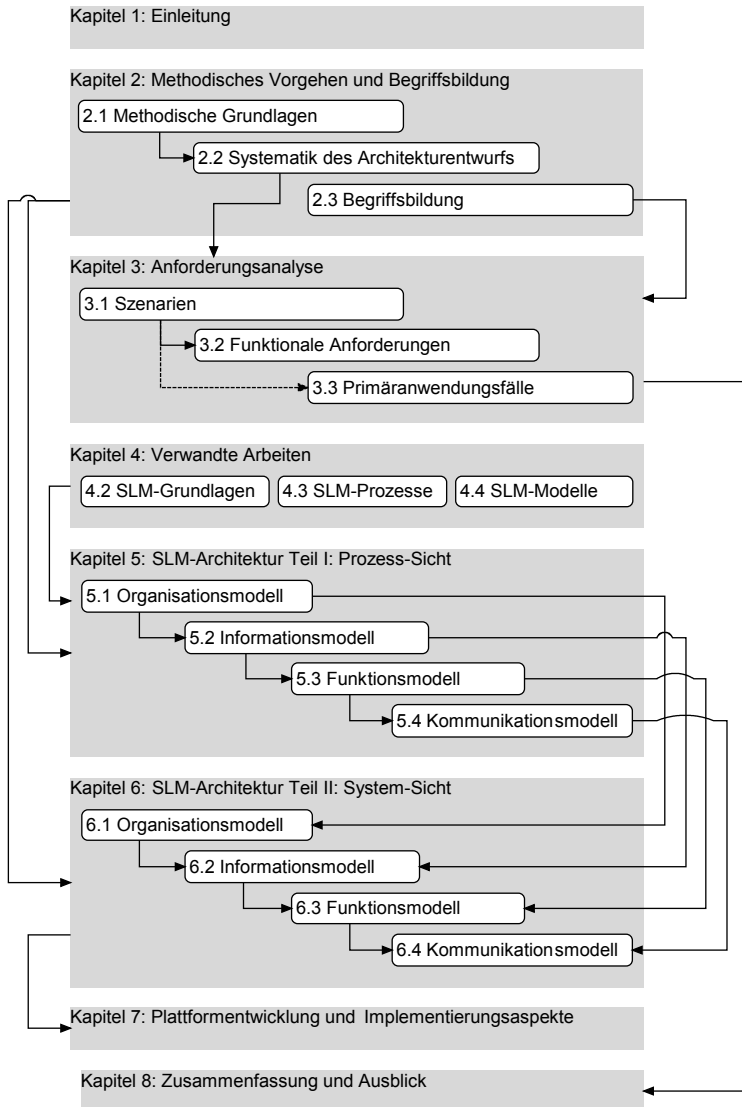


Abbildung 1.4: Vorgehensmodell

Schwerpunkte der Anforderungsanalyse Die Qualität einer Managementarchitektur steht und fällt mit der Plausibilität, Validität, Relevanz und Vollständigkeit der ihr zugrunde gelegten funktionalen Anforderungen. Aus diesem Grund wird der erste maßgebliche Schwerpunkt dieser Arbeit darauf gelegt, sicherzustellen, dass die Anforderungsanalyse diesen Forderungen gerecht wird. Dazu werden alle funktionalen Anforderungen an die Architektur aus realistischen IT-Provider-Szenarien abgeleitet. Die Auswahl der Szenarien wird durch eine morphologische Analyse unterstützt, die die unterschiedlichen Szenarien anhand festgelegter Merkmale voneinander abgrenzt. Jede identifizierte funktionale Anforderung wird im Kontext eines typischen Dienstlebenszyklus analysiert. Das hilft dabei, aus den funktionalen Anforderungen konkrete Anwendungsfälle für ein Managementsystem abzuleiten.

Schwerpunkte im Funktionsmodell Ein zentraler Aspekt im Funktionsmodell ist die Analyse der Kontroll- und Objektflüsse im Rahmen der aus der Anforderungsanalyse übergebenen Anwendungsfälle. Einen wesentlichen Teil des Funktionsmodells bilden daher Referenz-Workflows, die die teilweise komplexen Anwendungsfälle in feiner-granulare Aktivitäten herunterbrechen.

Schwerpunkte im Informationsmodell Im Informationsmodell liegt der Schwerpunkt auf der Modellierung der im Rahmen der Anforderungsanalyse identifizierten Informationsartefakte. Hier liefern die modellbasierten Ansätze aus dem Umfeld dieser Arbeit zahlreiche Ansatzpunkte und Modellvorschläge, die in dieser Arbeit nach entsprechender Anpassung und Erweiterung Verwendung finden. Zusätzlich sollen insbesondere die Zusammenhänge zwischen verschiedenen Informationsobjekten erfasst und dargestellt werden. Außerdem soll aufgezeigt werden, wie diese Artefakte auf einer gemeinsamen Datenbasis unter Vermeidung von Redundanzen erstellt, gepflegt und überwacht werden können. Es geht also darum, auf der Grundlage der gemeinsam zur Verfügung stehenden Managementinformation alle benötigten Artefakte mit Inhalt zu füllen und in ein effektives Service-Level-Management einzubetten.

Schwerpunkte im Kommunikationsmodell Im Rahmen des Kommunikationsmodells wird zwischen Intraprozess- und Interprozesskommunikation unterschieden. Im zweiten Fall nimmt der Zusammenhang zwischen der Disziplin des Service-Level-Managements und anderen Teilbereichen des Dienstmanagements eine zentrale Rolle ein. Die Relationen und Abhängigkeiten zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen

sind vielfältig. Ziel ist es, dass die entwickelte Architektur mit ihren plattformunabhängigen Modellen sowie eine darauf aufbauende Implementierung wohldefinierte Schnittstellen zu anderen Disziplinen realisiert, um wechselseitige Interaktion zu ermöglichen, ohne dabei konkrete Annahmen über die in diesen Bereichen jeweils konkret eingesetzten Lösungen und Werkzeuge zu machen.

Diese beiden Ziele *Integration von SLM mit anderen Disziplinen* und *keine Annahmen über die Ausgestaltung der jeweiligen Managementsysteme* sind natürlich konfliktär und in dieser Ausprägung unvereinbar. Daher wird im Rahmen des Kommunikationsmodells ein Ansatz verfolgt, der an dieser Stelle einen Kompromiss realisiert: Statt ein einziges, vollständig integriertes Dienstmanagementsystem zu fordern, das alle denkbaren Teildisziplinen – und somit auch das Service-Level-Management als eine von ihnen – umschließt, wird von mehreren verschiedenen und prinzipiell eigenständigen Managementsystemen ausgegangen, die jeweils für sich genommen einen integrierten Ansatz verwirklichen. Zwischen diesen Managementsystemen wird jedoch (nur) ein kooperativer Ansatz verfolgt, der sich von einem integrierten dadurch unterscheidet, dass nicht eine gemeinsame Managementplattform mit zugehöriger Informationsbasis existieren muss. Das hat den Hauptnachteil, dass Informationsredundanzen und Inkonsistenzen nicht völlig ausgeschlossen werden können, bringt dafür aber im Gegenzug erhöhte Flexibilität und mehr Freiheitsgrade, da kein bis auf die Ebene der Datenmodellierung vereinheitlichtes Informationsmodell benötigt wird.

Zusammengefasst: Der Schwerpunkt des Interprozess-Kommunikationsmodells liegt in der Verfolgung eines dualen Managementsystem-Ansatzes mit vollständig integrierten Managementsystemen auf Ebene der Teildisziplinen und Kooperation ohne gemeinsame Plattform zwischen diesen Systemen.

Schwerpunkte im Implementierungsleitfaden Im Rahmen der Implementierungsaspekte steht die Realisierung einer Managementplattform im Vordergrund, die das maßgebliche technische Fundament für ein integriertes, IT-gestütztes Managementsystem darstellt. Komponenten einer solchen Plattform sind das Kernsystem, Basisanwendungen, Managementapplikationen und eine grafische Benutzeroberfläche für einen interaktiven Zugang zu den Plattformfunktionalitäten. Wichtigstes Ziel des Implementierungsleitfadens ist es darzustellen, wie diese Komponenten auf Basis der Architekturteilmodelle realisiert werden können.

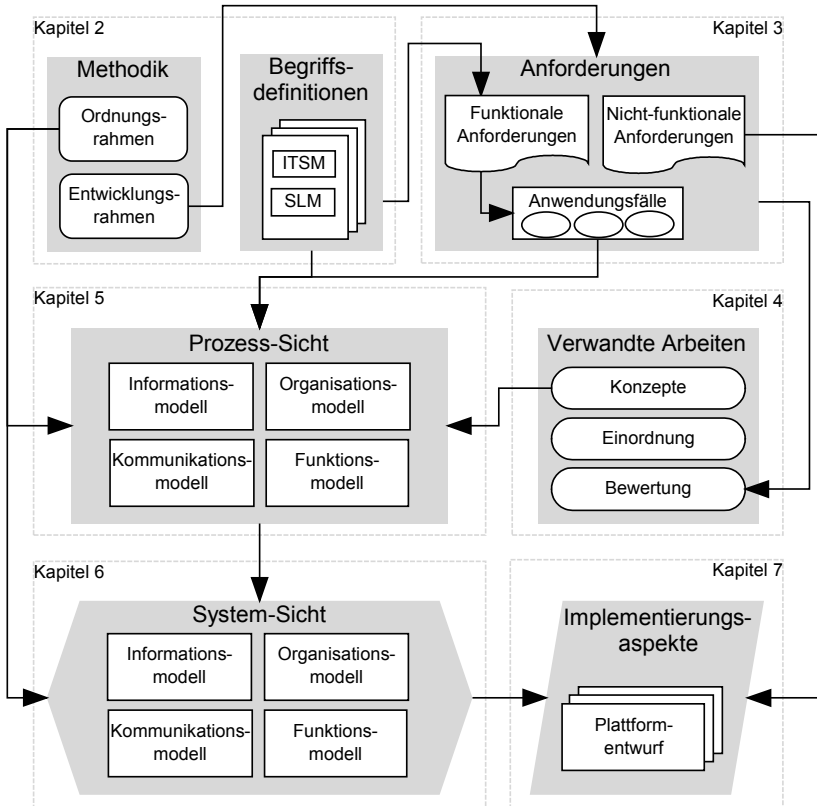


Abbildung 1.5: Zusammenhänge der Teilergebnisse

1.4.4 Zusammenhänge der Teilergebnisse

Jeder einzelne Schritt des Vorgehensmodells liefert Teilergebnisse, die wiederum Eingaben für spätere Schritte sein können. Während im vorigen Abschnitt der Ablauf der einzelnen Schritte des Vorgehens dargestellt wurde, zeigt Abbildung 1.5, wie die erzielten Teilergebnisse untereinander zusammenhängen.

1.5 Einordnung und Abgrenzung

In einem kurzen Überblick soll nun noch dargestellt werden, wie sich die vorliegende Arbeit im Kontext verwandter wissenschaftlicher Arbeiten und akademischer wie industrieller Forschungsprojekte sowohl innerhalb als auch außerhalb des *Munich Network Management*-Teams¹ positioniert. Dabei werden Schnittmengen und Unterschiede aufgezeigt. Ziel ist es, die thematische Ausrichtung dieser Arbeit – unabhängig von der in Kapitel 4 durchgeführten Analyse verwandter Arbeiten – klar von den Zielsetzungen anderer Autoren und Projekte abzugrenzen.

Nachdem vorangehend dargestellt wurde, welche Fragestellungen durch die vorliegende Arbeit adressiert werden sollen, werden in diesem Abschnitt auch ihre inhaltlichen Begrenzungen und Schnittstellen zu anderen Arbeiten explizit herausgearbeitet.

1.5.1 Arbeiten aus dem MNM-Team

Die Dissertationen und Habilitationsschriften der nachfolgenden Autoren beschäftigen sich mit verschiedensten Fragestellungen aus dem Bereich des Dienstmanagements. Alle Arbeiten verbindet eine gemeinsame Forschungsausrichtung: das Management komplexer IT-Infrastrukturen und der durch sie erbrachten Dienste zu erleichtern.

Michael Brenner diskutiert in seiner Dissertation [9] allgemein das Thema der Werkzeugunterstützung (*Tool Support*) für ITIL²-konforme Prozesse. Im Rahmen einer hierfür entwickelten Taxonomie wird Service-Level-Management als gering strukturierter Prozess beschrieben, der von wechselseitigem Informationsaustausch zwischen verschiedenen Parteien geprägt ist. Er fällt damit in die Kategorie *Cooperation Support*, die in der Arbeit von Michael Brenner nicht weiter behandelt wird.

Die größte Gemeinsamkeit der Arbeit von Brenner und dieser Dissertation besteht in der Zielsetzung, Dienstmanagement durch den gezielten Einsatz IT-gestützter Werkzeuge in seiner Wirksamkeit und seinem Wirkungsgrad zu verbessern. Ähnlichkeiten bestehen auch in der methodischen Herangehensweise: Auch Brenner verfolgt insgesamt einen modellgetriebenen Ansatz. Als unterschiedlich hingegen erweist sich jeweils

¹<http://www.mnm-team.org>

²*IT Infrastructure Library*

die konkrete Vorgehensmethodik. Und während sich diese Arbeit spezifisch mit Service-Level-Management beschäftigt, adressiert Brenner in seiner Arbeit eher allgemeine Aspekte der Werkzeugunterstützung für Dienstmanagementprozesse, wobei schwerpunktmäßig einige der Prozesse aus dem ITIL-Band *Service Support* [10] genauer untersucht werden. Die Einordnung der Ergebnisse von Brenner in den Kontext der vorliegenden Dissertation wird in Kapitel 4 ausführlich diskutiert.

Martin Sailer beschäftigt sich in seiner Dissertation [11] mit Möglichkeiten der Dienstinformationsmodellierung. Den Mittelpunkt bildet dabei das Konzept der Service-MIB (*Management Information Base*). Die hierfür entwickelten Konzepte wie zum Beispiel die *Service Monitoring Architecture* (SMONA) oder die *Service Information Specification Language* (SISL) schließen ein Stück weit die Lücke zwischen dem klassischen Netz- und Systemmanagement und dem Dienstmanagement, die gerade im Service-Level-Management eine signifikante Herausforderung darstellt.

Das Problem einer Abbildung zwischen Managementinformationen und -aktionen des Netz- und Systemmanagements und solchen des Dienstmanagements wurde bis heute nicht in Gänze gelöst, auch wenn die Arbeit von Sailer maßgebliche Ansätze und Prototypen insbesondere für den Bereich der Überwachung (*Monitoring*) liefert. Die Weiterentwicklung dieser Ansätze ist *nicht* Teil der vorliegenden Arbeit. Profitieren kann sie allerdings insbesondere von Sailer's Informationsanalyse dienstbezogener Managementinformation.

David Schmitz präsentiert in seiner Dissertation ein Rahmenwerk zur Unterstützung der Dienst- und Geschäftsauswirkungsanalyse und der Auswahl geeigneter Wiederherstellungsoptionen nach dem Auftreten von Infrastruktur-Fehlern [12]. Workflows zur Auswirkungsbestimmung und Wiederherstellungsplanung sowie eine unterstützende Komponentenarchitektur bilden den Kern seiner Arbeit, die in die Dienstmanagementdisziplin des Fehlermanagements einzuordnen ist.

Aus der Perspektive einer SLM-Architektur ist weniger die konkrete Ausgestaltung dieser Workflows und Komponenten von Interesse als vielmehr die Realisierung einer möglichen Schnittstelle zwischen einem SLM-unterstützenden Managementsystem und einem auf den Ergebnissen von Schmitz basierendem Support-System für Fehlermanagement. Diesem Aspekt wird in der vorliegenden Arbeit im Rahmen des Kommunikationsmodells der Architektur eine große Bedeutung zuteil, wie bereits im Abschnitt 1.4.3 beschrieben wurde.

Markus Garschhammer führt in seiner Dissertation [13] ein Verfahren zur formalen Spezifikation von Dienstgütemerkmalen ein. Hierzu wird die formale Sprache QoSSL eingeführt sowie ein zugehöriger Übersetzer, der aus formalen Dienstgütespezifikationen automatisch Messsysteme erzeugen kann. Dadurch wird erreicht, dass Dienstgütemerkmale nicht nur während der Verhandlungsphase eines Dienstes rechnergestützt festgelegt, sondern auch während der Betriebsphase entsprechend gemessen werden können.

Die Arbeit von Garschhammer ist somit in das qualitätsorientierte Dienstmanagement einzuordnen. Die Spezifikation und Messung von Dienstgüteparametern ist ein Teilaspekt des Service-Level-Managements, und insofern liefert Garschhammer mit seinen Ergebnissen wertvollen Input für eine SLM-Architektur.

Holger Schmidt beschäftigt sich in seiner Dissertation [14] speziell mit dem Entwurf und der konkreten Ausgestaltung von Dienstvereinbarungen. Die Idee, die seiner Arbeit zugrunde liegt, besteht darin, SLAs auf der Basis von Dienst(management)prozessen zu erstellen, wobei Schmidt den Prozessbegriff synonym zum Workflow-Begriff verwendet. Zehn verschiedene Prozesse wurden in dieser Arbeit für den SLA-Entwurf berücksichtigt: Bereitstellung, Änderungsmanagement, Wartung, Problemmanagement, Sicherheitsmanagement, Abrechnung, Kundenbetreuung, Betrieb, Vereinbarungsmanagement und Beendigung. Jeder dieser Prozesse wurde detailliert mit dem Ziel modelliert, während der Modellierung alle notwendigen Informationen für die Erstellung von Dienstvereinbarungen zu identifizieren.

Die Ergebnisse der Arbeit von Schmidt liefern wertvollen und maßgeblichen Input für die Erstellung eines SLA-Datenmodells, das in der vorliegenden Arbeit im Rahmen von Kapitel 6 adressiert wird.

Gabi Dreo Rodosek präsentiert in ihrer Habilitationsschrift [4] ein äußerst umfassendes und breit angelegtes Rahmenwerk (*Framework*) für das Management von IT-Diensten, dessen Kern aus einem generischen Dienstmodell besteht. Ausgehend von diesem Dienstmodell werden schwerpunktmäßig zwei Aspekte in Dreos Arbeit adressiert: die dynamische Dienstbereitstellung (*dynamic service provisioning*) und das Management der Dienstqualität (*service quality management*). Die Arbeit bedient sich konzeptuell an etablierten Lösungen aus dem klassischen Netz- und Systemmanagement und überträgt diese Konzepte in eine dienstorientierte Sicht.

Insgesamt stellt Dreo mit ihren Ergebnissen den Nährboden für zahlreiche Arbeiten innerhalb des MNM-Teams zur Thematik Dienstmanagement bereit. Insbesondere Sailer und Garschhammer bauen darauf auf, indem sie ausgewählte Aspekte des Rahmenwerks in der Tiefe behandeln. Dreos Rahmenwerk und besonders ihr Dienstmodell sowie der von ihr verwendete Dienstlebenszyklus finden auch in der vorliegenden Dissertation Verwendung (vgl. zum Beispiel Kapitel 2.3 und 3.2.2). Eine detailliertere Auseinandersetzung mit den Ergebnissen von Dreo wird in Kapitel 4 vorgenommen.

1.5.2 Forschungsprojekte

Auch außerhalb des MNM-Teams werden Forschungsfragestellungen aus dem Bereich des dienstorientierten IT-Managements im Allgemeinen und des Service-Level-Managements im Speziellen zunehmend aufgegriffen. Die wichtigsten zum Zeitpunkt der Veröffentlichung dieser Arbeit relevanten und anerkannten Projekte mit Forschungsausrichtung seien hier kurz genannt und ebenfalls im Kontext der Fragestellungen dieser Arbeit positioniert.

EMANICS (*European Network of Excellence for the Management of the Internet and Complex Services*)³ versteht sich als ein EU-weites Netzwerk wissenschaftlicher Einrichtungen zur Untersuchung verschiedenster Fragestellungen aus dem Bereich IT-Service-Management und zum Austausch wissenschaftlicher Forschungsergebnisse. Service-Level-Management spielt insbesondere im Forschungsteilprojekt (*Work Package*) mit dem Titel *Economic Management* eine wichtige Rolle, wobei hier hauptsächlich bestehende Ansätze und Tools analysiert werden.

SLA@SOI (*Empowering the Service Economy with SLA-aware Infrastructures*)⁴ ist ein EU-finanziertes Forschungsprojekt mit der übergeordneten Vision, mit Hilfe von SLAs Umfelder zu schaffen, in denen IT-Dienste wie ökonomische Güter gehandelt werden können. Drei ambitionierte Ziele werden konkret verfolgt:

1. Vorhersehbarkeit und Verlässlichkeit (*Predictability and Dependability*): Die Qualitätseigenschaften von IT-Diensten sollen bereits während der Verhandlungsphase genau vorhergesagt und zur Laufzeit erzwungen werden können.

³<http://www.emanics.org>

⁴<http://www.sla-at-soi.org/>

2. Transparentes SLA-Management: Es soll sichergestellt sein, dass SLAs über die gesamte Geschäfts- und IT-Hierarchie transparent verwaltet werden können.
3. Automatisierung: Die Verhandlung von SLAs sowie die Bereitstellung, der Betrieb und die Überwachung von IT-Diensten soll vollständig automatisiert werden.

Ob diese Ziele durch SLA@SOI erreicht werden, kann zum Zeitpunkt der Veröffentlichung dieser Dissertation noch nicht beurteilt werden, da sich das Projekt noch im ersten seiner für drei Jahre angesetzten Laufzeit befindet. Speziell für das dritte Projektziel könnte diese Arbeit aber einen wertvollen Beitrag leisten.

1.6 Angestrebte Ergebnisse dieser Arbeit

Vor dem Hintergrund der in diesem Kapitel aufgeführten Fragestellungen können die wichtigsten beabsichtigten Ergebnisse der Arbeit wie folgt zusammengefasst werden:

1. Die Arbeit liefert eine umfassende Nomenklatur des begrifflichen Kontextes im Bereich Service-Level-Management.
2. Ausgehend von verschiedenen Szenarien liefert die Arbeit einen fundierten Anforderungskatalog für eine Managementarchitektur im Bereich Service-Level-Management.
3. Es wird ein umfassender Überblick über aktuell verfügbare Arbeiten im Bereich Service-Level-Management gegeben, mit besonderem Fokus auf prozessorientierte Rahmenwerke und modellbasierte Lösungsansätze.
4. Die Arbeit liefert eine vollständige, plattformunabhängige und somit wiederverwendbare Managementarchitektur unter Spezifikation aller relevanten Teilmodelle, die als „Baukasten“ und Leitfaden für den Entwurf eines SLM-unterstützenden Managementsystems im Sinne des Anforderungskatalogs herangezogen werden kann.
5. Es wird gezeigt, wie die Architektur zur Realisierung einer konkreten Managementplattform instrumentiert werden kann und wie auf diese Weise ein IT-gestütztes Managementsystem für Service-Level-Management realisiert werden kann.

Mögliche Adressaten der Ergebnisse dieser Arbeit sind

- Softwareentwickler, die Werkzeuge oder eine Managementplattform für IT-gestütztes Service-Level-Management implementieren möchten,
- IT-Service-Manager oder *Chief Information Officers* (CIOs), die ein besseres Verständnis für Anforderungen und Zusammenhänge im Service-Level-Management aufbauen möchten,
- Auditoren, die die Konformität eines Managementsystems für Service-Level-Management gegen einen generischen Anforderungskatalog überprüfen möchten,
- Autoren von Standards, *Best Practice*-Titeln und *Self Assessment*-Handbüchern im Bereich Service-Level-Management sowie
- Wissenschaftler, die sich mit verwandten Forschungsfragestellungen aus dem Bereich des Dienstmanagements beschäftigen.

Kapitel 2

Methodisches Vorgehen und Begriffsbildung

Inhalt dieses Kapitels

2.1	Methodische Grundlagen	24
2.1.1	Modellgetriebene Architektur	25
2.1.2	Prozessorientierung	25
2.1.3	Systembasierter Ansatz	27
2.1.4	Zusammenfassung: Methodische Grundlagen	31
2.2	Systematik des Architekturentwurfs	32
2.2.1	Ordnungsrahmen	32
2.2.2	Entwicklungsrahmen	35
2.3	Begriffsbildung	38
2.3.1	Der Dienstbegriff	38
2.3.2	Begriffe der Providerdomäne	45
2.3.3	Begriffe der Kundendomäne	46
2.3.4	Begriffe auf der Schnittstelle zwischen Provider und Kunde	47
2.4	Zusammenfassung	48

Nachdem im vorigen Kapitel einleitend dargestellt wurde, welche Herausforderungen die Dienstorientierung im IT-Management mit sich bringen, und motiviert wurde, wie in der Disziplin des Service-Level-Managements durch eine Managementarchitektur der Weg hin zu einem integrierten, werkzeugorientierten Ansatz geebnet werden kann, befasst sich dieses Kapitel schwerpunktmäßig mit dem methodisch-systematischen Ansatz, der die Grundlage für diese Arbeit bildet.

In diesem Rahmen wird unter anderem ausführlich begründet, warum der in der Einleitung bereits kurz eingeführte teilmodellbasierte Ansatz zielführend

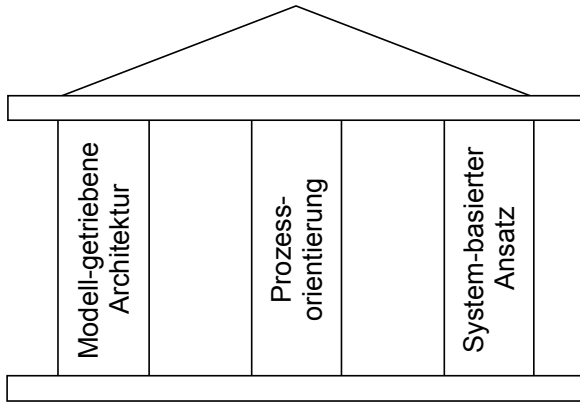


Abbildung 2.1: Drei Säulen des methodischen Vorgehens

für die Entwicklung der Architektur ist und einen sinnvollen Ordnungsrahmen für die Bestandteile der Managementarchitektur darstellt.

2.1 Methodische Grundlagen

Drei wichtige Konzepte bilden den methodischen Ausgangspunkt für diese Arbeit und sind somit die tragenden Säulen der zu entwickelnden Architektur für Service-Level-Management (vgl. Abbildung 2.1). Diese Kern-Konzepte sind

1. das Paradigma einer *modellgetriebenen Architektur*, welches ein striktes Top-Down-Vorgehen impliziert und den Grundprinzipien der *Model Driven Architecture (MDA)* [15, 16] folgt,
2. *Prozessorientierung* als maßgeblicher „Treiber“ eines effektiven Dienstmanagements, sowie
3. die Verfolgung eines *systembasierten Ansatzes* im Dienstmanagement.

Alle drei Säulen werden nun kurz vorgestellt, wobei für jedes Konzept schwerpunktmäßig dargestellt wird, *warum* es einen Teil des Fundaments dieser Arbeit bildet und *wie* sich dies jeweils auf das weitere Vorgehen auswirkt.

2.1.1 Modellgetriebene Architektur

Insgesamt verfolgt die Arbeit einen strikten Top-Down-Ansatz, wie auf der linken Seite der Abbildung 2.2 skizziert. Ziel ist es, auf der Basis realer Szenarien (funktionale) Anforderungen abzuleiten, die – in Anwendungsfälle überführt – den Ausgangspunkt für die Entwicklung plattformunabhängiger Modelle bilden und somit als Zielvorgaben für alle weiteren Arbeitsschritte fungieren. Die plattformunabhängigen Modelle bilden den eigentlichen Kern der Managementarchitektur. Auf ihrer Basis werden schließlich Aspekte der Implementierung einer geeigneten Managementplattform betrachtet.

Dieses Vorgehen orientiert sich an dem Ansatz der *Model Driven Architecture (MDA)*, der in Kapitel 4 noch detaillierter vorgestellt wird. Abbildung 2.2 zeigt, wie die Schritte des Top-Down-Ansatzes dieser Arbeit auf die grundlegenden MDA-Konzepte abgebildet werden können.

Ein solcher Top-Down-Ansatz diktiert ein logisches und streng lineares Vorgehen. Offen bleiben jedoch die Fragen, wie die einzelnen Schritte konkret aufeinander aufbauen und ineinander überführt werden können und wie die Ergebnisse der einzelnen Schritte sinnvoll und strukturiert dokumentiert werden sollten. Die erste Frage (Wie gelangt man von Schritt n zu Schritt $n + 1$?) beantwortet der Entwicklungsrahmen der Architektur (Abschnitt 2.2.2), die zweite Frage (Wie werden die Ergebnisse von Schritt n strukturiert?) wird durch den Ordnungsrahmen adressiert (Abschnitt 2.2.1). Beide zusammen bilden die Systematik des Architekturentwurfs (siehe auch Brenner et al. [17]).

2.1.2 Prozessorientierung

Das Prinzip der Prozessorientierung steht im Mittelpunkt nahezu aller modernen Ansätze für geschäftsorientiertes IT-Dienstmanagement. Die prominentesten Beispiele sind der internationale Standard ISO/IEC 20000 [18], die *IT Infrastructure Library (ITIL)* [19] und alle auf ihr aufbauenden Rahmenwerke sowie die *enhanced Telecom Operations Map (eTOM)* [20]. Weil das Ziel dieser Arbeit in der Entwicklung einer Architektur für Service-Level-Management liegt, die sich für die Anwendung in prozessorientierten Managementumfeldern eignet, müssen die zentralen Ideen der Prozessorientierung in den Architekturentwurf einfließen. Daher wird Prozessorientierung als zweites methodisches Kernkonzept dieser Arbeit zugrunde gelegt.

Prozessorientierung Bei dem Prinzip der Prozessorientierung handelt es sich um eine Grundhaltung in einer (IT-)Organisation, die jede (wiederkeh-

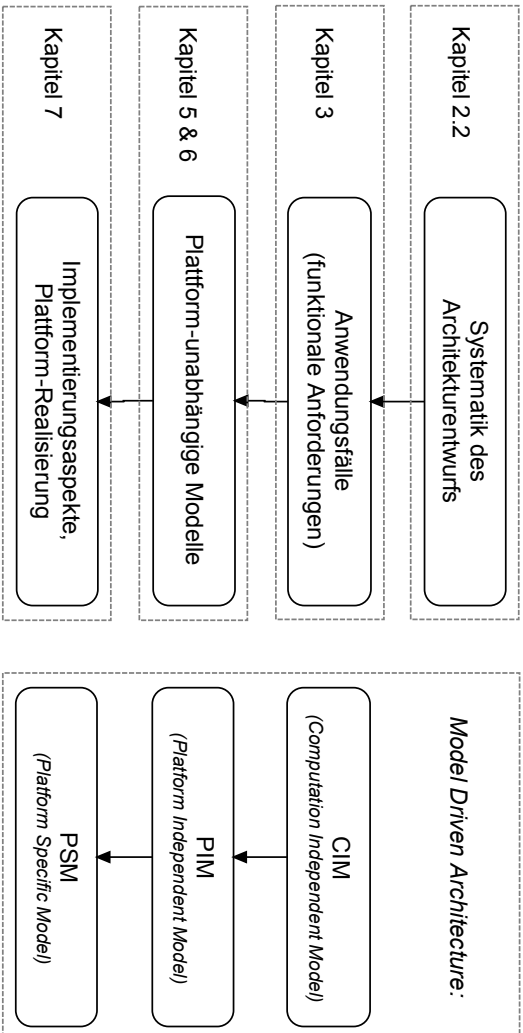


Abbildung 2.2: Top-Down-Vorgehen dieser Arbeit im Kontext des MDA-Ansatzes

rende) Aktivität als Teil eines festgelegten Prozesses betrachtet. Ziel ist die Steigerung von Qualität und Produktivität.

Vielfach wird der Begriff des Prozesses gleichgesetzt mit „Ablauf“ oder „Folge von Aktivitäten“. Diese Sichtweise ist allerdings etwas zu restriktiv und beschreibt nur einen Teilaspekt des Prozesskonzepts. Insbesondere bei der Betrachtung von Managementprozessen müssen zwangsläufig eine Reihe weiterer Punkte betrachtet werden, wie die folgende an DIN/ISO 9000 [21] angelehnte Definition des Prozessbegriffs deutlich macht:

Prozess Ein Prozess beschreibt einen Satz von in Wechselbeziehungen stehenden Mitteln und Tätigkeiten, die Eingaben in Ergebnisse umwandeln. Zu den Mitteln können Personal, Einrichtungen und Anlagen, Technologie und Methodologie gehören.

Wichtige Aspekte allgemeiner Prozesse sind neben Eingaben (Inputs), Verarbeitungsschritten und Ergebnissen (Outputs) auch die Prozessziele, Steuergrößen (Synonym: Parameter), Regeln und Verantwortlichkeiten sowie wechselseitige Schnittstellen zwischen verschiedenen Prozessen. Ein gutes Prozessmodell sollte all das adressieren, und je umfassender dies geschieht, umso effektiver kann die Instrumentierung der Prozesse zu den vorgesehenen Managementzwecken gelingen.

Zur Vollständigkeit seien an dieser Stelle noch zwei weitere Begrifflichkeiten genannt, die im Zusammenhang mit Prozessen und Prozessorientierung oft Erwähnung finden. Es handelt sich um die Begriffe *Richtlinie* und *Verfahren*.

Richtlinie Eine Richtlinie (*Policy*) dokumentiert die übergeordnete Ausrichtung sowie allgemeine Absichten und Vorschriften für einen klar abgegrenzten Anwendungsbereich (*Scope*), wie zum Beispiel einen Prozess als Teil eines Managementsystems (siehe nächster Abschnitt).

Verfahren Ein Verfahren (*Procedure*) definiert eine festgelegte Art und Weise, eine Tätigkeit oder einen Prozess auszuführen.

2.1.3 Systembasierter Ansatz

Der Systembegriff ist gerade im IT-Umfeld stark überladen. Dieser Abschnitt soll verdeutlichen, was mit einem *system approach for management* (vgl. DIN/ISO 9001 [22]) gemeint ist und welche Rolle dieser Ansatz im Rahmen dieser Arbeit spielt.

In der Systemtheorie existieren zahlreiche Definitionen für den Begriff des *Systems*. Eine universelle Begriffsdefinition lieferten Hall und Fagen bereits in den 50er Jahren [23]:

System Ein System ist eine Ansammlung von Elementen und deren Eigenschaften, die durch Wechselbeziehungen miteinander verbunden sind.

Nach Bossel [24] ist ein System gekennzeichnet durch „*seinen Systemzweck (Funktion), seine Systemelemente und Wirkungsverknüpfungen (Wirkungsstruktur) und seine Systemintegrität*“. Klassische Beispiele für Systeme sind Wirtschaftssysteme, Verkehrssysteme, Ökosysteme oder aber auch Betriebssysteme (vgl. Skriptum Einführung in die Systemwissenschaft [23]).

Dieser kurze systemtheoretische Abriss zeigt, dass der Systembegriff beliebig generisch ist und dass Systeme in allen Bereichen des Lebens eine bedeutende Rolle spielen. Die Systeme, die im Rahmen dieser Arbeit von Interesse sind, sind die sogenannten *Managementsysteme*. So besteht die Haupt-Motivation für diese Arbeit in der Vision von einem vollständig integrierten, IT-gestützten Managementsystem für Service-Level-Management. Direkter Gegenstand der Arbeit ist die Entwicklung einer *Managementarchitektur*, die einen pragmatischen Weg in Richtung dieses Fernziels aufzeigen soll. Somit sind das Managementsystem und die Managementarchitektur zwei zentrale Konzepte im Zusammenhang mit dieser Arbeit.

Die folgende allgemeine Definition für den Begriff des Managementsystems orientiert sich am Verständnis, das unter anderem den prozess- und systemorientierten Normen DIN/ISO 9000 und ISO/IEC 20000 zugrunde liegt:

Managementsystem Als Managementsystem wird die Gesamtheit aller Richtlinien, Prozesse, Verfahren, Werkzeuge (Tools) und Ressourcen (Menschen, Anlagen) bezeichnet, die in koordinierter Weise eingesetzt werden, um die anfallenden Managementaufgaben ziel-, kunden- und qualitätsorientiert zu planen, auszuführen, zu dokumentieren und ständig zu verbessern.

Bemerkung: Die Notwendigkeit eines Managementsystems wächst ganz allgemein mit der Komplexität der Managementaufgabe. Einfache Managementaufgaben mögen ohne dedizierte Managementsysteme auskommen, sofern Effektivität und Effizienz trotzdem in ausreichender Ausprägung sichergestellt sind. In solchen Fällen könnte ein Managementsystem mit seinen Richtlinien, Verfahren, Prozessen

und Dokumentationsanforderungen sogar eher schaden als nützen, da es einen zu großen *Overhead* erzeugt.

Entscheidend für das Verständnis der Idee eines Managementsystems ist die Tatsache, dass das Vorhandensein eines solchen grundsätzlich noch keinerlei Automatisierung impliziert. Prinzipiell können Managementsysteme völlig ohne informationstechnische Komponenten und somit ohne IT-Unterstützung auskommen. Fehlende IT-Unterstützung für ein Managementsystem resultiert aber in zwei Hauptproblemen:

1. Die ständige und systemkonforme Umsetzung der Richtlinien, Prozesse und Verfahren verlangt größte Disziplin der am System beteiligten Menschen. Je größer ihre Anzahl und je komplexer das System ist, umso unwahrscheinlicher wird es, dass diese Disziplin verlangt und dauerhaft gewährleistet werden kann. Der „Faktor Mensch“ stellt also in allen Managementsystemen – und insbesondere dort, wo deterministisches Verhalten ein kritischer Erfolgsfaktor ist – ein potenzielles Risiko dar.
2. Selbst wenn das Managementsystem ohne IT-Unterstützung effektiv funktionieren würde, so stellt sich die Frage nach der Optimalität des Ressourceneinsatzes und damit nach der Effizienz des Systems. Manuelle Dokumentation ohne Dokumentenmanagement, Kommunikation ohne technische Unterstützung, Ablaufkoordination ohne Workflow-Management-Komponenten und vieles andere mehr machen deutlich, welche Defizite in so einem Fall offensichtlich werden würden.

Es liegt auf der Hand, dass heute nahezu jedes Managementsystem in irgendeiner Weise IT-gestützt betrieben wird. Wünschenswert wäre natürlich eine optimale und umfassende IT-Unterstützung für ein Managementsystem, die die Idee eines vollständig integrierten Managements möglich macht (vgl. Kapitel 1). Das wiederum führt zum Konzept der *Managementplattform*, das in Anlehnung an Hegering et al. [1] wie folgt beschrieben werden kann:

Managementplattform Eine Managementplattform ist eine Laufzeit- und Entwicklungsumgebung für verteilte Managementanwendungen, die gemäß den Konzepten von Managementarchitekturen arbeiten und somit einen integrierten Ansatz ermöglichen.

Weiter sind Managementplattformen durch folgende Eigenschaften gekennzeichnet (Auszug):

- Sie bieten offen gelegte Programmierschnittstellen für Anwendungen und sind dadurch erweiterbar.
- Alle Anwendungen, die auf der Plattform aufsetzen, basieren auf einem gemeinsamen Informationsmodell für Managementobjekte.
- Durch gemeinsame Basisanwendungen wird eine funktionelle Integration des Managementsystems unterstützt.

Für den Moment ist dieser Überblick über Managementplattformen ausreichend. Kapitel 7 setzt an dieser Stelle wieder an, um auf Basis der bis dahin entwickelten Architektur Implementierungsaspekte einer Plattform für Service-Level-Management zu adressieren. Entscheidend ist, dass nur eine geeignete Managementplattform die (Teil-)Automatisierung eines Managementsystems ermöglicht. Daraus ergibt sich die folgende Definition eines *vollständig IT-gestützten Managementsystems*:

Vollständig IT-gestütztes Managementsystem Ein (vollständig) IT-gestütztes Managementsystem ist ein Managementsystem, das auf einer geeigneten Managementplattform aufsetzt, die die Planung, Ausführung und Dokumentation anfallender Managementaufgaben effektiv und effizient durch ihre Laufzeitumgebung unterstützt.

Ziel dieser Arbeit ist es nicht, ein solches integriertes, IT-gestütztes Managementsystem für Service-Level Management für ein gegebenes Szenario zu programmieren, sondern die Entwicklung solcher konkreter Systeme durch Bereitstellung einer geeigneten und generischen *Managementarchitektur* zu erleichtern. Entsprechend wird der Begriff der Managementarchitektur gemäß der folgenden Definition verwendet:

Managementarchitektur Eine Managementarchitektur ist die Gesamtheit aller Modelle und Entwurfsrichtlinien zur Entwicklung eines konkreten (IT-gestützten) Managementsystems.

Diese einfache und prägnante Definition des Begriffs der Managementarchitektur umfasst alle drei Kernkonzepte dieser Arbeit – von der modellgetriebenen Architektur über die Prozessorientierung bis hin zum systembasierten Ansatz.

<i>Methodisches Konzept</i>	<i>Ziele</i>	<i>Zusammenhänge</i>
Modelle	Formalisierung von Managementobjekten und Zusammenhängen	Bestandteile einer Managementarchitektur
Managementarchitektur	Gesamtheit aller Modelle zum Entwurf eines (IT-gestützten) Managementsystems	liefert Eingaben (Inputs) für ein konkretes Managementsystem/eine konkrete Managementplattform
(Management-) Prozess	Satz von in Wechselbeziehung stehenden Aktivitäten	Teil eines Managementsystems
Managementsystem	Gesamtheit aller Richtlinien, Prozesse, Verfahren, Ressourcen	Unterstützung von Managementaufgaben, realisiert (vorwiegend organisatorische) Konzepte der Managementarchitektur
Managementplattform	Laufzeit-/Entwicklungs-umgebung für Managementanwendungen	Basis für ein IT-gestütztes Managementsystem, realisiert (vorwiegend systemtechnische) Konzepte der Managementarchitektur
IT-gestütztes Managementsystem	Managementsystem + unterstützende Managementplattform	

Tabelle 2.1: Methodische Konzepte und Begriffe im Überblick

2.1.4 Zusammenfassung: Methodische Grundlagen

Tabelle 2.1 fasst alle wesentlichen methodischen Konzepte und Begriffe dieses Abschnitts zusammen und beschreibt abschließend noch einmal ihre wechselseitigen Zusammenhänge und Abhängigkeiten.

Damit ist das methodische Fundament für diese Arbeit weitgehend vollständig gelegt, und es ist im Folgenden – aufbauend auf diesen Vorüberlegungen –

festzulegen, *wie genau* die intendierte SLM-Architektur entwickelt und strukturiert werden kann. Dazu wird im nächsten Abschnitt die Systematik des Architekturentwurfs unter Angabe eines Ordnungs- sowie eines Entwicklungsrahmens beschrieben.

2.2 Systematik des Architekturentwurfs

Es wurde dargestellt, wie Managementsystem (als übergeordnetes Ziel) und Managementarchitektur (als der Weg dorthin) zusammenhängen. Dieser Abschnitt adressiert nun zwei wichtige methodisch-systematische Aspekte der Arbeit: den Ordnungs- und den Entwicklungsrahmen. Der *Ordnungsrahmen* gibt an, wie die Managementarchitektur strukturiert ist, der *Entwicklungsrahmen* definiert die Schrittfolge, die „abgearbeitet“ werden muss, um zur fertigen Architektur zu gelangen.

2.2.1 Ordnungsrahmen

Die Idee eines Teilmodell-basierten Ansatzes beim Architekturentwurf wurde bereits in Kapitel 1 vorgestellt. Sie resultiert in einer Partitionierung der Architektur in ein Organisationsmodell, ein Funktionsmodell, ein Informationsmodell und Kommunikationsmodell. Seinen Ursprung hat dieser Ansatz in der OSI-Managementarchitektur für ein integriertes Management vernetzter Systeme. Abbildung 2.3 illustriert in sehr abstrakter Weise eine der wesentlichen Grundideen der OSI-Management-Architektur: das Management durch den Zugriff auf Management-Informationsbasen (MIBs).

Das Manager-Agent-Prinzip realisiert in diesem Zusammenhang einen streng hierarchischen Management-Ansatz, in dem Manager und Agent Rollen sind, die sich nicht gegenseitig ausschließen müssen. Der Manager ist die Komponente, die managementrelevante Aktionen in aller Regel anstößt, der Agent ist die Komponente, die sie auf dem (ggf. entfernt) gemanagten System ausführt. Über die MIB-Zugriffsschnittstelle greift der Agent auf die Managementobjekte (*Managed Objects*) zu, um diese – je nach Zielsetzung – entweder zu überwachen (*Monitoring*) oder zu steuern/manipulieren (*Controlling*). Im ersten Fall werden Statusabfragen abgesetzt, im zweiten Fall Steuerinformationen ausgetauscht. Die Initiative dabei ergreift im Allgemeinen der Manager. Anders bei asynchronen Ereignismeldungen, die vom gemanagten System selbst ausgehen und ebenfalls über diese dritte Schnittstelle laufen. Betrachtet man die Schnittstellen in diesem abstrakten Modell genauer, so werden unmittelbar drei der

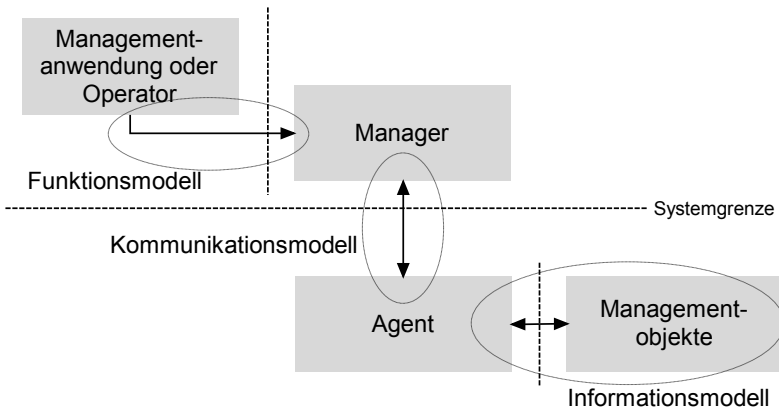


Abbildung 2.3: OSI-Management: Management durch MIB-Zugriff

vier Teilmodelle einer Managementarchitektur sichtbar. Es fehlt in dieser Darstellung lediglich das Organisationsmodell, welches eine eher übergeordnete Rolle einnimmt.

Das Funktionsmodell beantwortet einerseits die Frage, wie die Schnittstelle zwischen Managementanwendung und Manager konkret aussieht, und legt andererseits die Funktionalität der *Managing Entity* fest. Also: Welche Funktionen können im Rahmen einer Managementapplikation angestoßen und durch den Manager ausgeführt werden? Der Gesamtaufgabenkomplex wird dazu in Teilbereiche zerlegt, für die allgemeine Managementfunktionen festgelegt werden. Das Kommunikationsmodell klärt, wie Manager und Agent miteinander kommunizieren können. Das schließt die Definition eines Kommunikationsprotokolls und der damit verbundenen Nachrichtenformate mit ein. Relevant sind alle oben genannten Kommunikationsarten: Überwachung, Steuerung und asynchrone Ereignismeldungen. Das Informationsmodell legt fest, wie Informationen zu den Managementobjekten modelliert und gespeichert werden und wie auf diese Informationen zugegriffen werden kann.

Da im Dienstmanagement heute noch keine Managementarchitekturen im Sinne der Definition dieser Arbeit existieren, stellen sich nun folgende Fragen:

- Lässt sich der OSI-Architekturansatz auf den Bereich des Dienstmanagements übertragen?

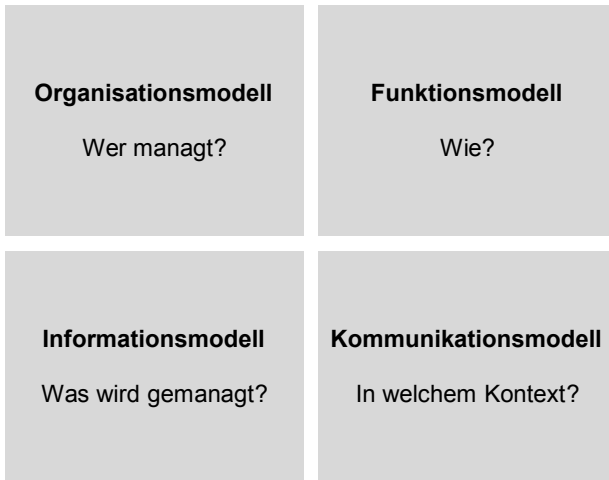


Abbildung 2.4: Zielsetzungen der Teilmodelle für Architekturen im Dienstmanagement

- Können Konzepte wie der Teilmodell-basierte Ansatz übernommen werden?
- Welche Anpassungen sind gegebenenfalls notwendig?

Untersucht man die *Management durch MIB-Zugriff*-Idee vor dem Hintergrund dieser Fragestellungen, so kommt man zu folgendem Ergebnis: Grundsätzlich macht es Sinn, die gleichen vier Teilmodelle auch im Dienstmanagement allgemein bzw. im Service-Level-Management als konkrete Disziplin zu betrachten, um ein strukturiertes und zielorientiertes Vorgehen zu realisieren. Auch im Hinblick auf eine spätere Plattformrealisierung macht es Sinn, funktionale Aspekte der Architektur und die Informationsmodellierung voneinander zu entkoppeln (mehr dazu in Kapitel 7). Ein modularer Architekturentwurf hilft außerdem, die Komplexität der Aufgabe beherrschbarer zu machen. Abbildung 2.4 zeigt in diesem Zusammenhang, welche Fragestellungen durch die Teilmodelle jeweils adressiert werden.

Allerdings muss auch festgehalten werden, dass die *Managed Objects* im Service-Level-Management (zum Beispiel Dienstkataloge, Dienstvereinbarungen oder Service-Level-Berichte) in der Regel Informationsartefakte sind, wäh-

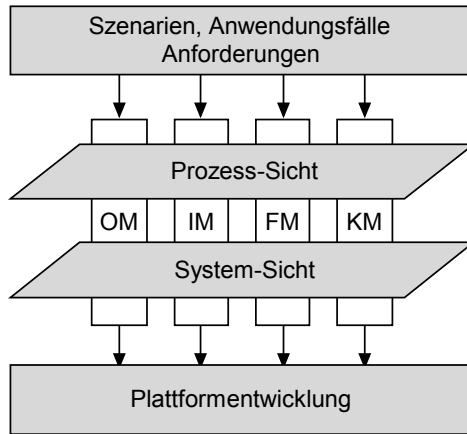


Abbildung 2.5: Ordnungsrahmen der Architektur

rend im klassischen Netz- und Systemmanagement konkrete Ressourcen (Komponenten) gemanagt werden. Das führt dazu, dass es nicht zwangsläufig eine klare Systemgrenze gibt, die Manager und Agent voneinander trennen. Darüber hinaus wurde bereits in Kapitel 1 begründet, dass eine Trennung zwischen einer Prozess- und einer System-Sicht sinnvoll ist, um den unterschiedlichen Modellanforderungen dieser beiden Perspektiven gerecht zu werden.

Verbindet man nun den Teilmodell-basierten Ansatz mit der dualen Sicht auf die plattformunabhängigen Architekturmodelle, so ergibt sich der in Abbildung 2.5 dargestellte Ordnungsrahmen als Verfeinerung des Top-Down-Vorgehens vom Beginn dieses Kapitels (vgl. Abbildung 2.2).

Jeder Bestandteil der in den Folgekapiteln vorgestellten Architektur lässt sich in diesen Ordnungsrahmen einbetten, indem er einer der horizontalen Ebenen und – im Falle der Kapitel 5 und 6 – einer der vier vertikalen Reihen zugeordnet wird.

2.2.2 Entwicklungsrahmen

Während der Ordnungsrahmen den statischen Aufbau der Managementarchitektur beschreibt, ist es das Ziel des Entwicklungsrahmens, darzustellen, wie der Ordnungsrahmen mit Inhalten gefüllt werden kann:

- Wie werden Anforderungen an die Architektur ermittelt? Und wie fließen diese Anforderungen in die Architekturteilmodelle ein? → Systematik der Anforderungsanalyse
- Welche Zusammenhänge bestehen zwischen den Teilmodellen der Prozess-Sicht und denen der System-Sicht? → Systematik des Modellentwurfs

2.2.2.1 Systematik der Anforderungsanalyse

Abbildung 2.6 visualisiert zunächst die Systematik der Anforderungsgewinnung: Aus verschiedenen IT-Provider-Szenarien, beschrieben in Kapitel 3.1, werden allgemeine funktionale Anforderungen an ein Managementsystem im Bereich Service-Level-Management abgeleitet. In der Gesamtheit bilden diese das abzudeckende *Funktionsspektrum*. Jede enthaltene Anforderung wird im Kontext eines typischen Dienstlebenszyklus untersucht, wodurch sichergestellt wird, dass alle Phasen, die ein Dienst durchlaufen kann, vom Funktionsspektrum berücksichtigt werden. Dies geschieht in Kapitel 3.2. Schließlich werden in Kapitel 3.3 aus dem Funktionsspektrum konkrete Anwendungsfälle (*Use Cases*) für ein SLM-unterstützendes Managementsystem abgeleitet.

Dabei wird also die Frage beantwortet: Welche SLM-spezifischen Aufgaben muss oder sollte ein integriertes, IT-gestütztes Managementsystem für SLM unterstützen können? Unterschiedliche Szenarien können völlig disjunkte, teilweise kongruente oder völlig identische Anwendungsfälle hervorbringen. Alle Anwendungsfälle werden in einheitlicher Weise dokumentiert. Gemeinsam bilden sie gewissermaßen die funktionale Spezifikation eines Managementsystems für SLM und somit auch den Ausgangspunkt für den Entwurf der Teilmodelle einer Managementarchitektur, die gewährleisten muss, dass alle Anwendungsfälle umfassend adressiert werden.

2.2.2.2 Systematik des Modellentwurfs

Durch den Ordnungsrahmen wurde bereits festgelegt, wie die Kernarchitektur aufgebaut wird. Die horizontale Differenzierung nach Prozess- und System-Sicht sowie die vertikale Einteilung in ein Organisations-, ein Funktions-, ein Informations- und ein Kommunikationsmodell sind hierfür entscheidend. Man erhält also insgesamt acht verschiedene Felder, in denen Modellelemente angeordnet werden, die in wechselseitiger Beziehung stehen. Abbildung 2.7 stellt diese Situation grafisch dar, wobei die Sichten aus Gründen der Darstellbarkeit

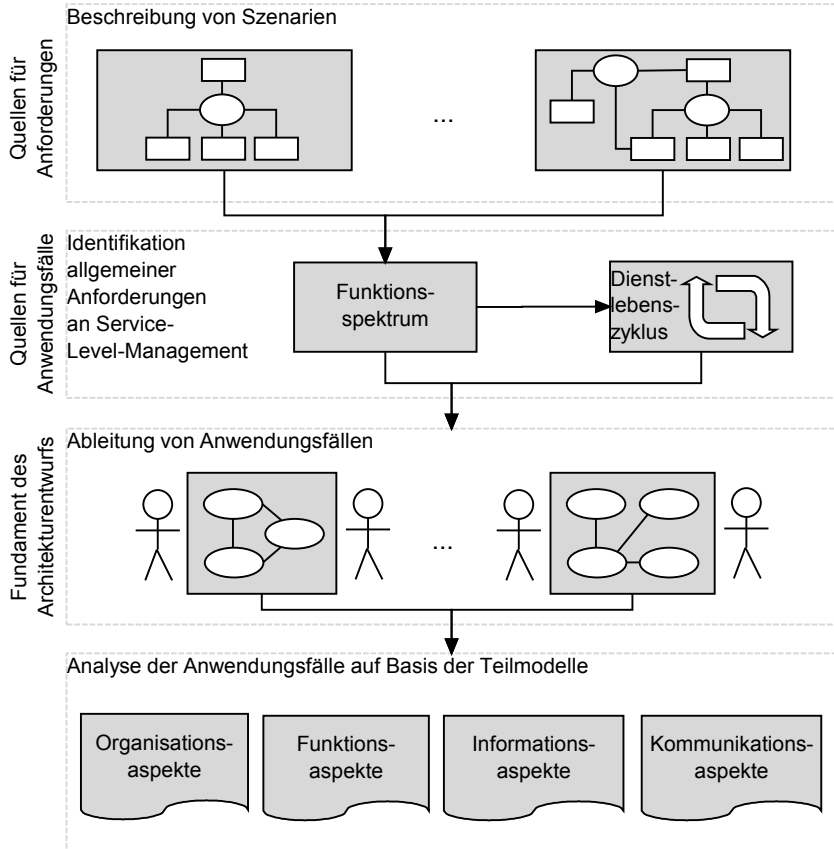


Abbildung 2.6: Entwicklungsrahmen, Teil 1: Systematik der Anforderungsanalyse

ausnahmsweise vertikal und die Teilmodelle horizontal abgebildet sind. Diese Abbildung illustriert auch die Zusammenhänge zwischen den unterschiedlichen Teilmodellen der beiden Ebenen und formt damit den zweiten Teil des Entwicklungsrahmens.

Bevor nun Anforderungsanalyse und Architekturentwurf gemäß des Entwicklungsrahmens durchgeführt werden, werden noch die wichtigsten Begriffe und Konzepte aus dem Umfeld des Service-Level-Managements erörtert und in Beziehung zueinander gesetzt.

2.3 Begriffsbildung

Ziel dieses Abschnitts ist die Schaffung einer einheitlichen und umfassenden Nomenklatur, die das begriffliche Fundament für die gesamte Arbeit im Allgemeinen und für die Anforderungsanalyse im Speziellen bildet. Verwandte Forschungsarbeiten im Bereich Service-Level-Management werden noch nicht näher untersucht, da diese für die Ermittlung der Anforderungen an dieser Stelle noch keine Rolle spielen. Insbesondere soll die Anforderungsanalyse ja gerade möglichst objektive Kriterien im Sinne des Top-Down-Ansatzes hervorbringen, die nicht beeinflusst, geprägt oder ausgerichtet sind auf ein bestimmtes Prozessrahmenwerk oder einen konkreten Modellansatz.

Bei der Begriffsbildung allerdings wäre es unklug, verbreitete Rahmenwerke zu ignorieren und von etablierter Terminologie abzuweichen oder diese „neu zu erfinden“. Insofern orientieren sich die Begriffsfestlegungen für diese Arbeit zumindest in Teilen an den erst im Kapitel 4 noch detaillierter vorgestellten Arbeiten und Rahmenwerken.

Im Folgenden wird zunächst der Dienstbegriff diskutiert, bevor die wichtigsten und grundlegenden Begriffe aus der Disziplin des Service-Level-Managements definiert werden. Dabei wird zwischen Begriffen differenziert, die überwiegend für den Dienstanbieter (Provider) relevant sind, und solchen, die eher für den Dienstnehmer (Kunden) eine wichtige Rolle spielen. Den Abschluss dieses Abschnitts bilden Begriffe, die auf der Schnittstelle zwischen Kunde und Provider liegen.

2.3.1 Der Dienstbegriff

Im Zusammenhang mit der Disziplin des Dienstmanagements und seiner Teildisziplinen, zu denen auch das Service-Level-Management gehört, nimmt das

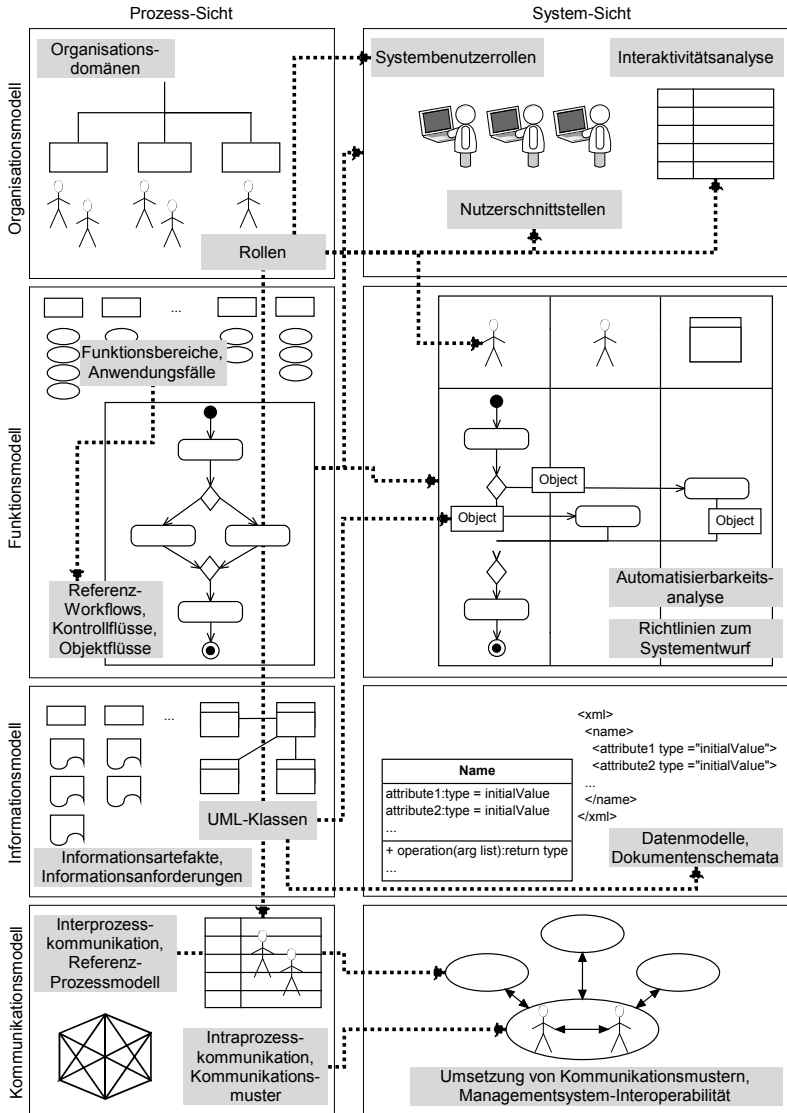


Abbildung 2.7: Entwicklungsrahmen, Teil 2: Zusammenhänge der Teilmodelle

Konzept des IT-Dienstes (*IT service*) eine zentrale und entscheidende Rolle ein. Eine klassische und einfache Definition des Dienstbegriffs ist die folgende:

Dienst – systemtechnische Definition Ein Dienst ist eine Funktionalität, die von einem Objekt an einer Schnittstelle angeboten wird.

Eine zweite Definition, die weniger die funktionalen Eigenschaften eines Dienstes als vielmehr seinen intendierten Wert und die in die Erbringung involvierten Parteien in den Vordergrund rückt, liefert die *IT Infrastructure Library* (ITIL) [25]:

Dienst – wertorientierte Definition Ein IT-Dienst ist ein Hilfsmittel, das einem Kunden von einem Provider bereitgestellt wird, um diesem die Erreichung seiner Ziele zu ermöglichen, ohne dass er selbst als Verantwortlicher von Kosten und Risiken auftritt.

Die wertorientierte Definition ist gerade aus Sicht des Service-Level-Managements interessant. Abbildung 2.8 illustriert in Anlehnung an *ITIL Service Strategy* [2], wie durch einen IT-Dienst Wert (*value*) für einen Kunden generiert wird. Dabei wird zwischen zwei Dimensionen differenziert: dem Nutzwert (*utility*) und dem Leistungswert (*warranty*).

Der Nutzwert eines Dienstes ergibt sich für einen Kunden daraus, dass durch den IT-Dienst entweder der Durchsatz bestimmter Teile von geschäftsrelevanten Aktivitäten (Synonym: Geschäftsprozessen, Wertschöpfungsprozessen) verbessert wird oder dass durch ihn bestimmte Hemmnisse oder Grenzen reduziert oder abgebaut werden, die ohne den IT-Dienst weiter bestehen würden. Beide Aspekte schließen sich nicht aus; insofern handelt es sich nicht um ein exklusives Oder, das diese Möglichkeiten verknüpft. Ein Beispiel für den ersten Fall – also Durchsatz-/Leistungssteigerung – ist jeder IT-Dienst, der zur Automatisierung bestimmter Abläufe beisteuert und somit dazu beiträgt, dass eine Dienstleistung effizienter erbracht oder ein Produkt schneller oder mit geringerem Ressourceneinsatz (insbesondere im Hinblick auf Personalressourcen) produziert werden kann. Der zweite Aspekt – Abbau von Hemmnissen – wird zum Beispiel durch alle Arten von Kommunikationsdiensten realisiert. Durch sie werden räumlich-geographische Barrieren innerhalb von Kommunikationsbeziehungen in den geschäftlichen Aktivitäten dezimiert.

Der Leistungswert eines Dienstes ergibt sich für einen Kunden daraus, unter welchen Rahmenbedingungen und in welcher Qualität der Nutzwert des

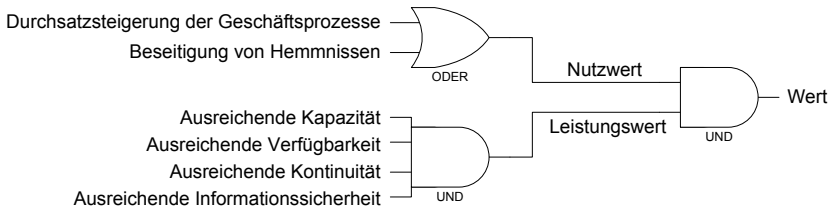


Abbildung 2.8: Wert-Dimensionen eines IT-Dienstes nach ITIL [2]

Dienstes erbracht wird. Kapazität, Verfügbarkeit, Kontinuität im Katastrophenfall und Informationssicherheit im Rahmen der Diensterbringung sind die wesentlichen Charakteristika, die den Leistungswert kennzeichnen.

Warum ist diese wertorientierte Definition des Dienstbegriffs interessant, wenn es um Service-Level-Management geht? Der Grund ist, dass das Service-Level-Management alle Dimensionen des Wertes, der durch einen Dienst generiert wird, berücksichtigen muss. Vielfach wird aber Service-Level-Management nur mit dem Leistungswert und dort hauptsächlich mit dem Aspekt der Dienstverfügbarkeit in Verbindung gebracht. Die wertorientierte Definition macht deutlich, dass das nicht ausreicht.

Beide vorgestellten Definitionen des Dienstbegriffs – die systemtechnische sowie die wertorientierte Definition – haben ihre Berechtigung und beleuchten den Dienstbegriff von jeweils unterschiedlichen Perspektiven. Für eine vertiefende Auseinandersetzung mit Aspekten des dienstorientierten IT-Managements, wie sie mit dieser Arbeit angestrebt wird, sind diese Definitionen jedoch zu oberflächlich und nicht ausreichend erschöpfend.

Eine deutlich tiefergehende, begriffsbildende Annäherung liefert zu diesem Zweck das MNM-Dienstmodell [26, 27], dargestellt in Abbildung 2.9. Ziel dieses Modells ist es, den Dienstbegriff selbst sowie die damit in engem Zusammenhang stehenden Rollen und Konzepte zu definieren. Das Modell vermittelt einen technologieunabhängigen Zugang zu den Grundlagen des Dienstmanagements und ein wichtiges Fundament für diese Arbeit.

Dienst – modellbasierte Definition Das MNM-Dienstmodell definiert einen Dienst als Abstraktion über Komponenten und Subdiensten, die sich im Wesentlichen aus einer Nutzungsfunktionalität, einer Managementfunktionalität und spezifischen Qualitätsmerkmalen (QoS-Parameter) zusammensetzt. In dieser Eigenschaft wird ein Dienst von einem Provider

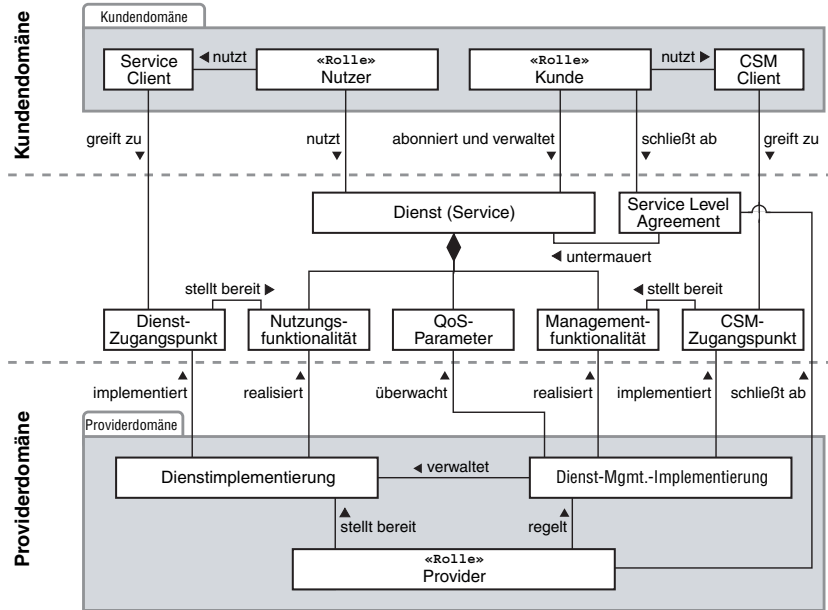


Abbildung 2.9: Das MNM-Dienstmodell

mittels (informations-)technischer Hilfsmittel implementiert, realisiert, überwacht und an eine Kundendomäne ausgeliefert.

Das dieser Definition zugrunde liegende Rollenmodell beschränkt sich auf die Domänen *Provider* und *Kunde* und differenziert in der Kundendomäne zwischen dem Kunden als Auftraggeber und Manager des Dienstes im Sinne eines *Customer Service Managements* (CSM) sowie dem Nutzer, der auf den Dienst über einen Dienst-Client zugreift. Für eine detaillierte Analyse der Kommunikationsbeziehungen im Zusammenhang mit der Disziplin des Service-Level-Managements wird dieses einfache Rollenmodell zwar die Grundlage liefern, mit seinen drei Rollen allerdings nicht ausreichend sein. Vielmehr muss insbesondere in der Domäne des Providers weiter differenziert werden. Dies ist aber Aufgabe des Organisationsmodells der Prozess-Sicht der Architektur (Kapitel 5.1).

Es folgen einige weitere wichtige Begriffsdefinitionen aus dem Kontext des Dienstmodells.

Dienstqualitätsparameter (*Quality of Service Parameter*) Sowohl die vereinbarte (aus Provider-Sicht: garantierte) als auch die tatsächlich erreichte (gemessene) Dienstqualität wird durch eine Menge von QoS-Parametern determiniert. Üblicherweise spielen QoS-Parameter beim Abschluss von Dienstvereinbarungen eine große Rolle, indem über sie die Service-Levels definiert werden. Dies kann prinzipiell geschehen durch Angabe von

- oberen/unteren Schranken,
- Wertebereichen (Intervallen) oder
- diskreten Werten.

Im Zusammenhang mit der Dienstqualität ist neben der Definition von QoS-Parametern die Definition des Messverfahrens bzw. der Messtechnik und der eingesetzten Werkzeuge mindestens ebenso wichtig. Beides – die Parameter und die zugehörigen Messverfahren – sollte Teil der Dienstvereinbarungen sein.

Dienstzugangspunkt (*Service Access Point, SAP*) Der Dienstzugangspunkt bezeichnet den physischen oder logischen „Ort“ der Dienstleistung. Er stellt einen wichtigen Inhalt jeder Dienstvereinbarung dar, da die Dienstleistung in nahezu allen Fällen auf bestimmte Dienstzugangspunkte beschränkt ist, was sich auf Nutzung und Qualität des Dienstes auswirkt.

Beispielhaft sei an dieser Stelle ein drahtloser Netzdienst (*Wireless LAN*) genannt, der über verschiedene Funkzugangsknoten von Anwendern über einen entsprechend vorkonfigurierten Software-Client genutzt werden kann. Die Anzahl und Verteilung dieser Knoten, die in diesem Beispiel als Dienstzugangspunkte fungieren, beeinflusst maßgeblich die Kapazität und Qualität des Dienstes (wie viele Nutzer können mit welcher Datenrate auf das Funknetz zugreifen?) sowie die geographischen Orte (Gebäude, Räumlichkeiten), an denen der Dienst überhaupt in ausreichender Qualität verfügbar ist.

Eine Dienstvereinbarung wird also nur mittels konkreter Angaben zu dem oder den Dienstzugriffspunkten auf ein belastbares Fundament gestellt. Im Beispiel ist jeder Dienstzugangspunkt zugleich eine Hardware-Komponente. Dies kann für den allgemeinen Fall jedoch nicht angenommen werden.

Dienst-Client Mit dem Dienst-Client wird vom Anwender auf einen Dienst zugegriffen, indem eine physische oder logische Verbindung zu einem

Dienstzugangspunkt hergestellt wird. Es handelt sich wieder um ein abstraktes Konzept, und nicht in jedem Fall muss der Dienst-Client ein Software-Client im klassischen Sinne sein. Im Gegensatz zum Funknetz-Beispiel sei an dieser Stelle das Beispiel „Einrichtung eines Standard-EDV-Arbeitsplatzes“ genannt. Der Dienst-Client kann im einfachsten Fall ein Telefon oder ein Web-Formular sein, über das die Erbringung des Dienstes vom dazu berechtigten Anwender angestoßen wird.

Managementfunktionalität Durch *Customer Service Management* (CSM) wird die Idee verfolgt, einen Kunden durch die Bereitstellung geeigneter Werkzeuge und Schnittstellen in die Lage zu versetzen, die von ihm beauftragten IT-Dienste selbst zu steuern und zu überwachen (vgl. Langer [28]). Ob auf diese Weise sämtliche Diensteigenschaften und Attribute durch den Kunden gesteuert werden können oder ob sich das CSM auf die Bereitstellung kundenspezifisch aufbereiteter Monitoring-Informationen beschränkt, hängt dabei von der tatsächlichen Implementierung der Managementfunktionalität ab.

CSM-Zugangspunkt Analog zum Dienstzugangspunkt beschreibt der CSM-Zugangspunkt den physischen oder logischen „Ort“ der Bereitstellung der Managementfunktionalität des Dienstes.

CSM-Client In Analogie zum Dienst-Client wird mittels des CSM-Clients vom Kunden auf die Managementschnittstelle des Dienstes zugegriffen.

Im Ergebnis liefert das MNM-Dienstmodell eine fundierte und grundsteinlegende Diskussion des Dienstbegriffs, wie er in dieser Arbeit verwendet wird. Es wird allerdings nochmals darauf hingewiesen, dass durch diesen Begriff das gesamte Spektrum IT-gestützter Dienstleistungen abgedeckt wird – also klassische Netz- und Applikationsdienste, aber auch Dienste wie Hardware-Bereitstellung oder Support. Diese Feststellung erweist sich gerade im Kontext des Service-Level-Managements als bedeutsam.

Wichtig ist ebenfalls festzustellen, dass das Dienstmodell ausschließlich statische Aspekte von Diensten beleuchtet, wie zum Beispiel: Welche Rollen und Konzepte bilden den Kontext der Diensterbringung? Und welche Zusammenhänge bestehen zwischen diesen Entitäten? Doch natürlich unterliegt jeder Dienst einer gewissen Dynamik, die durch einen Dienstlebenszyklus beschrieben werden kann.

Dienstlebenszyklus (*Service Lifecycle*) Ein typischer Dienstlebenszyklus besteht aus den Phasen Planung, Verhandlung, Bereitstellung, Betrieb,

Anpassung und Auflösung (vgl. Dreo [4]). Ein in dieser Weise definierter Lebenszyklus stellt ein generisches Zustandsmodell für Dienste mit definierten Zustandsübergängen dar.

Kapitel 3.2.2 greift das Konzept des Dienstlebenszyklus im Rahmen der Anforderungsanalyse auf und setzt die ermittelten funktionalen Anforderungen in den Kontext desselben. In diesem Zusammenhang werden die sechs Lebenszyklusphasen in Anlehnung an Dreo detaillierter beschrieben und zusammengefasst, weshalb an dieser Stelle darauf verzichtet wird.

In den nun folgenden Abschnitten werden wichtige und etablierte Konzepte aus der Disziplin des Service-Level-Managements vorgestellt und definiert.

2.3.2 Begriffe der Providerdomäne

Die in diesem Abschnitt definierten Begriffe aus der Disziplin Service-Level-Management beschäftigen sich mit Konzepten, die für SLM-spezifische Aufgaben in der Provider-Domäne wichtig sind, für den Dienstnehmer aber keine direkte Bedeutung haben.

Dienstportfolio (*Service Portfolio*) Das Dienstportfolio bezeichnet die Gesamtheit aller von einem Provider verwalteten (gemanagten) Dienste. In dieser Eigenschaft ist das Dienstportfolio als Verzeichnis aller aktiven und inaktiven, geplanten und tatsächlich betriebenen Dienste ausschließlich dem Provider zugänglich. Es enthält zu jedem Dienst alle managementrelevanten Informationen – darunter eine Beschreibung der Funktionalität, Festlegungen von Dienstmetriken, technische Spezifikationen, eine Diensthistorie und weitere relevante Daten.

Lieferantenvertrag (*Underpinning Contract, UC*) Provider, die IT-Dienste von externen Dienstleistern beziehen, um dadurch ihre eigenen Dienste zu realisieren oder zu unterstützen, nehmen gegenüber den externen Zulieferern selbst eine Kundenrolle ein. Die hierbei geschlossenen Verträge sind grundsätzlich Dienstvereinbarungen (*Service Level Agreements, SLAs*) im Sinne der noch folgenden Definition. Sie werden allerdings als Lieferantenverträge (*Underpinning Contracts*, „zugrunde liegende Verträge“) bezeichnet, um sie explizit von den Dienstvereinbarungen zwischen dem betrachteten Provider und seinen eigenen Kunden zu unterscheiden. Aus zwei Gründen macht diese Trennung Sinn: Zum einen sind UCs im Unterschied zu SLAs für einen Kunden transparent. Zum zweiten ist es für einen Provider entscheidend wichtig, die Abhängigkeiten,

die sich zwischen UCs und SLAs einstellen, bei der Verhandlung beider Vertragsarten in sinnvoller Weise zu berücksichtigen.

Operative Vereinbarung (*Operational Level Agreement, OLA*) Operative Vereinbarungen sind Vereinbarungen zur Sicherstellung von Zulieferleistungen, die für die Erfüllung von SLAs erforderlich sind. Damit weisen sie in ihrer Zielsetzung eine Ähnlichkeit zu UCs auf. Allerdings werden OLAs eingesetzt, um providerinterne IT-Fachabteilungen zu verpflichten, bestimmte Leistungen zu festgelegten Bedingungen und Konditionen zu erbringen, um so ihren Beitrag zur Einhaltung der SLAs zu leisten.

2.3.3 Begriffe der Kundendomäne

Aus der Perspektive des Kunden ist im Kontext von IT-Diensten vor allem ein Aspekt von vorrangiger Bedeutung: dass die durch den Provider bereitgestellten IT-Dienste die geschäftlichen Aktivitäten, die in der Verantwortung des Dienstnehmers stehen, „optimal“ unterstützen. Im Folgenden werden zwei Begriffsdefinitionen gegeben, die sich in diesen Zusammenhang einordnen.

Geschäftsprozess (*Business Process*) Das Kerngeschäft eines Unternehmens wird durch seine Geschäftsprozesse (Synonym: Wertschöpfungsprozesse) realisiert. Im Zusammenhang mit Service-Level-Management ist die Betrachtung von Geschäftsprozessen von großer Bedeutung, da diese kundenseitig im Mittelpunkt des Interesses stehen, während IT-Dienste als unterstützende Komponenten nur indirekt an den eigentlichen Geschäftszielen beteiligt sind. Jeder Geschäftsprozess lässt sich durch die folgenden beiden Eigenschaften charakterisieren und insbesondere von unterstützenden (IT-)Diensten abgrenzen:

- Er trägt unmittelbar zur Vorbereitung, Herstellung, Lieferung oder Unterstützung eines Produktes oder einer Dienstleistung bei.
- Er trägt auf diese Weise direkt und messbar zur Wertschöpfung und somit zum monetären Umsatz und Gewinn des Unternehmens bei.

Geschäftsorientierung der IT (*Business IT Alignment*) Unter Geschäftsorientierung der IT oder auch *Business-driven IT Management* (BDIM) versteht man alle Maßnahmen, die dazu beitragen, Ziele, Leistungen und Prozesse zwischen der IT-Organisation (Provider) und den Fachbereichen eines Unternehmens aufeinander abzustimmen. Hintergrund ist

die enge Verbindung zwischen den unternehmerischen Wertschöpfungsprozessen und den IT-Diensten, die eben diese Prozesse unterstützen. Beispiele für konkrete Maßnahmen sind die Abbildung von Leistungsindikatoren der IT-Ebene auf Leistungsindikatoren der Geschäftsprozessebene oder auch die Bestimmung der Opportunitätskosten im Falle eines IT-Ausfalls oder der Degradierung von IT-Diensten. Service-Level-Management ist als Dienstmanagementdisziplin in besonderer Weise von dem Aspekt der Geschäftsorientierung der IT tangiert, da die abgeschlossenen Dienstvereinbarungen und die darin enthaltenen Qualitätsvereinbarungen bezüglich bestimmter IT-Dienste deren Beitrag zu den Geschäftsprozessen reflektieren sollten. Diesen Punkt bei SLA-Abschlüssen zu berücksichtigen liegt in erster Linie in der Verantwortung des Kunden.

2.3.4 Begriffe auf der Schnittstelle zwischen Provider und Kunde

Im Sinne des MNM-Dienstmodells können die im Folgenden definierten Begriffe auch als domänenunabhängig angesehen werden. Sie füllen also den Bereich zwischen der Provider- und der Kundendomäne – im Original-Modell auch *side independent* genannt. Das bedeutet allerdings nicht etwa, dass sich weder Kunde noch Provider für die folgenden Artefakte interessieren oder verantwortlich fühlen. Das Gegenteil ist der Fall: Sie spielen aus der Perspektive beider Domänen eine entscheidende Rolle zur Untermauerung der Beziehungen zwischen den Domänen und realisieren in dieser Eigenschaft die Schnittstelle zwischen ihnen.

Dienstkatalog (*Service Catalog*) Ein Dienstkatalog ist ein Verzeichnis von Diensten, die von Kunden beauftragt (abonniert) werden können. Das schließt sowohl bereits genutzte als auch noch nicht beauftragte, aber verfügbare und aktive Dienste ein. Somit realisiert der Dienstkatalog eine gegebenenfalls *kundenspezifische Sicht* auf das Dienstportfolio (siehe oben) und ist somit stets eine echte Teilmenge des gesamten Dienstportfolios. Prinzipiell kann ein Provider einen oder mehrere Dienstkataloge bereithalten und pflegen. Im Falle mehrerer Dienstkataloge können unterschiedliche Kataloge beispielsweise für verschiedene Kunden oder Kundengruppen vorgehalten werden. Eine andere Möglichkeit besteht darin, unterschiedliche Kataloge für verschiedene Arten von Diensten – zum Beispiel ein Katalog für Kommunikationsdienste, einer für Serverdienste usw. – vorzuhalten. Auch Kombinationen sind prinzipiell denkbar.¹

¹Die hier formulierte Definition des Begriffs *Dienstkatalog* ist angelehnt an die Definition

Service-Level Ein Service-Level beschreibt Spannen zulässiger Werte für eine Menge von QoS-Parametern und repräsentiert damit umfassend die akzeptable Qualität eines IT-Dienstes.

Dienstvereinbarung (*Service Level Agreement, SLA*) Eine Dienstvereinbarung ist eine schriftliche Vereinbarung in Vertragsform zwischen einem Provider und einem Kunden, in der Dienstleistungen und Service-Levels spezifiziert und dokumentiert sind. Eine Dienstvereinbarung bezieht sich im einfachsten Fall auf die Beauftragung eines Dienstes durch einen Kunden. Im IT-gestützten Service-Level-Management bietet es sich an, aus Dienstvereinbarungen auf Dienstkataloge zu referenzieren.

Der Begriff der Dienstvereinbarung (*Service Level Agreement*) ist, wie zuvor gesehen, Teil des MNM-Dienstmodells. Der Dienstkatalog aber fehlt in diesem Modell. Warum ist das so? Um dies zu verstehen, muss man sich klar machen, welches Ziel das Dienstmodell verfolgt und welchen Betrachtungsbereich es abdeckt: Für den einzelnen Dienst und dessen Erbringung ist das zugehörige SLA von durchaus entscheidender Bedeutung, weil die dort vereinbarten QoS-Garantien maßgeblichen Einfluss auf die technische Dienstleistung haben können.

Der Dienstkatalog spielt in der Kunden-Provider-Kommunikation eine große Rolle, ist aber in der isolierten Betrachtung des einzelnen Dienstes nicht essenziell. Somit ist es keine Unzulänglichkeit des Dienstmodells, dass dieses Artefakt hier nicht berücksichtigt wurde, sondern eine Frage der Perspektive und des Kontextes. Im Zusammenhang mit Service-Level-Management sind beide Konzepte von entscheidender Wichtigkeit.

2.4 Zusammenfassung

In drei Teilen wurde mit diesem Kapitel das methodische und begriffliche Fundament der vorliegenden Arbeit gelegt. Zunächst wurden die *drei Säulen des methodischen Vorgehens* beschrieben: das Prinzip einer modellgetriebenen Architektur, das Paradigma der Prozessorientierung sowie die Verfolgung eines

aus der *IT Infrastructure Library* (ITIL) [25], weicht aber in einem Aspekt leicht von ihr ab: Streng genommen sieht ITIL keine verschiedenen Dienstkataloge im Sinne von Teilsichten auf das Dienstportfolio vor. Diese Einschränkung ist allerdings in einem IT-Umfeld mit mehreren Kunden nicht sinnvoll. Auch ISO/IEC 20000 trifft diese Einschränkung nicht, allerdings enthält der Standard in seinem Abschnitt 2 (*Terms and definitions*) keine Definition für den Begriff des Dienstkatalogs.

systembasierter Ansatzes. Im Anschluss daran wurde die *Systematik des Architekturentwurfs* dargestellt, die aus zwei Teilen besteht: einem Ordnungsrahmen, der die grundsätzliche Struktur der Architektur reflektiert, und einem Entwicklungsrahmen, der einerseits die Systematik der Anforderungsanalyse und andererseits die Zusammenhänge der verschiedenen architekturellen Teilmodelle ausdrückt. Schließlich wurden die *wesentlichen Begriffe*, die mit der Disziplin des Service-Level-Managements in engem Zusammenhang stehen, definiert.

Kapitel 3

Anforderungsanalyse

Inhalt dieses Kapitels

3.1 Szenarien	53
3.1.1 Morphologischer Kasten zur Einordnung der Szenarien	54
3.1.2 Szenario 1: Transportdienstleister (ITG)	59
3.1.3 Szenario 2: Hochschulrechenzentrum (LRZ)	63
3.1.4 Szenario 3: Flughafen (FMG)	66
3.1.5 Vollständigkeit der Szenarien	70
3.2 Funktionale Anforderungen	72
3.2.1 SLM-Zielsetzung und Grundannahmen	74
3.2.2 Definition eines generischen Dienstlebenszyklus	76
3.2.3 Funktionale Anforderungen an Service-Level-Management	78
3.2.4 Auswertung der allgemeinen Funktionsanforderungen	96
3.2.5 Zusammenfassung und nächste Schritte	99
3.3 Primäranwendungsfälle	99
3.3.1 Definition der Anwendungsfälle	101
3.3.2 Zusammenhänge zwischen den Anwendungsfällen	117
3.4 Nicht-funktionale Anforderungen	127
3.5 Zusammenfassung und nächste Schritte	130

Ziel dieses Kapitels ist es, die Anforderungen an den architekturbasierten Ansatz zum integrierten Service-Level-Management zu bestimmen. Abbildung 3.1 zeigt, wie sich dieses Kapitel damit in den Ordnungsrahmen der Architektur einordnet. In seiner Struktur folgt das Kapitel der in Kapitel 2.2.2.1 beschriebenen Systematik der Anforderungsanalyse als Teil des Entwicklungsrahmens der Architektur.

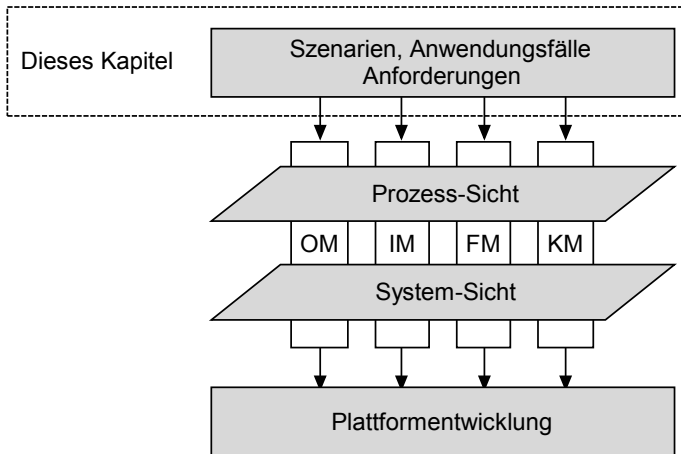


Abbildung 3.1: Einordnung dieses Kapitels in den Ordnungsrahmen

Ein in der Softwareentwicklung gängiges Vorgehen zur Anforderungsanalyse ist es, Anwendungsfälle zu beschreiben, die durch das intendierte Softwaresystem abgedeckt werden sollen. Die Anwendungsfälle, die im Abschnitt 3.3 entwickelt werden, dienen als funktionale Anforderungen an ein konkretes IT-gestütztes Managementsystem sowie an die im Rahmen dieser Arbeit zu entwickelnde Architektur gleichermaßen. Denn letztlich muss durch das Design der Architektur sichergestellt sein, dass ein gemäß den Vorgaben der Architektur entwickeltes Managementsystem mit zugehöriger Plattform den funktionalen Anforderungen genügt.

Um Anwendungsfälle systematisch, strukturiert und vor allem zielorientiert ermitteln zu können und dabei sicherzustellen, dass sich die Anwendungsfälle mit den Erfordernissen der Praxis decken, wird der Ansatz einer *szenariobasierten Anforderungsanalyse* gewählt: Dabei werden reale Unternehmens- und IT-Szenarien zugrunde gelegt, in denen SLM-bezogene Anforderungen identifiziert werden. Indem mehrere unterschiedliche Szenarien herangezogen werden, soll erreicht werden, dass ein möglichst breites Spektrum an Anwendungsfällen berücksichtigt wird. Klar ist aber, dass Vollständigkeit im Sinne einer allumfassenden Menge an Anwendungsfällen nur schwer erreicht und noch schwerer nachgewiesen werden kann.

Im weiteren Verlauf des Kapitels werden darüber hinaus nicht-funktionale An-

forderungen entwickelt, die von der Architektur erfüllt werden müssen. Hierbei werden Aspekte wie beispielsweise Erweiterbarkeit oder Änderbarkeit der Architektur angesprochen – also Anforderungen an die Architektur, die nichts mit den funktionalen Eigenschaften eines Managementsystems zu tun haben und auch nicht unbedingt SLM-spezifisch sind. Abbildung 3.2 zeigt, wie funktionale und nicht-funktionale Anforderungen zusammenhängen.

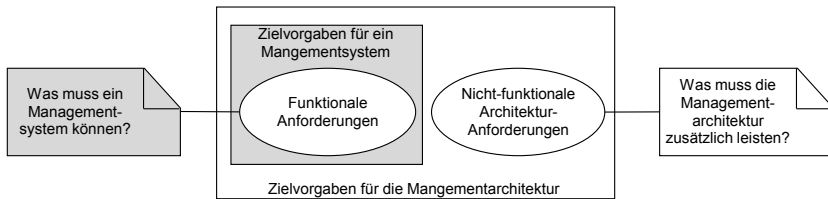


Abbildung 3.2: Abgrenzung der funktionalen und nicht-funktionalen Anforderungen

3.1 Szenarien

Im ersten Schritt werden also verschiedene Szenarien eingeführt und beschrieben. Im Hinblick auf die anschließende Ableitung des Funktionsspektrums und der Identifikation von Anwendungsfällen werden die Szenarien vor allem im Hinblick auf folgende Charakteristika untersucht:

- Geschäftsmodell und Geschäftsprozesse
- Aufbauorganisatorische Aspekte
- Spezifika der IT-Organisation
- Kundenumfeld aus Sicht des IT-Providers
- Lieferantenabhängigkeiten des IT-Providers
- Dienstlandschaft (Dienstportfolio)
- Status Quo des Service-Level-Managements
- Zielsetzungen für das Service-Level-Management

Anzahl/Charakteristik IT-Dienste	1		> 1, homogen		> 1, heterogen	
Anzahl IT-Kunden-Standorte	1			mehrere		
Kundenumfeld	nur interne Kunden		gemischt		nur externe Kunden	
IT-Leistungsverrechnung	nicht geplant		in Planung		vorhanden	
Anzahl IT-Provider-Standorte	1			mehrere		
Lieferantenabhängigkeiten	gering		mittel		stark	
CMMI-Reifegrad ITSM	1	2	3	4	5	
ISO/IEC 20000 Zertifizierung	nicht angestrebt		angestrebt		erreicht	

Abbildung 3.3: Morphologischer Kasten zur Einordnung der Szenarien

Die so aufgebauten Szenariobeschreibungen liefern die Grundlage für eine morphologische Einordnung, die im Anschluss an die Beschreibung für jedes Szenario durchgeführt wird.

3.1.1 Morphologischer Kasten zur Einordnung der Szenarien

Die morphologische Analyse ist eine analytische Methode, die eingesetzt werden kann, um komplexe Problembereiche vollständig zu erfassen und alle möglichen Ausprägungen zu betrachten (vgl. Schulte-Zurhausen [29]). Ziel der morphologischen Einordnung der einzelnen Szenarien ist es, diese anhand definierter Kriterien vergleichbar zu machen und voneinander abzugrenzen. Vor allem aber wird bei der Vollständigkeitsbetrachtung im Rahmen von Abschnitt 3.1.5 Bezug auf die Ergebnisse der szenariospezifischen morphologischen Analysen genommen. Dabei wird die Frage nach der Vollständigkeit der gewählten Szenarien im Hinblick auf die Ableitung von Anforderungen auf die Frage der Plausibilität der Kriterien reduziert, die für die morphologische Analyse herangezogen werden.

Abbildung 3.3 zeigt die verwendete Ausprägung eines so genannten *morphologischen Kastens*, der den Kern jeder Analyseinstanz bildet. Er beschreibt verschiedene Dimensionen, die in der Summe eine aussagestarke Charakterisierung eines Szenarios ermöglichen.

Die hier gewählten Merkmale (Dimensionen) und Abstufungen ihrer jeweili-

gen Ausprägungen stellen nur *eine* sinnvolle Möglichkeit dar, einen morphologischen Kasten zu bilden. Auch andere Varianten können plausibel sein. Die Merkmale sind so gewählt, dass sie in der Vereinigung eine hilfreiche Taxonomie zur Einordnung und Abgrenzung der drei Szenarien aus Sicht des Service-Level-Managements ergeben.

Bei der Festlegung der möglichen Ausprägungen zu einem Merkmal wurden folgende Bedingungen berücksichtigt:

- Verschiedene Ausprägungen schließen sich gegenseitig aus – andernfalls wäre die Abgrenzung der Szenarien auf Basis der Ausprägungen zu einem Merkmal zu unscharf und dadurch nicht zielführend.
- Für jedes Szenario kann die zutreffende Ausprägung jedes Merkmals eindeutig ermittelt werden.
- Unterschiedliche Ausprägungen eines Merkmals führen zu unterschiedlichen Anforderungen an das Service-Level-Management oder zu einer unterschiedlichen Signifikanz der gleichen Anforderung – andernfalls hätte die Unterscheidung zwischen diesen Ausprägungen im Hinblick auf die Anforderungsanalyse keinen Wert.

Zu den ausgewählten Merkmalen im Folgenden einige Erläuterungen:

1. Merkmal: Anzahl/Charakteristik IT-Dienste — Der IT-Provider des betrachteten Szenarios wird dahingehend eingeordnet, ob er im Wesentlichen als *Single Service Provider* auftritt, der auf die Erbringung eines einzigen Dienstes spezialisiert ist, oder ob ein differenziertes Dienstspektrum angeboten wird. Im zweiten Fall wird weiter unterschieden, ob die von unterschiedlichen Kunden beauftragten Dienste eher gleichartig sind, die Anforderungen verschiedener Kunden an die IT-Dienste also weitgehend homogen sind, oder ob sich die Kundenanforderungen stark unterscheiden können.
→ Aus SLM-Sicht führen diese drei Ausprägungen zu jeweils unterschiedlichen Anforderungen an das Service-Level-Management (und damit auch an ein SLM-unterstützendes, IT-gestütztes Managementsystem): Während im ersten Fall nicht einmal ein differenziertes Dienstportfolio zu verwalten ist, wird ein solches im zweiten Fall in jedem Fall benötigt. Im dritten Fall werden sogar mehrere unterschiedliche Dienstkataloge benötigt, um die Anforderungsprofile der verschiedenen Kunden im Hinblick auf benötigte Dienste und Dienstvarianten zu berücksichtigen.

2. Merkmal: Anzahl IT-Kunden-Standorte — Es wird unterschieden, ob alle vom IT-Provider versorgten IT-Kunden an einem einzigen geographischen Standort angesiedelt sind oder ob mehrere Standorte mit Diensten versorgt werden müssen.
 - Der zweite Fall führt dazu, dass gegebenenfalls mehrere Dienstkataloge und SLAs für unterschiedliche Standorte des gleichen IT-Kunden angeboten bzw. abgeschlossen werden müssen. Zusätzlich wirkt sich die geographische Verteilung auf die Mess- und Berichtsanforderungen aus. Die Anforderungen ans Service-Level-Management als Disziplin und somit auch an ein SLM-unterstützendes Managementsystem divergieren also in diesen beiden Fällen.
3. Merkmal: Kundenumfeld — Anhand dieses Merkmals wird zwischen Providern differenziert, die ausschließlich interne Kunden mit IT-Diensten beliefern, solchen, die nur externe Kunden beliefern sowie Providern, die sowohl interne als auch externe Kunden haben. Die Begriffe „intern“ und „extern“ beziehen sich in diesem Zusammenhang darauf, ob Provider und Kunde zur gleichen Organisation gehören oder ob zwischen beiden eine organisatorische Grenze verläuft und damit eine klassische Geschäftsbeziehung besteht.
 - Service-Level-Management gewinnt als Disziplin ganz generell an Bedeutung, je „geschäftsmäßiger“ die Beziehung zwischen einem Provider und seinen Kunden ist. Formale Dienstkataloge sowie eindeutige und (juristisch) bindende SLAs werden tendenziell umso wichtiger, je stärker das Kundenumfeld von externen Kunden geprägt ist.
4. Merkmal: IT-Leistungsverrechnung — Leistungsverrechnung (*Charging*) ist neben Kostenplanung (*Budgeting*) und Kostenrechnung (*Accounting*) die dritte Disziplin im klassischen *Financial Management for IT Services* nach ITIL [30]. Im Gegensatz zur Kostenplanung und -rechnung wird die Verrechnung von IT-Dienstleistungen allerdings als optional erachtet. Es wird bei diesem Merkmal unterschieden, ob ein Provider alle oder einen Teil seiner IT-Dienste weiterverrechnet oder nicht, und falls nicht, ob die Einführung eines Systems zur Leistungsverrechnung für IT-Dienste geplant ist.
 - Leistungsverrechnung stellt besondere Anforderungen an die Formulierung und Ausgestaltung von SLAs und somit auch ans Service-Level-Management. Konkret bedeutet das, dass verrechnungsrelevante Informationen wie Preise und Abrechnungseinheiten (*accountable units*) in SLAs aufgenommen werden müssen. Zudem muss das genaue Verfahren

der Leistungsverrechnung mit dem Kunden vereinbart und dokumentiert werden. Führt ein IT-Provider gar keine IT-Leistungsverrechnung durch, entfällt all das.

5. Merkmal: Anzahl IT-Provider-Standorte — Ob der IT-Provider zentral an einem geographischen Standort oder an mehreren solchen operiert, ist ein weiteres Merkmal, dessen Ausprägung zu unterschiedlichen Anforderungen ans Service-Level-Management führen kann.

→ Die zentrale Erbringung und Administration aller IT-Dienste ist aus SLM-Sicht der einfachere Fall, während sich bei mehreren Standorten zusätzliche Herausforderungen und Anforderungen ergeben. Unter anderem kann es für den Abschluss einer Dienstvereinbarung relevant sein, an welchem oder welchen IT-Standort(en) ein Dienst technisch realisiert und administrativ verwaltet wird. Beispielsweise können verschiedene geographische Standorte über in den Leistungsmerkmalen unterschiedliche WAN-Anbindungen verfügen, die bestimmte Service-Levels der Dienstleistung an bestimmten Standorten ausschließen. Ob Service-Level-Management diesen und andere Aspekte, die mit der IT-Standort-Charakteristik zusammenhängen, berücksichtigen muss, hängt somit von der Ausprägung dieses Merkmals im betrachteten Szenario ab.

6. Merkmal: Lieferantenabhängigkeiten — Mit diesem Merkmal soll eine Einschätzung darüber getroffen werden, wie stark die Bereitstellung von IT-Diensten durch den Provider an seine Kunden von der Zulieferung durch externe Lieferanten abhängt. Drei Abstufungen werden hier vorgenommen: Die Abhängigkeit von Lieferanten wird als gering erachtet, wenn der Wegfall der zugelierten Dienste und Subdienste zwar zu Beeinträchtigungen im Dienstbetrieb führen kann, diese Einschränkungen aber keine kritischen Auswirkungen auf den Betrieb der Geschäftsprozesse in den Kundendomänen hätten. Von einer mittelstarken Abhängigkeit wird ausgegangen, wenn durch zugelierte Dienste wesentliche Funktionen geschäftskritischer IT-Dienste realisiert werden. Schließlich wird eine starke Lieferantenabhängigkeit angenommen, wenn das Dienstportfolio des Providers maßgeblich durch extern bezogene Dienste geprägt ist, der Provider vielfach also als Vermittler oder *Lead Provider*¹ auftritt.

→ Die Ausprägung der Lieferantenabhängigkeiten wirkt sich auf die Anforderungen ans Service-Level-Management insbesondere im Hinblick auf

¹Von einem Lead Provider wird gesprochen, wenn eine als Provider gegenüber Kunden auftretende IT-Organisation hauptsächlich Dienste von nachgeordneten Providern (*Subcontracted Provider*) zu einem Dienstportfolio bündelt und seinen Kunden anbietet bzw. vermittelt.

die Bereiche SLM-Stakeholder und Bedeutung von Lieferantenverträgen (UCs) aus. So ist die effektive Administration von UCs und Abstimmung ihrer Inhalte mit den SLAs bei Providern mit mittelmäßig oder stark ausgeprägten Lieferantenabhängigkeiten unbestritten eine Kernanforderung an ein SLM-unterstützendes Managementsystem.

7. Merkmal: CMMI-Reifegrad ITSM — CMMI (*Capability Maturity Model Integration*) ist ein anerkanntes und weit verbreitetes Reifegradmodell, das zur Beurteilung der Reife einer IT-Organisation im Hinblick auf ihre Prozessorientierung eingesetzt werden kann. Das fünf-stufige Modell kennt die Reifegrade 1 (*Initial*), 2 (*Repeatable*), 3 (*Defined*), 4 (*Managed*) und 5 (*Optimizing*).

→ Je höher die organisatorische Reife einer IT-Organisation ist, umso mehr gewinnen Anforderungen an formale Spezifikation, Dokumentation und Berichtswesen an Bedeutung, da Prozessmanagementsysteme, wie sie spätestens ab Reifegrad 3 in einer Organisation existieren, auf diese Aspekte angewiesen sind.

8. Merkmal: ISO/IEC 20000 Zertifizierung — Der internationale Standard ISO/IEC 20000 spezifiziert Mindestanforderungen an das Managementsystem eines IT-Providers. Diese Anforderungen beziehen sich vor allem auf Managementprozesse, Dokumente und Aufzeichnungen, unter anderem auch im Bereich Service-Level-Management. Provider, die eine Zertifizierung nach ISO/IEC 20000 anstreben oder schon erreicht haben, sind also auf die Erfüllung aller Norm-Anforderungen angewiesen. Insofern stellt auch dieser Aspekt einen wichtigen Einflussfaktor für die Anforderungsidentifikation dar.

→ Wird keine Zertifizierung nach ISO/IEC 20000 durch die betrachtete IT-Organisation angestrebt, so müssen auch keine Norm-Anforderungen in den Anforderungskatalog für die Managementarchitektur übernommen werden, sofern sich diese nicht ohnehin direkt aus dem zugehörigen Szenario ergeben. Anders, wenn eine Zertifizierung angestrebt wird. Bei bereits erreichter Zertifizierung sind *alle* Anforderungen der Norm auch zwingend durch ein SLM-unterstützendes Managementsystem zu realisieren.

Mit diesem morphologischen Kasten, seinen Merkmalen und Merkmalsausprägungen im Hintergrund werden nun die Szenarien, die dieser Arbeit zugrunde liegen, analysiert.

3.1.2 Szenario 1: Transportdienstleister (ITG)

Als erstes Unternehmens- und IT-Szenario soll ein internationaler Logistik-Dienstleister mit den zwei Hauptsparten Fracht und Personenbeförderung betrachtet werden. Die Internationale Transportgesellschaft (ITG) beschäftigt 2.800 Mitarbeiter, davon sind 2.200 als Fahrer im Einsatz. Die verbleibenden 600 Mitarbeiter verteilen sich auf die drei Hauptniederlassungen und 20 lokalen Verkaufsbüros in ganz Europa. Das ITG-Szenario ist angelehnt an eine Fallstudie des *Examination Institute for Information Science* (EXIN), die zu Lehrzwecken in diversen IT-Trainings eingesetzt wird. Das hat den Vorteil, dass das gesamte Szenario und insbesondere die Spezifika der IT-Organisation umfassend dokumentiert sind und diese Dokumentation leicht zugänglich ist.

3.1.2.1 Beschreibung des ITG-Szenarios

Im Folgenden werden Geschäftsmodell, IT-Dienstlandschaft sowie Status Quo und Ziele des Service-Level-Managements der ITG beschrieben.

Geschäftsmodell und Organisationsstruktur der ITG Die ITG generiert ihre Umsätze und Gewinne ausschließlich mit dem Transport von Frachtgut sowie der Beförderung von Personen. Sie ist also zugleich Spediteur und Charterbus-unternehmer. Abbildung 3.4 zeigt ein Organigramm des Unternehmens, das klassisch funktional organisiert ist und durch klare Hierarchien gekennzeichnet ist. Interessant aus Sicht des IT-Managements sind die Fachabteilungen als Kunden und somit Nutznießer aller IT-Dienstleistungen.

IT-Dienstlandschaft der ITG Die IT-Abläufe der ITG werden aus einer der Hauptniederlassungen (Utrecht) zentral verwaltet (gemanagt), die IT-Dienste jedoch nicht zentral erbracht. Sechs Informationssysteme unterstützen die Geschäftsprozesse in vielfältiger Weise. Diese Systeme sind:

- **System ADMIN** – Einsatzbereich: Finanzielle Berichterstattung
 - Anwender: Mitarbeiter der Verwaltungsabteilung
 - Bestandteile: Module für die Administration von Debitoren, Kreditoren, Gehältern, Buchhaltung
 - Unterstützte Aufgaben: Automatische Rechnungsstellung nach Auftragsabwicklung, Zahlungen an Lieferanten (von rollendem Material) und für spezialisierte Wartung

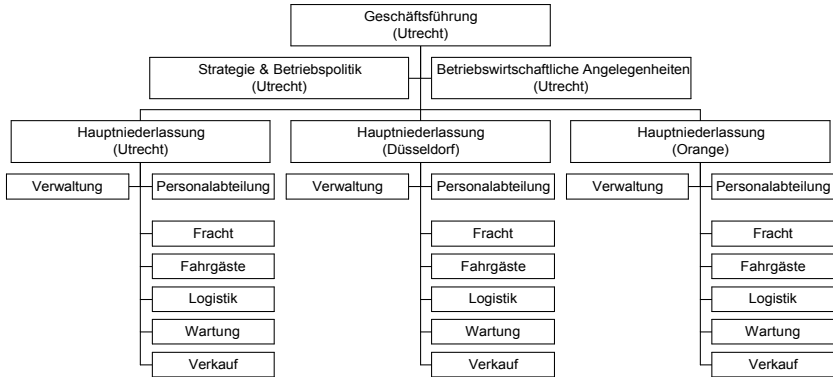


Abbildung 3.4: Organigramm der Internationalen Transportgesellschaft (ITG)
(Quelle: EXIN [3])

- Integration: gekoppelt an SHIPPING und MAINTAIN
- Sonstiges: System mit den meisten Unterschieden in den einzelnen Niederlassungen (Grund: örtliche Gesetzgebung)
- **System SHIPPING** – Einsatzbereich: Abwicklung von Frachtaufträgen
 - Anwender: Mitarbeiter in den Verkaufsabteilungen der Hauptniederlassungen sowie den örtlichen Verkaufsbüros
 - Unterstützte Aufgaben: Eingabe von Transportaufträgen, Prüfung des Status der Fracht, Erstellung erforderlicher Reisedokumente
 - Integration: gekoppelt an ADMIN und PLANNER, Datenaustausch mit ADMIN bei Auftragseingabe und Lieferung
- **System TRAVEL** – Einsatzbereich: Abwicklung von Transportaufträgen für Fahrgäste
 - Anwender: Mitarbeiter in den Verkaufsabteilungen der Hauptniederlassungen sowie den örtlichen Verkaufsbüros
 - Unterstützte Aufgaben: Eingabe von Fahrgast-Transportaufträgen, Erstellung erforderlicher Reisedokumente

- **System PLANNER** – Einsatzbereich: Logistische Planung, Koordination der Fahrer-Einsätze
 - Anwender: Mitarbeiter in den Logistikabteilungen der Hauptniederlassungen
 - Unterstützte Aufgaben: Für alle Auftragsarten wird der Einsatz von Fahrern und rollendem Material optimal geplant.
 - Integration: eng gekoppelt and SHIPPING und TRAVEL

- **System MAINTAIN** – Einsatzbereich: Wartung des Fuhrparks
 - Anwender: Mitarbeiter in den Wartungsabteilungen der Hauptniederlassungen
 - Unterstützte Aufgaben: Speicherung von Informationen über den technischen Stand des rollenden Materials wie Kilometerleistung, Inspektionstermine oder bekannte Fahrzeugmängel
 - Integration: gekoppelt an PLANNER zur Planung der Wartungsfenster

- **System OFFICE** Einsatzbereich: Klassische Büroautomatisierung
 - Anwender: Alle Mitarbeiter des Unternehmens (alle Niederlassungen und Verkaufsbüros)
 - Unterstützte Aufgaben: E-Mail, Kalender, Internet-Browser, Textverarbeitung, Datenbankanwendungen

Service-Level-Management bei der ITG Die ITG ist ein klassisches *Old Economy*-Unternehmen mit über Jahrzehnte gewachsenen Strukturen und Abläufen. Dienstmanagement im Sinne eines prozessorientierten Ansatzes wurde nie bewusst gelebt.

Status Quo: Verträge über die Dienstleistung bestehen nur zwischen der ITG-IT und ihren externen IT-Lieferanten, ausgehend von diesen Lieferanten. Dienstvereinbarungen (SLAs) zwischen der ITG-IT und ihren Kunden, den Unternehmensabteilungen an den verschiedenen Standorten, wurden noch nicht geschlossen. Grund hierfür ist unter anderem, dass das Dienstportfolio bislang nicht einheitlich dokumentiert wurde.

Ziele: Die IT-Sparte der ITG strebt an, Dienstvereinbarungen mit allen ihren Kunden abzuschließen, um die IT-Dienstleistung sowohl aus Sicht der internen IT-Organisation als auch aus Sicht der Fachabteilungen auf ein belastbares

Fundament zu stellen. Aufgrund des Geschäftsmodells der ITG ist die Zuverlässigkeit der IT-Systeme und der Unterstützung (Support) im Fehlerfall der wichtigste Aspekt, der durch SLAs abgedeckt und quantifiziert werden soll.

3.1.2.2 Morphologie des ITG-Szenarios

Abbildung 3.5 zeigt, wie sich das ITG-Szenario in den morphologischen Kästen zur Abgrenzung der Szenarien einordnet.

Anzahl/Charakteristik IT-Dienste	1		> 1, homogen		> 1, heterogen	
Anzahl IT-Kunden-Standorte	1			mehrere		
Kundenumfeld	nur interne Kunden		gemischt		nur externe Kunden	
IT-Leistungsverrechnung	nicht geplant		in Planung		vorhanden	
Anzahl IT-Provider-Standorte	1			mehrere		
Lieferantenabhängigkeiten	gering		mittel		stark	
CMMI-Reifegrad ITSM	1	2	3	4	5	
ISO/IEC 20000 Zertifizierung	nicht angestrebt		angestrebt		erreicht	

Abbildung 3.5: Morphologie des ITG-Szenarios

Als IT-Dienste werden hauptsächlich die sechs genannten Informationssysteme angesehen, die vom IT-Provider betrieben werden. Kunden sind die Fachabteilungen an den drei Hauptniederlassungen sowie die 20 lokalen Verkaufsbüros. Damit ist das Dienstspektrum zwar differenziert, die Anforderungen verschiedener Kunden an die IT-Dienste sind allerdings weitgehend homogen. Insbesondere bestehen keine grundlegend unterschiedlichen Ansprüche an die Leistungsmerkmale ein- und desselben Systems. IT-Dienstleistungen werden grundsätzlich nicht verrechnet. Die Systeme werden nicht in einem zentralen Rechenzentrum erbracht und administriert, sondern es wird je ein voll ausgestattetes Rechenzentrum an jedem Standort einer Hauptniederlassung betrieben. Somit bestehen insgesamt drei Standorte für den IT-Provider. Die ITG-IT bezieht Telekommunikationsdienste, insbesondere zur Vernetzung der Hauptniederlassungen untereinander sowie der Hauptniederlassungen und der Verkaufsbüros (*Wide Area Network*, WAN) von externen Telekommunikationsdienstleistern. Außerdem werden zur Realisierung der Dienste ADMIN,

PLANNER und OFFICE Produkte von externen Softwareherstellern verwendet, für die entsprechende Lizenzen eingekauft wurden. Der Wegfall dieser zugelieferten Dienste hätte zwar erhebliche Auswirkungen auf den IT-Betrieb, allerdings handelt es sich hauptsächlich um Standard-Dienste, die mit mäßigem Aufwand substituierbar sind. Daher werden die Lieferantenabhängigkeiten noch als mittelstark ausgeprägt bewertet. Der CMMI-Reifegrad ist mit 2 gering, eine Zertifizierung nach ISO/IEC 20000 auch langfristig kein Thema.

3.1.3 Szenario 2: Hochschulrechenzentrum (LRZ)

Als zweites Szenario wird das Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften² betrachtet, welches den Universitäten und Fachhochschulen im Großraum München in seiner Funktion als Hochschulrechenzentrum vielfältige IT-Dienste bereitstellt. Das LRZ agiert darüber hinaus als nationales Höchstleistungsrechenzentrum und Betreiber des Münchner Wissenschaftsnetzes (MWN) als Zugang zum Deutschen Forschungsnetz (DFN). Das LRZ bietet sich vor allem deswegen als ein Szenario für den Einsatz eines IT-gestützten Managementsystems für SLM an, weil die Vielfalt von Diensten und Kunden ausgesprochen groß ist. Zudem unterscheidet sich das LRZ in vielen Punkten maßgeblich vom ITG-Szenario (vgl. Morphologie des Szenarios).

3.1.3.1 Beschreibung des LRZ-Szenarios

Es werden wie schon für das ITG-Szenario wieder Geschäftsmodell, IT-Dienstlandschaft sowie Status Quo und Ziele des Service-Level-Managements beschrieben.

Geschäftsmodell des LRZ Das LRZ operiert auf zwei Hauptgeschäftsfeldern: Zum einen ist es als Hochschulrechenzentrum regionaler IT-Dienstleister für Hochschulen und Universitäten, zum anderen auf nationaler und internationaler Ebene Betreiber eines Höchstleistungsrechners, der zur Nutzung durch Einrichtungen aus Wissenschaft und Forschung zur Verfügung steht.

IT-Dienstlandschaft des LRZ Die durch das LRZ für seine Kunden bereitgestellten Dienste lassen sich wie folgt klassifizieren:

²<http://www.lrz.de>

- **Kommunikationsdienste** – Netzbasierte Dienste
 - Anwender: Mitarbeiter und Studenten der angeschlossenen Universitäten
 - Dienste: Elektronische Post (E-Mail), Drahtlosnetze (WLAN), Videokonferenzen, Virtuelle private Netze (VPN), u.a.

- **Datensicherung** – Dienste zur Sicherstellung der Datenverfügbarkeit
 - Anwender: Mitarbeiter der Lehrstühle angeschlossener Universitäten sowie der Verwaltungsabteilungen
 - Dienste: Sicherung (Backup), Wiederherstellung (Recovery) und Datenarchivierung

- **Server-Dienste** – Server-basierte Dienste
 - Anwender: Mitarbeiter der Lehrstühle angeschlossener Universitäten sowie der Verwaltungsabteilungen
 - Dienste: Fileserver, Webhosting, u.a.

- **Special Facilities** – Besonders ausgestattete Computersysteme oder Räumlichkeiten
 - Anwender: Studenten sowie Mitarbeiter der Lehrstühle angeschlossener Universitäten
 - Dienste: Multimedia-Arbeitsplätze, Großformat-Drucker und -Scanner, Virtual-Reality-Labor, u.a.

- **Hochleistungsrechnen** – Komplexe (zumeist wissenschaftliche) Berechnungen
 - Anwender: Mitarbeiter deutscher Forschungsinstitute
 - Dienst: Bereitstellung von Rechenzeit auf dem nationalen Hochleistungsrechner (Supercomputer) SGI Altix 4700

Service-Level-Management am LRZ Nicht nur in seiner Eigenschaft als Hochschulrechenzentrum, sondern auch als Forschungsstandort und Höchstleistungsrechenzentrum steht das LRZ in Konkurrenz zu regionalen und nationalen Anbietern von IT- und Telekommunikationsdiensten sowie europa- und weltweit zu anderen Hochleistungsrechenzentren.

Status Quo: Verträge über die Dienstleistung bestehen nur mit externen Zulieferern und Partner-Rechenzentren.

Ziele: Eine konsistente Verwaltung von Dienstinformationen und -katalogen wird vom LRZ mit hoher Priorität angestrebt. Darauf aufbauend sollen mittelfristig Dienstvereinbarungen mit allen Kunden abgeschlossen werden.

3.1.3.2 Morphologie des LRZ-Szenarios

Auch für das LRZ-Szenario wird wieder eine Einordnung in den morphologischen Kasten vorgenommen. Das Ergebnis ist in Abbildung 3.6 dargestellt.

Anzahl/Charakteristik IT-Dienste	1		> 1, homogen		> 1, heterogen	
Anzahl IT-Kunden-Standorte	1			mehrere		
Kundenumfeld	nur interne Kunden		gemischt		nur externe Kunden	
IT-Leistungsverrechnung	nicht geplant		in Planung		vorhanden	
Anzahl IT-Provider-Standorte	1			mehrere		
Lieferantenabhängigkeiten	gering		mittel		stark	
CMMI-Reifegrad ITSM	1	2	3	4	5	
ISO/IEC 20000 Zertifizierung	nicht angestrebt		angestrebt		erreicht	

Abbildung 3.6: Morphologie des LRZ-Szenarios

Im LRZ-Szenario ist nicht nur die Anzahl der Dienste sehr hoch, sondern auch die Kundenanforderungen fallen teils sehr unterschiedlich aus. Beispielsweise sind universitäre Lehr- und Forschungseinheiten technischer oder informationstechnischer Ausrichtung deutlicher stärker auf bestimmte IT-Dienste und Spezialeinrichtungen in hoher Verfügbarkeit und Leistungsfähigkeit angewiesen als Lehrstühle mit nicht-technischer Ausrichtung oder universitäre Verwaltungseinrichtungen, die wieder andere Anforderungen an die IT-Systeme

stellen. Die Zahl der geographischen Kundenstandorte ist hoch; hinzu kommt, dass die Entfernungen zwischen verschiedenen Kundenstandorten zum Teil sehr groß sind. Da das LRZ als Einrichtung der Bayerischen Akademie der Wissenschaften eine eigenständige Organisation darstellt, müssen die IT-Kunden als externe Kunden aufgefasst werden, auch wenn die Kundenbeziehung nicht der einer klassischen Geschäftsbeziehung entspricht. Alle IT-Dienste werden an einem Standort zentral gesteuert und unterstützt, Außenstellen oder verteilte Rechenzentren gibt es nicht. Kritische Lieferantenabhängigkeiten gibt es verhältnismäßig wenige.

Der CMMI-Reifegrad kann mit 3 angegeben werden, was deutlich macht, dass Prozessorientierung im IT-Management am LRZ eine zunehmend bedeutendere Rolle einnimmt. Eine Zertifizierung nach ISO/IEC 20000 wird aus verschiedenen Gründen gezielt angestrebt, darunter Qualitätsgesichtspunkte, aber auch die Erzielung von Wettbewerbsvorteilen zur besseren Positionierung in nationalen und internationalen Ausschreibungen.

3.1.4 Szenario 3: Flughafen (FMG)

Der Flughafen München Franz Josef Strauß gehört zu den beiden größten Verkehrsflughäfen in Deutschland. Auf einer Fläche von 1.600 Hektar befinden sich zwei Terminals sowie zwei Start- und Landebahnen von jeweils 4 Kilometern Länge. Auf insgesamt 27.400 Beschäftigte kommen gut 34 Millionen Passagiere und 430.000 Flugbewegungen pro Jahr. Neben den Passagierflügen werden jährlich etwa 250.000 Tonnen Luftfracht abgefertigt.

3.1.4.1 Beschreibung des FMG-Szenarios

Auch für das FMG-Szenario werden wieder Geschäftsmodell und Organisationsstruktur, IT-Dienstlandschaft sowie Status Quo und Ziele des Service-Level-Managements skizziert.³

Geschäftsmodell und Organisationsstruktur der FMG Der Flughafen München wird von einer GmbH – der Flughafen München GmbH, kurz FMG – geführt. Diese GmbH hat zum Ziel, den reibungslosen täglichen Flugbetrieb zu ermöglichen. Kunden der FMG sind hauptsächlich Fluggesellschaften und

³Die Beschreibung des FMG-Szenarios wurde auf Basis von öffentlich zugänglichen Informationen und Interviews erstellt. Einige Zusammenhänge wurden in der Darstellung vereinfacht oder abgeändert, sofern die zur Anforderungsgewinnung maßgebliche Charakteristik des Szenarios dadurch nicht beeinträchtigt wurde.

andere am Flughafen operierende Unternehmen, sowie indirekt auch die Fluggäste selbst. Die beiden Haupt-Geschäftsfelder sind *Ground Operations* und *Aviation*.

Zu *Ground Operations* gehören alle Aktivitäten, die in direktem Zusammenhang mit der Passagier- und Gepäckabfertigung stehen. Beispiele für Geschäftsprozesse in diesem Bereich sind Gepäck-Erfassung, Weiterleitung und Verladung, Pflege und Bereitstellung von Fluginformationen, Busverkehr auf dem Flughafengelände und viele andere mehr.

In den Bereich *Aviation* fallen alle Aktivitäten im Zusammenhang mit der Flugvor- und -nachbereitung. Beispiele für Geschäftsprozesse sind hier Gate- und Startbahnbereitstellung, Gangway-Bereitstellung, Flugzeug-Betankung, Flugzeug-Reinigung und weitere.

Der Flughafen München wird als gewinnorientiertes Unternehmen geführt. Haupt-Einnahmequellen sind geschäftliche Aktivitäten wie die zuvor genannten, die gegenüber den Fluggesellschaften und anderen Drittunternehmen kostenpflichtig erbracht werden. Das genaue Geschäftsmodell ist außerordentlich komplex, da es vielfältige Geschäftsprozesse mit zahlreichen Kunden-, Partner- und Lieferantenbeziehungen beinhaltet. Zudem sind die gesetzlichen Rahmenbedingungen sowie nationale und internationale Regelungen zum Flugverkehr zu berücksichtigen. Auf all diese Aspekte muss aber an dieser Stelle nicht näher eingegangen werden.

Die FMG bietet sich vor allem deswegen als ein Szenario für den Einsatz eines IT-gestützten Managementsystems für SLM an, weil sich die Vielfalt der IT-Dienste sowie die Anforderungen an deren Qualität in Teilen von den anderen beiden Szenarien unterscheiden.

IT-Dienstlandschaft des FMG Die durch die FMG für ihre Kunden bereitgestellten Dienste lassen sich wie folgt klassifizieren:

- **Kommunikationsdienste** – Netzbasierte Dienste
 - Anwender: Mitarbeiter der FMG, der Fluggesellschaften sowie anderer angesiedelter Firmen und Geschäfte
 - Dienste: Computer-integrierte VoIP⁴-Telefonie, Elektronische Post (E-Mail), LAN und Internetzugang, Funk-LAN, u.a.
- **Applikationsdienste** – Standard- und Spezialanwendungen

⁴VoIP: Voice over Internet Protocol

- Anwender: Mitarbeiter der FMG und der Fluggesellschaften
- Dienste: Klassische Office-Anwendungen, SAP R/3-Module, Spezial-Applikationen für Fahrzeug-Einsatzplanung, Rollfeld- und Gate-Belegungsplanung, u.a.
- **Informationsdienste** – Datenbank-basierte Dienste mit Fokus auf Verwaltung und Bereitstellung strukturierter Informationen
 - Anwender: Mitarbeiter der FMG und der Fluggesellschaften
 - Dienste: Fluginformationssystem (Verwaltung und Pflege aktueller Ankunfts- und Abfluginformationen), Gate-Informationssystem (Verwaltung und Pflege Gate-spezifischer Abfluginformationen), Gepäckinformationssystem (Tagging, Identifikation, Routing und Ortung aufgebener Gepäckstücke)
- **Hardwarenahe Dienste** – Dienste im Zusammenhang mit der Bereitstellung, Wartung und Unterstützung technischer Geräte
 - Anwender: Mitarbeiter der FMG und der Fluggesellschaften
 - Dienste: Bereitstellung und Konfiguration von Standard-Arbeitsplätzen/Benutzerterminals, Speicherplatzbereitstellung (Network Storage), Datensicherung (zu Archivierungs- und Wiederherstellungszwecken)

Service-Level-Management bei der FMG Von allen drei Szenarien sind die grundlegenden Ideen und Konzepte des Service-Level-Managements im FMG-Szenario am umfassendsten und konsequentesten implementiert.

Status Quo: Service-Level-Management wird gemäß der Mindestanforderungen des Standards ISO/IEC 20000 durchgeführt. Das bedeutet vor allem, es gibt einen für alle IT-Kunden zugänglichen Dienstkatalog sowie Verträge mit allen externen Zulieferern der IT. Zudem bestehen SLAs zwischen dem IT-Provider der FMG und internen wie externen Kunden, wobei für gewisse Standarddienste allgemeingültige Nutzungsbedingungen festgelegt wurden, wodurch die Aushandlung individueller Vereinbarungen für diese Dienste nur noch als zusätzliche Option gesehen wird.

Ziele: Wünschenswert aus Sicht der FMG wäre die Integration der Informationen und Daten im Zusammenhang mit dem Service-Level-Management, um redundante Datenhaltung und damit die Gefahr von Inkonsistenzen und Fehlern zu verringern und effektiver und effizienter im Management von Diensten und Dienstvereinbarungen zu werden.

3.1.4.2 Morphologie des FMG-Szenarios

Abbildung 3.6 zeigt das Ergebnis der morphologischen Analyse des FMG-Szenarios.

Anzahl/Charakteristik IT-Dienste	1		> 1, homogen		> 1, heterogen	
Anzahl IT-Kunden-Standorte	1			mehrere		
Kundenumfeld	nur interne Kunden		gemischt		nur externe Kunden	
IT-Leistungsverrechnung	nicht geplant		in Planung		vorhanden	
Anzahl IT-Provider-Standorte	1			mehrere		
Lieferantenabhängigkeiten	gering		mittel		stark	
CMMI-Reifegrad ITSM	1	2	3	4	5	
ISO/IEC 20000 Zertifizierung	nicht angestrebt		angestrebt		erreicht	

Abbildung 3.7: Einordnung des FMG-Szenarios in den Morphologischen Kästen

Wie auch bereits im LRZ-Szenario ist die Anzahl der IT-Dienste vergleichsweise hoch, während die Kundenanforderungen an diese Dienste sehr heterogen sind. So können beispielsweise Verfügbarkeitsanforderungen an ein- und denselben Informationsdienst je nach Kunde stark divergieren – in Abhängigkeit von der Kritikalität des Dienstes im Hinblick auf die Unterstützung oder Ermöglichung geschäftlicher Aktivitäten. Im Unterschied zu den ITG- und LRZ-Szenarios werden IT-Dienste nur an einem (großen) Standort erbracht, wobei als Kunden sowohl Abteilungen und Teams der FMG als auch andere Unternehmen (zum Beispiel Fluggesellschaften oder am Flughafen angesiedelte Gastronomie- und Einzelhandelsbetriebe) auftreten. Alle IT-Dienstleistungen werden verrechnet – auch gegenüber den FMG-internen Kunden.

Die IT-Organisation der FMG operiert von einem zentralen Standort auf dem Flughafengelände aus. Lieferanten nehmen für die FMG-IT eine bedeutende Rolle ein. Softwaretechnik, aber auch Hardwaresysteme und Anlagen sowie die Weitverkehrsnetzanbindung des Flughafens und zahlreiche Subdienste werden von externen Providern, Herstellern und Systemhäusern bezogen. Der CMMI-Reifegrad kann mit 4 angegeben werden, was sich nicht zuletzt in einer erfolgreichen Zertifizierung nach ISO/IEC 20000 widerspiegelt.

3.1.5 Vollständigkeit der Szenarien

Es sei an dieser Stelle noch einmal darauf hingewiesen, dass die Szenarien im Rahmen des Entwicklungsrahmens der Arbeit instrumentiert werden, um funktionale Anforderungen an das Service-Level-Management und darauf aufbauend Anwendungsfälle für ein IT-gestütztes Managementsystem im Bereich Service-Level-Management zu bestimmen. Vor diesem Hintergrund stellt sich natürlich die Frage, warum gerade diese Szenarien als Basis für die Anforderungsermittlung gewählt wurden und ob sie in der Summe überhaupt ausreichen, um eine Grundlage für einen *allgemein relevanten und vollständigen* Katalog an funktionalen Anforderungen an eine *generische* Managementarchitektur zu bilden.

Kurz: *Ist es glaubwürdig, dass die vorgestellten Szenarien ein umfassendes „Erzeugendensystem“ für die funktionalen Architekturansprüche darstellen?*

Diese Frage zu adressieren ist das Ziel dieses Abschnitts. Einleitend sei darauf hingewiesen, dass ein Nachweis im Sinne eines mathematischen Vollständigkeitsbeweises nicht erbracht werden kann. Dazu müsste man Kenntnis über den gesamten Raum aller möglichen Szenarien haben und diese auch berücksichtigen, was unmöglich ist. Man ist also gezwungen, in diesem Punkt einen Schritt zurückzutreten und die Frage der Vollständigkeit im Hinblick auf die folgenden beiden Ziele zu betrachten:

1. Es soll dargestellt werden, dass die vorgestellten Szenarien in der Summe hinreichend viele Aspekte und Anforderungen an die Disziplin Service-Level-Management im Allgemeinen sowie an ein IT-gestütztes Managementsystem für Service-Level-Management mit sich bringen, sodass eine Managementarchitektur, die sich auf diese Anforderungen stützt, auch für andere, hier nicht konkret erfasste Szenarien einen anwendbaren Lösungsansatz darstellen kann.
2. Wohlwissend, dass auf diese Weise keine absolute generische Vollständigkeit erreicht werden kann und sich bei Anwendung der Architektur zur Entwicklung eines Managementsystems vor dem Hintergrund eines spezifischen Szenarios weitere Anwendungsfälle und Anforderungen ergeben können, darf weder im Rahmen der Anforderungsanalyse, noch in späteren Kapiteln eine explizite oder implizite Annahme über die Abgeschlossenheit der Menge der Anforderungen getroffen werden. Diese Forderung beeinflusst den Architekturentwurf insofern, dass die (nicht-funktionale) Anforderung der Erweiterbarkeit der Architektur und aller enthaltenen Teilmodelle stets berücksichtigt werden muss. Im Detail wird

dieser Aspekt im Abschnitt 3.4 (Nicht-funktionale Anforderungen) und im Kapitel 7 (Implementierungsaspekte) adressiert.

Zusammengefasst: Es soll gezeigt werden, dass die eingeführten Szenarien in der Summe hinreichend vollständig für eine zielführende Anforderungsanalyse sind. Gleichzeitig wird aber kein Anspruch auf absolute Vollständigkeit erhoben.

Um dieses Ziel zu erreichen, wird Bezug auf die Morphologien der vorgestellten Szenarien genommen. Legt man die morphologischen Kästen aus den Analysen der einzelnen Szenarien übereinander, so erhält man den in Abbildung 3.8 dargestellten morphologischen Kasten. Hier wird deutlich, dass alle Szenarien in der Summe 21 von 24 Merkmalsausprägungen abdecken.

Anzahl/Charakteristik IT-Dienste	1		> 1, homogen		> 1, heterogen	
Anzahl IT-Kunden-Standorte	1			mehrere		
Kundenumfeld	nur interne Kunden		gemischt		nur externe Kunden	
IT-Leistungsverrechnung	nicht geplant		in Planung		vorhanden	
Anzahl IT-Provider-Standorte	1			mehrere		
Lieferantenabhängigkeiten	gering		mittel		stark	
CMMI-Reifegrad ITSM	1	2	3	4	5	
ISO/IEC 20000 Zertifizierung	nicht angestrebt		angestrebt		erreicht	

Abbildung 3.8: Zusammenfassende morphologische Analyse aller Szenarien

Vor dem Hintergrund dieses Ergebnisses decken die vorgestellten Szenarien also einen großen Teil der möglichen Ausprägungen ab. Diese Beurteilung steht und fällt natürlich mit dem Aufbau des morphologischen Kastens, seiner Merkmale und der vorgesehenen Merkmalsausprägungen. Immerhin konnte die komplexe und ohne solche Hilfsmittel nicht beantwortbare Frage nach der Vollständigkeit der Szenarien zurückgeführt werden auf die Frage nach der Plausibilität des morphologischen Kastens.

Die drei nicht adressierten Merkmalsausprägungen stellen keine nennenswerte oder gar kritische Hürde für den Entwurf einer generisch-vollständigen Architektur dar. Für *Single Service Provider* (Merkmal: Anzahl/Charakteristik

IT-Dienste, Ausprägung: 1) beschränken sich die Anforderungen ans Service-Level-Management und somit an ein unterstützendes automatisiertes Managementsystem im Vergleich zu Anbietern mehrerer Dienste auf eine reduzierte Anforderungsteilmenge, da der Aspekt der Differenzierung zwischen verschiedenen Diensten und ihren spezifischen technisch-funktionalen Eigenschaften wegfällt. Service-Level-Management kann als Disziplin auch für solche Service-Provider von Bedeutung sein, ist dann allerdings auf die Kernaspekte wie die Definition der Service-Level-Ziele in SLAs, deren Messung und ein entsprechendes Berichtswesen (*Reporting*) fokussiert. Diese Anforderungen werden aber durch die drei verwendeten Szenarien ohnehin adressiert, wie in Kürze gezeigt wird.

Provider geringster und höchster Prozessreife im IT-Service-Management (Merkmal: CMMI-Reifegrad ITSM, Ausprägung: 1 bzw. 5) sind in der Menge der zugrunde gelegten Szenarien nicht enthalten. Für den ersten Fall ist dies leicht zu vertreten: IT-Provider mit dem geringsten Reifegrad 1 haben mit Sicherheit keine Bestrebungen in Richtung Service-Level-Management, geschweige denn einer Unterstützung der damit verbundenen Prozesse durch ein IT-gestütztes Managementsystem. Hat ein Provider hingegen den höchstmöglichen Reifegrad erreicht, so ist davon auszugehen, dass Service-Level-Management bereits „in Perfektion“ betrieben wird. Eigentlich wäre ein solcher Provider – wenn es ihn denn gibt – ein interessantes Szenario für die intendierte Managementarchitektur. Allerdings würde auch die Gefahr bestehen, dass im direkten Vergleich zu einem Provider mit Reifegrad 4 vor allem sehr spezifische und überwiegend als optional zu bewertende Anforderungen als neue und zusätzliche Aspekte abfallen. Angesichts der Komplexität, die mit dem Entwurf der Architektur ohnehin einhergeht, ist eine Vernachlässigung dieser potenziellen Anforderungsquelle vertretbar.

Damit wurde abschließend auch argumentiert, warum die vorgestellten Szenarien ausreichen, um eine vollständige Menge funktionaler Anforderungen zu generieren – auch wenn sie den morphologischen Kasten in der Summe nicht zu 100 Prozent ausfüllen können.

3.2 Funktionale Anforderungen

Nun, da die Szenarien vorgestellt wurden und ihre Vollständigkeit diskutiert wurde, sollen Anwendungsfälle für ein Managementsystem im Service-Level-Management identifiziert werden, die die Grundlage für das Design der Managementarchitektur bilden.

Gemäß der in Abbildung 2.6 (Kapitel 2) dargestellten Systematik erfolgt die Ableitung von Anwendungsfällen für ein IT-gestütztes Managementsystem aus den Szenarien über einen Zwischenschritt, nämlich die Identifikation allgemeiner Anforderungen ans Service-Level-Management – kurz: die Darstellung des *allgemeinen Funktionsspektrums*.

Bemerkung: An dieser Stelle sei noch einmal auf die Unterscheidung zwischen *Managementsystem* und *Managementarchitektur* hingewiesen, wie sie in Kapitel 2 vorgenommen wurde. Primär werden in diesem Abschnitt die funktionalen Eigenschaften eines Managementsystems beschrieben. Gleichzeitig liefern diese aber die Zielvorgaben für die *Managementarchitektur*, die ja die Erstellung eines solchen Systems erleichtern soll.

Das Funktionsspektrum soll die Aufgabenbereiche, die mit der Disziplin Service-Level-Management einhergehen und die bisher nur ausschnittsartig angesprochen wurden, konkret ausformuliert und strukturiert darstellen, ohne dabei schon auf einzelne Anwendungsfälle eines Managementsystems einzugehen. Allerdings wird das Funktionsspektrum im Anschluss natürlich auf die Anwendungsfälle abgebildet. Somit stellt das Funktionsspektrum das „Bindeglied“ zwischen Szenarien und Managementsystem-Anwendungsfällen dar.

Bemerkung: Das allgemeine Funktionsspektrum soll Anforderungen und Aufgabenbereiche eines (im Sinne der zugrunde gelegten Szenarien) idealen und vollständigen Service-Level-Managements beschreiben. Automatisierungsaspekte und Spezifika eines IT-gestützten Managementsystems spielen hier zunächst noch keine Rolle.

Wichtig ist, dass das Funktionsspektrum alle SLM-relevanten Anforderungen in den Szenarien identifiziert und sich dabei nicht auf eine oder mehrere bestimmte Phasen des Dienstlebenszyklus beschränkt, sondern diesen vollständig umfasst. Folglich sind die relevanten Eingaben (*Inputs*) zur Ermittlung des Funktionsspektrums:

1. Szenariobeschreibungen und zugehörige morphologische Kästen — Welches sind reale Situationen und Randbedingungen, aus denen sich Funktionsanforderungen für SLM ergeben?
2. Zieldefinition für Service-Level-Management — Was wird überhaupt als Zielsetzung von Service-Level-Management erachtet? Diese Zielsetzung

ist der „Treiber“ für die Identifikation der Bestandteile des Funktionsspektrums. Die Szenarien bieten ein breites Spektrum verschiedenartiger Anforderungen ans Dienstmanagement. Um den Fokus zielführend auf SLM-Anforderungen und -Aspekte zu legen, muss eine Definition der Zielsetzung des Service-Level-Managements gegeben werden, von der im Rahmen dieser Arbeit ausgegangen wird.

3. Dienstlebenszyklus — Welche Lebenszyklusphasen eines Dienstes müssen dabei berücksichtigt und abgedeckt werden?

Der erste Punkt, die Szenariobeschreibungen, wurde im vorangegangenen Abschnitt bereits adressiert. Es fehlen also noch eine SLM-bezogene Zieldefinition sowie ein Modell eines generischen Dienstlebenszyklus. In beiden Punkten wird im Folgenden auf bestehende Literatur zurückgegriffen.

3.2.1 SLM-Zielsetzung und Grundannahmen

Zunächst wird eine Zieldefinition für die Aufgaben im Service-Level-Management angegeben. Danach werden zur Präzisierung einige Grundannahmen getroffen.

3.2.1.1 Definition der Zielsetzung von Service-Level-Management

Die *IT Infrastructure Library* (ITIL) formuliert die Ziele, die mit Service-Level-Management verfolgt werden sollen, wie folgt (sinngemäße Übersetzung):

Zielsetzung von Service-Level-Management Service-Level-Management hat das Ziel sicherzustellen, dass alle IT-Dienste in der vereinbarten Qualität geliefert werden und vereinbarte Qualitätsziele erreichbar sind. Zu diesem Zweck müssen alle in Betrieb befindlichen Dienste überwacht und ihre Leistungsparameter gemessen werden. Hierüber müssen aussagekräftige Berichte erstellt werden.

Ganz ähnliche, in der Tendenz vergleichbare Definitionen liefern auch ISO/IEC 20000, COBIT und eTOM (vgl. Kapitel 4). Und so deckt sich diese Definition auch mit dem SLM-Begriff, der dieser Arbeit zugrunde liegt und in Kapitel 1 dargestellt wurde. Ein SLM-unterstützendes Managementsystem muss nun Funktionalitäten zur Verfügung stellen, die die Erreichung dieser Ziele ermöglichen.

Auf die Differenzierung zwischen dem Nutzwert (*utility*) und dem Leistungswert (*warranty*) eines Dienstes, wie sie bereits in Abschnitt 2.3.1 vorgenommen wurde, sei an dieser Stelle nochmals hingewiesen. So bedeutet Dienstqualität nicht nur die Bereitstellung eines Dienstes mit der vereinbarten Performanz (zum Beispiel Verfügbarkeit, Zuverlässigkeit, Zugriffszeiten, etc.), sondern auch in der vereinbarten Funktionalität. Dieser funktionale Aspekt wird in vielen SLM-Ansätzen vernachlässigt.

3.2.1.2 Grundannahmen zum Service-Level-Management

Da die Zielsetzung allein noch zu viele Freiheits- und Interpretationsgrade offen lässt, um auf ihrer alleinigen Basis eine fundierte Anforderungsanalyse durchzuführen, führt kein Weg daran vorbei, einige zusätzliche Annahmen zu formulieren, die etwas darüber aussagen, *wie* die SLM-Ziele erreicht werden können. Diese nun folgenden Annahmen greifen einige der in Kapitel 2 eingeführten Begriffe und Konzepte auf und präzisieren die im vorherigen Teilabschnitt angegebene allgemeine SLM-Zielsetzung durch die Postulierung einiger genereller Zusammenhänge und Abhängigkeiten.

1. Annahme: Jede IT-Dienstleistungserbringung zwischen dem IT-Provider und einem seiner Kunden soll durch eine gültige Dienstvereinbarung (SLA) untermauert werden.
→ Diese Annahme deckt sich mit dem generischen Dienstmodell (vgl. Abschnitt 2.3.1).
2. Annahme: Jede gültige Dienstvereinbarung soll mit Hilfe geeigneter interner operativer Vereinbarungen (OLAs) und Verträge mit externen Lieferanten/Sub-Providern (UCs) auf ein belastbares Fundament gestellt werden.
→ Ein Verzicht auf diese Grundannahme hätte zur Folge, dass SLAs quasi „in der Luft“ hängen können, losgelöst von jeder Erfüllungsverbindlichkeit von an der Dienstleistung beteiligten Parteien.
3. Annahme: Es soll für jeden Kunden (mindestens) einen gültigen Dienst-katalog geben, der alle Dienste des Gesamtportfolios enthält, die von diesem Kunden beauftragt werden können und der aus den mit diesem Kunden abgeschlossenen Dienstvereinbarungen referenziert wird.
→ Diese Annahme soll den Weg zu einer konsistenten Administration SLM-relevanter Dienstinformationen ebnen.

Somit bilden SLAs, OLAs und UCs sowie Dienstkataloge und das Dienstportfolio die fünf zentralen Informationsartefakte im Service-Level-Management, von denen im Folgenden ausgegangen wird. Durch die Ableitung von Anforderungen und die Überführung in konkrete Managementsystem-Anwendungsfälle werden sich weitere ergeben, die zu diesen fünf Objekten in Beziehung gesetzt werden müssen. Detailspekte hierzu werden im Rahmen des Informationsmodells zunächst aus Prozess- und später aus Systemsicht adressiert.

3.2.2 Definition eines generischen Dienstlebenszyklus

In einigen Dienstmanagement-Ansätzen wird die Idee verfolgt, die anfallenden Aufgaben und Aktivitäten in den Kontext eines Dienstlebenszyklus zu setzen. Auch ITIL verfolgt in seiner aktuellen Version 3 [19] diesen Ansatz und unterscheidet die Kern-Phasen Dienstentwurf (*Service Design*), Dienstbereitstellung (*Service Transition*) und Dienstbetrieb (*Service Operation*).

Ein etwas differenzierteres, in seiner Grundaussprache aber vergleichbares Modell für einen Dienstlebenszyklus liefert das Dienstmanagement-Rahmenwerk von Dreo [4]. Abbildung 3.9 illustriert dieses Modell. Nachfolgend einige Erläuterungen dazu, gefolgt von einer Erklärung, inwiefern dieses Modell für die weiteren Schritte der Anforderungsanalyse instrumentiert wird.

3.2.2.1 Die Dienstlebenszyklusphasen

Die Planungsphase eines Dienstes umfasst nicht nur technische, sondern auch administrative Aspekte der Dienstbereitstellung und des Dienstbetriebs. Ziel dieser Phase muss es sein, einen Dienst zu planen, der geschäftliche und Kundenanforderungen erfüllt, technisch effektiv und effizient realisierbar ist und über seinen gesamten Lebenszyklus, also insbesondere in den der Planung nachgelagerten Lebenszyklusphasen, langfristig „managebar“ ist und bleibt. Nach Hegering et al. [1] erfordert die Phase der Dienstplanung zu diesem Zweck die Durchführung zahlreicher Analysen wie Anwendungs-, Bedarfsschwerpunkt-, Bedarfsgrößen- und Komponentenanalysen. Aus Sicht des Service-Level-Managements ist die Planungsphase vor allem deshalb von Interesse, weil hier die Grundsteine für spätere SLM-Kernaktivitäten wie die Verhandlung der Dienstfunktionalität und des Service-Levels im Rahmen von Dienstvereinbarungen gelegt werden.

In der Verhandlungsphase wird eine auf der vorherigen Planung basierende Dienstvereinbarung zwischen Provider und Kunde ausgehandelt. Im Unterschied zur Planungsphase wird nun eine ganz konkrete Dienstinstanz betrach-

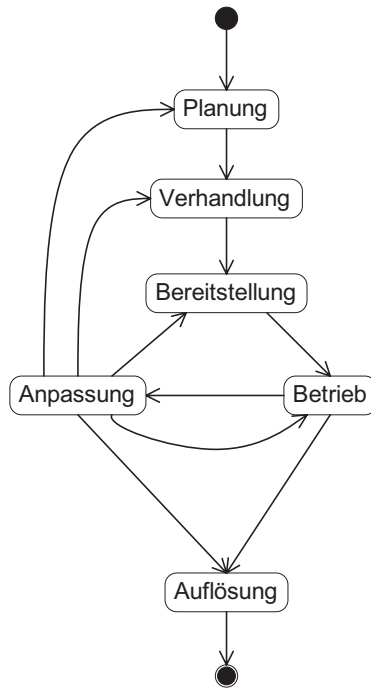


Abbildung 3.9: Typischer Dienstlebenszyklus nach Dreo [4]

tet. Die finale Beauftragung des Dienstes durch den Kunden bildet den Abschluss der Verhandlungsphase.

In der Bereitstellungsphase wird die im SLA vereinbarte Dienstfunktionalität durch den Provider realisiert. In dieser Phase werden Ressourcen in der Art und Weise bereitgestellt, installiert, konfiguriert und getestet, dass der Dienst vereinbarungsgemäß betrieben werden kann.

Nach Abnahme des Dienstes durch den Kunden (sofern erforderlich) wird dieser in Betrieb genommen. Zusätzlich zum eigentlichen Dienstbetrieb werden vom Provider während dieser Lebenszyklusphase Managementaufgaben insbesondere im Bereich der Dienstunterstützung (Support) übernommen.

Die Anpassungsphase umfasst schließlich alle Aktivitäten, die mit Änderungen an der Dienstfunktionalität bzw. der Realisierung dieser Funktionalität

einhergehen. In Abhängigkeit von der Ausprägung der im Rahmen dieser Phase vorgenommenen Änderungen kann eine Rückkopplung zur Bereitstellungs-, Verhandlungs- oder gar zur Planungsphase sinnvoll oder notwendig sein. Im einfachsten Fall aber erfordert die Dienstanpassung keine neuen oder zusätzlichen Bereitstellungsaktivitäten und geht somit wieder direkt in die Betriebsphase über.

Den Abschluss des Dienstlebenszyklus bildet die Auflösungsphase, in der der Dienst formal stillgelegt, alle durch die Implementierung belegten Ressourcen wieder freigegeben und die für das Management des Dienstes eingerichteten Prozesse und Verfahren eingestellt werden, sofern sie nicht für den Betrieb anderer Dienste benötigt werden.

3.2.2.2 Der Dienstlebenszyklus als Hilfsmittel in der Anforderungsanalyse

Service-Level-Management umfasst verschiedenste Aufgabenbereiche, die unterschiedliche Dienstlebenszyklusphasen tangieren. Wenn die Anforderungen an Service-Level-Management nachfolgend auf Grundlage der Szenarien und mit Hilfe der SLM-Zieldefinition identifiziert werden, wird zusätzlich jeweils festgelegt, welche Lebenszyklusphasen eines Dienstes jede Anforderung konkret betrifft. Dies geschieht in Vorbereitung auf die Definition von Anwendungsfällen für ein SLM-unterstützendes Managementsystem.

Die Orientierung am Dienstlebenszyklus hilft vor allem sicherzustellen, dass das gesamte Funktionsspektrum durch die Primäranwendungsfälle vollständig erfasst wird. Sie stellt außerdem eine Verfeinerung der funktionalen Anforderungen dar. Somit können Anwendungsfallbeschreibungen später leichter abgeleitet werden.

3.2.3 Funktionale Anforderungen an Service-Level-Management

Betrachtet man die vorgestellten Szenarien (Abschnitte 3.1.2, 3.1.3 und 3.1.4) vor dem Hintergrund der SLM-Zielsetzung und Grundannahmen (Abschnitt 3.2.1) und des Dienstlebenszyklus (Abschnitt 3.2.2), so ergeben sich in erster Instanz die folgenden funktionalen Anforderungen an ein umfassendes Service-Level-Management bzw. an ein Managementsystem zur Unterstützung dieser Vorgaben:

F1: VERWALTUNG DER DIENSTE UND ZUGRIFF AUF DIENSTINFORMATIONEN ALS BASIS FÜR ALLE DIENSTVEREINBARUNGEN
--

Zunächst muss eine konsistente Informationsbasis mit allen durch den Provider bereitgestellten oder in Planung befindlichen IT-Diensten existieren, auf die sich alle abgeschlossenen oder abzuschließenden Dienstvereinbarungen (SLAs) beziehen können. Diese Informationsbasis wird Dienstportfolio genannt. SLM- und SLA-relevante Informationen zu allen Diensten müssen ermittelt, gespeichert und administriert werden.

Diese Anforderung ist für alle drei Szenarien relevant. Sie ergibt sich aus der gemeinsamen Tatsache, dass die IT-Provider aller Szenarien ein differenziertes, wenn auch in der Breite unterschiedlich ausgeprägtes Dienstportfolio aufweisen. Jede abgeschlossene Dienstvereinbarung muss sich auf dieses Dienstportfolio beziehen, und es müssen Informationen bereit stehen, auf deren Basis die zu vereinbarenden Dienstfunktionalität und die Dienstperformanz im SLA spezifiziert werden können.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** Während der Planungsphase eines Dienstes werden Dienstinformationen generiert und als Spezifikation des geplanten Dienstes gespeichert. Solche Spezifikationen bilden die Grundlage für alle Dienstkataloge und Dienstvereinbarungen.
- **Verhandlung:** In der Verhandlungsphase wird auf Dienstspezifikationen Bezug genommen, indem Verhandlungen über einen Dienst auf Basis seiner Dienstbeschreibung geführt werden.
- **Bereitstellung:** Bei der Bereitstellung eines Dienstes gilt es, zu gewährleisten, dass die Dienstspezifikation mit den bereitgestellten Mitteln (Ressourcen) wie geplant umgesetzt werden kann.
- **Betrieb:** Ein Dienst muss gemäß seiner Spezifikation betrieben werden. Managementaktivitäten beziehen sich während dieser Phase des Dienstlebenszyklus in der Regel nur auf die Überwachung (Monitoring) des Dienstes, und so wird während der Betriebsphase klassischerweise lesender Zugriff auf entsprechende Informationen aus der Dienstbeschreibung benötigt.
- **Anpassung:** Im Falle der Anpassung eines Dienstes müssen auch die zugehörigen Dienstinformationen gegebenenfalls angepasst und aktualisiert werden.
- **Auflösung:** Wird ein Dienst außer Betrieb genommen, führt dies zur Löschung oder Archivierung der zugehörigen Dienstinformationen.

Eine schwerpunktmäßige Auseinandersetzung mit dem Thema der funktionalen Anforderung F1 im Kontext der Betriebsphase liefert Sailer mit seiner bereits im Abschnitt 1.5 vorgestellten Dissertation [11], die zu diesem Zweck die Konzepte der *Service Management Information Base* (SMIB) und einer *Service Monitoring Architecture* (SMONA) vorstellt. Bei der Realisierung der Anforderung F1 im Rahmen der SLM-Architektur dieser Arbeit werden diese Ansätze an den entsprechenden Stellen aufgegriffen.

F2: VERWALTUNG DER IT-KUNDEN ALS VERTRAGSPARTNER FÜR DIENSTVEREINBARUNGEN

Voraussetzung für die Kernaktivitäten im Service-Level-Management ist, dass der Provider die Dienstnehmerseite kennt. SLM-relevante Informationen über Kunden, die IT-Dienste beauftragen (können), müssen also zur Verfügung stehen. Dazu gehören Informationen über den oder die Standorte eines Kunden, die Anzahl der Nutzer, ihre Arbeits- und damit Dienstnutzungszeiten und vieles andere mehr – festzulegen im Rahmen des Informationsmodells der Architektur.

Diese Anforderung ist für alle drei Szenarien relevant. Sie ergibt sich aus der gemeinsamen Tatsache, dass Dienste in allen Szenarien für unterschiedliche Kunden (wie zum Beispiel unterschiedliche interne Unternehmensabteilungen im Falle der ITG oder verschiedene universitäre Lehrstühle im Falle des LRZ) erbracht werden.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** Bereits bei der Dienstplanung können Informationen über den oder die künftigen (potenziellen) Kunden des geplanten Dienstes erforderlich sein, um den Dienst beispielsweise auf spezifische Kundenanforderungen hin auszurichten oder um ihn korrekt und sinnvoll zu dimensionieren.
- **Verhandlung:** Bei der Aushandlung einer Dienstvereinbarung werden kundenspezifische Informationen benötigt, um eine generische Dienstspezifikation in den Kontext des betreffenden IT-Kunden (Verhandlungspartner) zu setzen und auf diese Weise zu den für diese Dienstvereinbarung gültigen Festlegungen hinsichtlich der Dienstleistung zu gelangen. Dieser Kunden-Kontext besteht unter anderem aus Informationen über Erfüllungsorte oder einzuplanende Workload-Charakteristika (wie viele

Nutzer nutzen den Dienst wie oft und wie lange zu welcher Zeit?). Zusätzlich werden natürlich auch grundlegende, SLM-relevante Kontaktinformationen zu jedem IT-Kunden benötigt.

- **Bereitstellung:** Wird ein Dienst gemäß vorangegangener Verhandlungen an einen Kunden bereitgestellt, so werden insbesondere Informationen über die für einen Kunden relevanten Dienstzugangspunkte benötigt.
- **Betrieb:** IT-Kunden sollten über Verletzungen der vereinbarten Service-Levels informiert werden, sobald diese absehbar sind (vgl. ISO/IEC 20000: *Code of Practice* [31]). Dies ist ein Beispiel, in dem Informationen über Benachrichtigungswege und Kontaktmöglichkeiten eines Kunden während der operativen Phase eines Dienstes verfügbar sein müssen.
- **Anpassung:** Anpassungen von Diensten können durch veränderte kundenseitige Randbedingungen erforderlich werden. Damit sind geänderte Kundeninformationen ein möglicher Auslöser des Übergangs von der Betriebs- in die Anpassungsphase.
- **Auflösung:** Bestimmte Ereignisse, die sich in Änderungen der Kundeninformationen widerspiegeln, können statt der Notwendigkeit einer Anpassung eines oder mehrerer Dienste auch eine Situation schaffen, die die Auflösung eines Dienstes impliziert oder sinnvoll macht. Ein Beispiel hierfür wäre ein Standortwechsel eines Kunden unter der Annahme, dass ein derzeit gelieferter Dienst am neuen Standort nicht mehr bereitgestellt werden kann.

In den Phasen der Bereitstellung und des Betriebs eines Dienstes werden die oben genannten (operativ) relevanten Informationen direkt aus den geschlossenen Dienstvereinbarungen bezogen, sodass die Verwaltung der IT-Kunden als Vertragspartner für SLAs im Kontext dieser Lebenszyklusphasen eines Dienstes keine *direkte* Rolle spielt.

F3: VERWALTUNG DER IT-FACHABTEILUNGEN UND LIEFERANTEN ALS BETEILIGTE PARTEIEN IN DER DIENSTREALISIERUNG/-ERBRINGUNG UND VERTRAGSPARTNER FÜR OLAS UND UCS
--

In Analogie zur Anforderung F2, die sich auf die Dienstnehmerseite bezieht, adressiert diese Anforderung die Notwendigkeit einer Informationsbasis für interne und externe Parteien auf Dienstanbieterseite. So müssen SLM-relevante Informationen über alle in die Realisierung eines Dienstes eingebundenen internen Fachabteilungen und externen Zulieferer vorgehalten werden. Dazu zählen

neben Kontakt- und Erreichbarkeitsinformationen schwerpunktmäßig Informationen über generelle (technische) Verantwortlichkeiten und Kernkompetenzen der einzelnen Parteien sowie im Falle von externen Zulieferern über Möglichkeiten zur Einsicht in die Dienstkataloge, die von diesen Lieferanten bereitgestellt werden.

Diese Anforderung ist für alle drei Szenarien relevant. Sie ergibt sich aus dem Umstand, dass an der Dienstbereitstellung und -erbringung verschiedene Parteien (IT-Fachgruppen oder externe Zulieferer) beteiligt sein können.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** Die Planung eines IT-Dienstes, seiner Bereitstellung und seines Betriebs kann nur sinnvoll unter Einbeziehung derjenigen Parteien erfolgen, die später die operative Verantwortung für seine technische Erbringung übernehmen. Daher werden schon während dieser Lebenszyklusphase Informationen über die IT-Fachabteilungen und mögliche Lieferanten benötigt.
- **Verhandlung:** Primäres Ziel dieser Lebenszyklusphase ist zwar der Abschluss einer Dienstvereinbarung mit einem Kunden (vgl. Abschnitt 3.2.2.1), aus SLM-Sicht ist dieser Aktivität jedoch der Abschluss von operativen Vereinbarungen (OLAs) und Lieferantenverträgen (UCs) vorgelagert (siehe Anforderungen F7 und F8), der damit ebenfalls in den Kontext der Verhandlungsphase fällt. Entsprechend werden Informationen zu IT-Fachabteilungen und Lieferanten in dieser Lebenszyklusphase so weit benötigt, wie es die Verhandlung von OLAs und UC erfordert.
- **Bereitstellung:** In der Phase der Bereitstellung eines IT-Dienstes muss sichergestellt werden, dass alle benötigten Subdienste von den erbringenden Parteien in einer Weise bereitgestellt und aufeinander eingestellt sind, dass der Enddienst in vereinbarter Weise erbracht werden kann. Zur Erfüllung dieser Koordinationsaufgabe werden Informationen über alle beteiligten Parteien benötigt.
- **Betrieb:** Während des Dienstbetriebs können Fehler zu Störungen führen, die durch die zuständigen Support-Gruppen behoben werden müssen. Dies geschieht im Rahmen der mit der Dienstunterstützung befassten Dienstmanagementprozesse (*Resolution Processes* gemäß ISO/IEC 20000) und somit außerhalb des direkten Betrachtungsbereiches von Service-Level-Management. Allerdings entsprechen die dem *First Level Support* nachgelagerten Support-Gruppen in der Regel genau den im

Rahmen von Service-Level-Management betrachteten an der Dienstleistung beteiligten Parteien. Zudem werden die Randbedingungen des Supports wie Reaktions- und Wiederherstellungszeiten im Rahmen von Dienstvereinbarungen festgelegt (siehe auch Anforderung F5).

- **Anpassung:** Anpassungen von Diensten können durch veränderte Randbedingungen auf Seiten der dienstbringenden Parteien erforderlich werden. Damit sind geänderte Informationen über IT-Fachabteilungen oder Lieferanten ein möglicher Auslöser des Übergangs von der Betriebs- in die Anpassungsphase. Umgekehrt können erforderliche Dienstanpassungen zu geänderten Anforderungen an die an der Dienstleistung beteiligten Parteien führen.
- **Auflösung:** Analog zur Bereitstellung muss auch die Auflösung eines IT-Dienstes in geordneter Weise koordiniert werden, wozu Informationen über die an der Dienstleistung beteiligten Parteien erforderlich sind.

F4: BEREITSTELLUNG VON DIENSTKATALOGEN AN DIE IT-KUNDEN

IT-Kunden müssen über die für sie verfügbaren IT-Dienste informiert sein. Ein SLM-unterstützendes Managementsystem muss zu diesem Zweck die Erstellung, Bereitstellung und Administration von Dienstkatalogen unterstützen, die eine gegebenenfalls kundenspezifische Sicht auf das Dienstportfolio realisieren.

Diese Anforderung ist eine Kernanforderung aus Szenario 3 (FMG) und ergibt sich zusätzlich aus der Grundannahme 3. Da der IT-Provider der FMG seine Dienste sowohl internen als auch an externen Kunden bereitstellt und das Dienstportfolio dabei gleichzeitig sehr breit ist (vgl. morphologische Analyse, Tabelle 3.7), müssen unterschiedliche kundenspezifische Dienstkataloge gepflegt werden, da ein einziger umfassender Katalog, der eine festgelegte Teilmenge des Dienstportfolios abdeckt, nicht zweckmäßig wäre.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** Die Planung eines konkreten Dienstes endet mit der Aufnahme des Dienstes in einen oder mehrere Dienstkataloge, wodurch Kunden die Möglichkeit zur Beauftragung erhalten und somit der Übergang in die Verhandlungsphase erfolgt.
- **Verhandlung:** Ausgangspunkt für die Verhandlungsphase eines Dienstes ist die Auswahl des Dienstes aus dem einem Dienstkatalog.

- Bereitstellung: —
- Betrieb: —
- Anpassung: Wird ein Dienst modifiziert, so müssen diese Änderungen auch in den Dienstkatalogen berücksichtigt werden.
- Auflösung: Die Außerbetriebnahme eines Dienstes führt zur Löschung des Dienstes aus allen Dienstkatalogen.

In den Phasen der Bereitstellung und des Betriebs eines Dienstes spielen Dienstkataloge keine wesentliche Rolle. Vom Provider benötigte Informationen werden direkt auf dem Dienstportfolio (siehe Anforderung F1) bezogen, auf das ein Dienstkatalog ja nur eine kundenspezifische Teilsicht realisiert.

F5: ANGEBOT VERSCHIEDENER DIENSTUNABHÄNGIGER SUPPORT-LEVELS FÜR IT-KUNDEN ZUR BERÜCKSICHTIGUNG IN DIENSTVEREINBARUNGEN
--

Beim Abschluss von Dienstvereinbarungen zwischen dem Provider und seinen IT-Kunden werden auch Vereinbarungen zur Unterstützung (Support) der Dienstnutzung sowie zur Entstörung im Fehlerfall getroffen. Ein SLM-unterstützendes Managementsystem muss dem Provider die Möglichkeit geben, verschiedene Support-Level-Profile zu definieren, aus denen ein IT-Kunde beim Abschluss einer Dienstvereinbarung in Abhängigkeit von den jeweiligen Anforderungen an Unterstützung und Entstörung auswählen kann.

Diese Anforderung ist eine Kernanforderung aus den Szenarien 2 (LRZ) und 3 (FMG). Sie ergibt sich vor allem aus dem großen, heterogenen Kundenumfeld und dem im Vergleich zu Szenario 1 (ITG) breiteren Dienstangebot. Ein einziges gültiges Support-Level für alle Dienste und SLAs könnte die heterogenen kunden- und dienstspezifischen Anforderungen an Unterstützung und Entstörung nicht abdecken. Der Verzicht auf klar definierte Support-Level-Profile zu Gunsten individueller Festlegungen bei jeder SLA-Verhandlung würde nicht nur die Verhandlungsphase verlangsamen, sondern schnell auch zu einer kaum noch handhabbaren Komplexität in den Unterstützungsgarantien führen, was sich negativ auf die Effektivität und Effizienz der Support-Prozesse auswirken könnte.

Diese Anforderung im Kontext des Dienstlebenszyklus

- Planung: —

- **Verhandlung:** Wählbare Support-Levels müssen in der Verhandlungsphase eines Dienstes zur Verfügung stehen. Beim Abschluss einer Dienstvereinbarung (SLA) wird durch den Kunden ein Support-Level ausgewählt.
- **Bereitstellung:** —
- **Betrieb:** Die Betriebsphase eines Dienstes erfordert den Zugriff auf Informationen zu Support-Levels im Falle einer Support-Anfrage, insbesondere bei der Entstörung.
- **Anpassung:** Die Dienstanpassung kann die Selektion eines anderen Support-Levels beinhalten; die entsprechende Informationsbasis muss also auch in dieser Phase verfügbar sein.
- **Auflösung:** —

Die Planung eines Dienstes ist in aller Regel unabhängig von der Festlegung von Support-Level-Profilen, da diese im Normalfall keine speziellen Dienstcharakteristika beinhalten. In den Phasen der Dienstbereitstellung und Auflösung spielen Support-Aspekte generell keine Rolle.

F6: VERWALTUNG VON DIENSTVEREINBARUNGEN (SLAs) ÜBER IHREN GESAMTEN LEBENSZYKLUS

Dienstvereinbarungen stellen ein zentrales Konzept im Service-Level-Management dar und müssen über ihren gesamten Lebenszyklus administriert werden. Dieser beinhaltet Abschluss, Anpassung und Aufhebung der entsprechenden Vereinbarungen.

Diese Anforderung ist für alle drei Szenarien relevant und eine zwingende Kern-Anforderung an ein SLM-unterstützendes Managementsystem. Sie ergibt sich zudem aus der Grundannahme 1.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** —
- **Verhandlung:** In der Verhandlungsphase eines Dienstes wird eine Dienstvereinbarung geschlossen und muss entsprechend für künftigen Zugriff hinterlegt werden.

- **Bereitstellung:** Im Rahmen der Dienstbereitstellung müssen die technischen Voraussetzungen zur Erfüllung der in den Dienstvereinbarungen definierten Service-Level-Ziele geschaffen werden. Somit wird in dieser Phase Zugriff auf die entsprechenden Informationen benötigt.
- **Betrieb:** Zwar nicht für die eigentliche operative Ausführung, wohl aber im Rahmen der Dienstunterstützung müssen Dienstvereinbarungen regelmäßig als Informationsquelle und Entscheidungsbasis herangezogen werden, um beispielsweise das zutreffende Support-Level oder Priorisierungsinformationen zu ermitteln.
- **Anpassung:** Die Veränderung eines Dienstes, seines Nutz- oder Leistungswertes aus Sicht des Kunden bedingt auch Anpassungen der für diesen Dienst bestehenden Dienstvereinbarungen.
- **Auflösung:** Wird ein Dienst außer Betrieb genommen, so geht damit auch die Auflösung der verbundenen Dienstvereinbarungen einher.

F7: VERWALTUNG VON OPERATIVEN VEREINBARUNGEN (OLAs) ÜBER IHREN GESAMTEN LEBENSZYKLUS
--

Operative Vereinbarungen (*Operational Level Agreements*, OLAs) mit internen IT-Fachgruppen stellen einen wichtigen Baustein in der Sicherstellung der Realisierung von Service-Level-Zielen dar, wie sie durch Dienstvereinbarungen vorgegeben sind. In operativen Vereinbarungen werden Zielvereinbarungen mit an der Dienstleistung beteiligten IT-Fachabteilungen dokumentiert. OLAs stellen somit sicher, dass jede an der Dienstleistung beteiligte interne Partei ihren erforderlichen (zumeist technischen) Beitrag zu Funktionalität und Performanz jedes Dienstes kennt und dass Service-Level-Ziele nur vor dem Hintergrund dessen vereinbart werden, was auch tatsächlich technisch zu den gegebenen Randbedingungen (Infrastruktur, Abhängigkeiten, Kosten, etc.) realisierbar ist.

Diese Anforderung ist für alle drei Szenarien relevant. Sie ergibt sich außerdem aus der Grundannahme 2.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** Bereits bei der Planung eines Dienstes müssen die realisierenden internen Fachabteilungen einbezogen und, wenn sinnvoll, bereits erste operative Vereinbarungen getroffen werden.

- **Verhandlung:** Die Verhandlungsphase eines Dienstes bezieht sich primär auf die Verhandlung einer zugehörigen Dienstvereinbarung, also eines SLAs. OLAs müssen als unterstützende Vereinbarungen prinzipiell bereits vorher geschlossen werden (siehe Planungsphase). Allerdings können sich während der Verhandlung eines SLAs mit einem Kunden Zusatzanforderungen ergeben, die wiederum mit internen Fachabteilungen koordiniert werden müssen, was Ergänzungen oder Änderungen der bestehenden OLAs impliziert. Daher tangiert die Verwaltung von OLAs ebenso die Verhandlungsphase eines Dienstes.
- **Bereitstellung:** Wird der Dienst technisch auf seinen Betrieb vorbereitet, beziehen die Dienst-realisierenden Fachabteilungen wichtige Informationen zu Verantwortlichkeiten und getroffenen Vereinbarungen – die Dienstbereitstellung betreffend – aus den geschlossenen OLAs.
- **Betrieb:** Auch zum Dienstbetrieb können OLAs Vereinbarungen enthalten, auf die sowohl der Provider als auch die Diensterbringer-Parteien während dieser Lebenszyklusphase zugreifen müssen. Beispiele umfassen unter anderem Festlegungen zu Eskalationsverfahren im Rahmen der Dienstunterstützung oder zum Umgang mit allen Arten von Anfragen bezüglich eines Dienstes.
- **Anpassung:** Die Modifikation eines Dienstes kann Implikationen auf bestehende SLAs haben (vgl. Anforderung F6) und somit auch Änderungen an operativen Vereinbarungen bedingen, die diesen SLAs vorgeschaltet sind.
- **Auflösung:** Wird ein Dienst außer Betrieb genommen, so können OLAs oder Teile daraus, die sich auf diesen Dienst beziehen, aufgehoben werden.

F8: VERWALTUNG VON LIEFERANTENVERTRÄGEN (UCs) ÜBER IHREN GESAMTEN LEBENSZYKLUS
--

Lieferantenverträge (*Underpinning Contracts*, UCs) spielen für alle IT-Provider eine wichtige Rolle, die in der Diensterbringung von Dienstleistungen externer Zulieferer (Sub-Provider) abhängig sind. Diese UCs sind im Prinzip identisch mit SLAs, wobei der betrachtete IT-Provider die Rolle des Kunden einnimmt.

Diese Anforderung ist eine Kernanforderung aus Szenario 1 (ITG) und 3 (FMG) sowie eine sinnvolle Anforderung aus Szenario 2 (LRZ). Nur bei gering

ausgeprägten Abhängigkeiten von externen Lieferanten kann gegebenenfalls auf eine formale Dokumentation der Zielvereinbarungen zur Subdienstleistung verzichtet werden. In allen anderen Fällen muss dies aber als funktionale Kernanforderung an ein SLM-unterstützendes Managementsystem angesehen werden. Wird beispielsweise ein Telekommunikationsdienst von einem externen Dienstleister geliefert, der die Grundlage für einen oder mehrere Dienste darstellt, über die Dienstvereinbarungen mit IT-Kunden bestehen, so muss der extern bezogene Telekommunikationsdienst einem Lieferantenvertrag unterliegen, der die Dienstvereinbarungen untermauert.

Diese Anforderung im Kontext des Dienstlebenszyklus Die Betrachtung der Anforderung F8 im Kontext des Dienstlebenszyklus ist weitgehend analog zur Dienstlebenszyklus-Betrachtung im Rahmen von Anforderung F7.

- **Planung:** Während der Planung eines Dienstes müssen die realisierenden externen Zulieferer einbezogen und, sofern sinnvoll, erste Lieferantenverträge getroffen werden.
- **Verhandlung:** Falls sich bei der Verhandlung einer Dienstvereinbarung mit einem Kunden Zusatzanforderungen ergeben, so müssen die bestehenden Lieferantenverträge gegebenenfalls angepasst werden.
- **Bereitstellung:** In der Bereitstellungsphase beziehen die an der Dienstleistung beteiligten externen Zulieferer wichtige Informationen zu Verantwortlichkeiten und getroffenen Vereinbarungen – die Dienstbereitstellung betreffend – aus den geschlossenen Verträgen.
- **Betrieb:** Bezüglich des Dienstbetriebs können Lieferantenverträge Vereinbarungen enthalten, auf die der Provider und die externen Zulieferer während dieser Lebenszyklusphase zugreifen müssen.
- **Anpassung:** Die Anpassung eines Dienstes kann Auswirkungen auf bestehende UCs haben.
- **Auflösung:** Wird ein Dienst außer Betrieb genommen, so können Lieferantenverträge oder Teile daraus, die sich auf diesen Dienst beziehen, aufgehoben werden.

F9: ANALYSE VON KONFLIKTEN ZWISCHEN UNTERSCHIEDLICHEN VEREINBARUNGEN UND VERTRÄGEN
--

Bedingt durch Grundannahme 2, die Untermuerung jeder Dienstvereinbarung mittels operativer Vereinbarungen und/oder Liferantenverträgen, können zwischen Dienstvereinbarungen, operativen Vereinbarungen und Lieferantenverträgen vielfältige Beziehungen und Abhängigkeiten entstehen. Ziel muss es sein, dass aufeinander aufsetzende Vereinbarungen keine konfliktären Ziele enthalten. Die Möglichkeit der Analyse von Vereinbarungen und Verträgen zur Identifikation von (potenziellen) Konflikten ist eine Anforderung an ein SLM-unterstützendes Managementsystem.

Diese Anforderung ist für alle drei Szenarien relevant. Sie ergibt sich außerdem aus der Grundannahme 2. Bezieht das LRZ beispielsweise einen Subdienst von einem externen Zulieferer, für den eine durchschnittliche Verfügbarkeit von 98 Prozent im Rahmen des betreffenden UCs vereinbart wurde, so kann das LRZ für keinen auf Basis dieses Subdienstes realisierten Enddienst eine höhere Verfügbarkeit garantieren. Die rechtzeitige Erkennung solcher Konflikte bzw. ihre Verhinderung, bevor sie entstehen, hat großen Einfluss auf die Belastbarkeit von Qualitätsgarantien im Rahmen von Dienstvereinbarungen.

Diese Anforderung im Kontext des Dienstlebenszyklus

- Planung: —
- Verhandlung: Sofern die Konfliktanalyse an den Dienstlebenszyklus gekoppelt wird, macht diese am ehesten im Kontext der Verhandlungsphase Sinn, wenn also neue SLAs verhandelt oder bestehende neu verhandelt werden.
- Bereitstellung: —
- Betrieb: —
- Anpassung: —
- Auflösung: —

Grundsätzlich ist die Konfliktanalyse entkoppelt von den Lebenszyklusphasen eines Dienstes.

F10: MESSUNG UND AUFZEICHNUNG VON IN SLAs VEREINBARTEN DIENSTGÜTEPARAMETERN

Die in SLAs vereinbarten Leistungsparameter müssen gemessen werden. Berichte (*reports*) über die erreichte Dienstqualität und ein Soll-Ist-Vergleich müssen generiert werden.

Diese Anforderung ist für alle drei Szenarien relevant.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** In der Planungsphase eines Dienstes muss auch die spätere Dienstmessung mitgeplant werden. Dazu müssen potenzielle Leistungsparameter und die zugehörigen erforderlichen Messverfahren sowie die benötigte Messtechnik festgelegt werden.
- **Verhandlung:** —
- **Bereitstellung:** Während der Phase der Vorbereitung eines Dienstes zum operativen Betrieb müssen alle Maßnahmen ergriffen werden, die notwendig sind, um die Messung und Aufzeichnung der vereinbarten Leistungsparameter im Rahmen des Dienstbetriebs sicherzustellen. Grundlage hierfür sind Dienstvereinbarungen und die darin enthaltenen Festlegungen bezüglich des Berichtswesens (siehe Anforderung F11). Sofern für die Berichterstellung oder die ihr zugrunde liegende Dienstüberwachung Vorbereitungen zu treffen sind (zum Beispiel die Einrichtung von Messtechnik, Geräten, Überwachungssystemen, etc.), so geschieht dies während der Bereitstellungsphase des betreffenden Dienstes.
- **Betrieb:** Dienstgüteparameter müssen während der operativen Betriebsphase eines Dienstes gemessen und aufgezeichnet werden.
- **Anpassung:** Die Anpassung eines Dienstes kann Änderungen an den zur Verfügung stehenden Parametern oder Messverfahren nach sich ziehen. Umgekehrt können dringend benötigte Messgrößen auch eine Änderung des Dienstes erforderlich machen.
- **Auflösung:** Wird ein Dienst aufgelöst, so werden auch alle Mess- und Aufzeichnungsaktivitäten im Zusammenhang mit diesem Dienst beendet.

F11: BERICHTSWESEN (REPORTING)

Das Berichtswesen umfasst die Erstellung von Berichten über die Dienstbringung in einem definierten, zurückliegenden Zeitraum. Ziel jedes Berichts ist

die differenzierte Auseinandersetzung mit dem tatsächlich erreichten Nutz- und Leistungswert im Vergleich zu den durch Dienstvereinbarungen und operative Vereinbarungen vorgegebenen Zielen. Daneben müssen Berichte insbesondere auch Informationen zu außerplanmäßigen Ereignissen reflektieren. Dazu gehören SLA-Verstöße und gegebenenfalls resultierende Strafzahlungen (Pönalen), überschrittene Schwellenwerte und generell die Dokumentation von Ausnahme-Ereignissen. Adressaten von Berichten sind sowohl die IT-Kunden als auch der Provider selbst. Eine Differenzierung zwischen externen und internen Berichten ist vor diesem Hintergrund sinnvoll. Ergänzend zu den regulär geplanten Berichten können bestimmte Ereignisse mit gravierender Auswirkung auf zugesicherte Diensteseigenschaften außerplanmäßige Mitteilungen an IT-Kunden erforderlich machen.

Diese Anforderung ist für alle drei Szenarien relevant.

Diese Anforderung im Kontext des Dienstlebenszyklus

- Planung: —
- Verhandlung: In der Verhandlungsphase wird vereinbart, welche Metriken unter Berücksichtigung ihrer jeweiligen Aussagekraft für den Kunden wie gemessen werden. Struktur und Inhalte von Berichten werden festgelegt. Dabei wird auch definiert, ob und wie bestimmte Messergebnisse im Rahmen der Berichterstellung aggregiert und/oder visualisiert werden. Daneben wird festgelegt, wann welche Berichte jeweils geliefert werden müssen.
- Bereitstellung: —
- Betrieb: Berichte werden in der Regel periodisch während der operativen Phase des Dienstbetriebs gemäß der Vorgaben in den Dienstvereinbarungen erstellt. Außerplanmäßige Mitteilungen können durch bestimmte Ereignisse wie gravierende SLA-Verstöße ausgelöst werden.
- Anpassung: Anpassungen am Dienst können sich auf die Berichtsaktivitäten auswirken. Umgekehrt können Berichtsergebnisse dazu führen, dass Anpassungen an Diensten vorgenommen werden.
- Auflösung: Die Auflösung eines Dienstes führt auch zur Beendigung der Berichtsaktivitäten im Zusammenhang mit diesem Dienst.

Bemerkung: Die Anforderungen F10 und F11 sind eng miteinander verweben: Während sich F10 ganz grundsätzlich mit Aspekten der technischen Überwachung und Aufzeichnung des Dienstbetriebs beschäftigt, formuliert F11 die Anforderung eines Berichtswesens, das auf der durch F10 geforderten Messung und Aufzeichnung von Dienstgüteparametern basiert.

F12: SLA-BASIERTE DIENSTUNTERSTÜTZUNG UND ENTSTÖRUNG

Klassische Dienstmanagement-Aufgaben in der operativen Phase eines Dienstes sind Unterstützung (Support) und Entstörung. In beiden Fällen ist es hilfreich, wenn ein SLM-unterstützendes Managementsystem entscheidungs- oder planungsrelevante Informationen bereithält bzw. ermitteln kann. Hierunter fällt zum Beispiel die Bestimmung der Dringlichkeit einer Benutzeranfrage (*Service Request*) oder Störungsmeldung (*Incident Notification*) auf belastbarer Basis und somit auf Grundlage der bestehenden Dienstvereinbarungen.

Diese Anforderung ist für alle drei Szenarien relevant. Selbst in Infrastrukturen mittlerer Größe spielt Entstörung (*Incident Management*) eine wichtige Rolle. Und üblicherweise ist die Instanziierungshäufigkeit des *Incident Management*-Prozesses bereits in kleinen IT-Organisationen so hoch, dass eine Priorisierung von Störungen und Anfragen unerlässlich ist. Diese Priorität unter anderem auf der Grundlage der geschlossenen SLAs zu ermitteln, ist sogar eine Forderung praktisch jedes Dienstmanagement-Rahmenwerks sowie des Standards ISO/IEC 20000.

Diese Anforderung im Kontext des Dienstlebenszyklus

- Planung: —
- Verhandlung: Während der Verhandlungsphase eines Dienstes wird im Rahmen der Dienstvereinbarung festgelegt, wie Störungen und Nutzeranfragen bezüglich dieses Dienstes zu priorisieren sind. Ein dienstspezifisches Priorisierungsschema und ein zugehöriges Priorisierungsverfahren werden definiert, um während der Betriebsphase in jeder Situation eine eindeutige Priorität festlegen zu können. Die konkrete Ausgestaltung des Priorisierungsschemas hängt maßgeblich von der Geschäftskritikalität des betreffenden Dienstes aus Sicht des IT-Kunden ab.
- Bereitstellung: —

- Betrieb: Support im Allgemeinen und Entstörung (*Incident Handling*) als spezifischer Teilaspekt hiervon werden vom Provider während der operativen Dienstlebenszyklusphase, dem Dienstbetrieb, geleistet. Somit werden Informationen aus Dienstvereinbarungen, die zur effektiveren, effizienteren oder kundenorientierteren Dienstunterstützung herangezogen werden, auch primär in dieser Lebenszyklusphase benötigt.
- Anpassung: —
- Auflösung: —

F13: SLA-BASIERTE LEISTUNGSVERRECHNUNG FÜR IT-DIENSTE

In SLAs können Informationen, die zur Leistungsverrechnung (*Charging*) herangezogen werden, definiert werden. Denkbar sind Festpreise für die Dienstnutzung (Flatrates) oder nutzungsabhängige Preise und Tarife.

Diese Anforderung ist insbesondere für Szenario 2 und 3 relevant, da nur hier IT-Dienstleistungen an die empfangenden Kunden verrechnet werden. Die Verrechnung direkt an Dienstvereinbarungen zu koppeln, stellt sicher, dass IT-Dienste gemäß der gültigen und vereinbarten Konditionen abgerechnet werden und sich Änderungen dieser Konditionen in den SLAs unmittelbar auf den Verrechnungsprozess auswirken. Da Leistungsverrechnung prinzipiell auch ohne Kopplung an SLAs realisierbar ist, handelt es sich hierbei nicht um eine Kern-Anforderung ans Service-Level-Management, sondern für die relevanten Szenarien 2 und 3 um eine optionale Anforderung.

Diese Anforderung im Kontext des Dienstlebenszyklus

- Planung: Sofern Leistungsverrechnung für einen Dienst durchgeführt werden soll, muss dieser Aspekt bereits in der Phase der Dienstplanung Niederschlag finden. Beispielsweise muss durch die Planung sichergestellt sein, dass der fertige Dienst adäquate Abrechnungseinheiten (*accountable units*) mit sich bringt.
- Verhandlung: Verrechnungskonditionen und -verfahren werden im Rahmen der Verhandlungsphase zwischen Provider und Kunde festgelegt und in Dienstvereinbarungen festgehalten.
- Bereitstellung: In der Bereitstellungsphase müssen alle Maßnahmen ergriffen werden, die erforderlich sind, um die Leistungsverrechnung des Dienstes auf Basis von SLAs zu realisieren.

- Betrieb: Verrechnungsinformationen aus Dienstvereinbarungen werden in der Phase des Dienstbetriebs benötigt, weil während der operativen Ausführung eines Dienstes die für den jeweiligen Kunden entstehenden Kosten zur Leistungsverrechnung ermittelt werden.
- Anpassung: Eine Anpassung des Dienstes kann Änderungen an den Verrechnungsverfahren oder -konditionen erforderlich oder sinnvoll machen.
- Auflösung: Nach Auflösung eines Dienstes können auch alle damit verbundenen Verrechnungsaktivitäten eingestellt werden.

F14: SLA-BASIERTE TRANSITION VON IT-DIENSTEN IN DIE OPERATIVE INFRASTRUKTUR

Wenn im Kontext von ITIL von der Dienstlebenszyklusphase *Service Transition* die Rede ist, fallen darunter klassischerweise die Dienstmanagementdisziplinen und -prozesse *Change Management* und *Release Management* sowie zum Teil auch *Configuration Management*. Aktive Dienstvereinbarungen und darin vereinbarte Service-Level-Ziele können in der Transitionsphase des Dienstlebenszyklus – also bei der Überführung eines neuen oder geänderten Dienstes in die operative Infrastruktur – eine wichtige Rolle spielen. Als Beispiel ist im *Change Management* die Bestimmung der Kategorie einer (geplanten) Veränderung oder die Ermittlung des idealen Zeitfensters für die Implementierung der Änderung zu nennen.

Ein SLM-unterstützendes Managementsystem sollte also in der Lage sein, Aktivitäten und Entscheidungen im Kontext der Diensttransition durch fundierte Auswertung bestehender Dienstvereinbarungen zu unterstützen.

Diese Anforderung ist für die Szenarien 2 und 3 relevant, in denen die Komplexität und Dynamik der Dienste hoch ist, was eine hohe Frequenz von Änderungen (*Changes*) und Freigaben *Releases* impliziert.

Diese Anforderung im Kontext des Dienstlebenszyklus

- Planung: —
- Verhandlung: Während der Verhandlungsphase eines Dienstes wird im Rahmen der Dienstvereinbarung festgelegt, wie Änderungen bezüglich dieses Dienstes zu priorisieren und zu kategorisieren sind. Die Kategorie einer Veränderung orientiert sich an deren Risiko und damit an den Fragen, wie wahrscheinlich ein ungeplanter Ausfall aufgrund einer Änderung

ist und wie groß in diesem Fall die Auswirkung auf die geschäftlichen Aktivitäten des Kunden sind.

- **Bereitstellung:** Im Rahmen der Überführung eines Dienstes in die operative Infrastruktur, also während der Bereitstellungsphase, können Informationen aus Dienstvereinbarungen verwendet werden, um die mit der Transition verbundenen Änderungen und Freigaben SLA-konform und damit vereinbarungsgemäß durchzuführen. Hierzu zählt beispielsweise eine mit SLAs zu vereinbarende Zeitplanung oder die korrekte Determinierung von Priorität und Kategorie aller mit der Dienstbereitstellung verbundenen Änderungen.
- **Betrieb:** —
- **Anpassung:** —
- **Auflösung:** —

F15: SLA-BASIERTE REALISIERUNG DER INFORMATIONSSICHERHEIT

Schließlich können SLAs als Grundlage für die Realisierung einer SLA-konformen Informationssicherheit einschließlich der Einrichtung entsprechender Zugriffsschutzmechanismen herangezogen werden.

Szenario 2 stellt eine Quelle für diese Anforderung dar. Das LRZ bietet verschiedene Kommunikationsdienste an, bei denen der Kunde das Niveau der Informationssicherheit in definierten Grenzen oder aus vorgegebenen Sicherheitsstufen selbst wählen kann. Oftmals ist ein ein Mehr an Sicherheit verbunden mit größeren Nutzungsbarrieren, einer höheren Komplexität in der Ausrolung des Dienstes (zum Beispiel, wenn Nutzerzertifikate verteilt und gemanagt werden müssen), erschwelter Bedienbarkeit oder höheren Kosten der Bereitstellung. Es macht Sinn, dass von einem Kunden geforderte Sicherheitsniveau im Rahmen einer Dienstvereinbarung festzulegen und bei der Bereitstellung des Dienstes die Erfüllung der Anforderungen an die Informationssicherheit gemäß SLA sicherzustellen.

Diese Anforderung im Kontext des Dienstlebenszyklus

- **Planung:** Bei der Planung eines Dienstes können verschiedene denkbare Sicherheitsstufen bereits eingeplant werden.

- **Verhandlung:** In der Verhandlungsphase werden Sicherheitsanforderungen an den Dienst durch den Kunden im Rahmen der Dienstvereinbarung festgelegt.
- **Bereitstellung:** Bei der Vorbereitung eines Dienstes auf seine Betriebsphase werden die Sicherheitsvorgaben aus der Dienstvereinbarung umgesetzt.
- **Betrieb:** Während des Betriebs muss sichergestellt sein, dass das durch die Dienstvereinbarung garantierte Sicherheitsniveau zu jeder Zeit eingehalten wird.
- **Anpassung:** Wird ein Dienst verändert, so kann sich dies auf die bereits getroffenen Vereinbarungen zur Informationssicherheit auswirken.
- **Auflösung:** Mit der Beendigung eines Dienstes entfallen auch dienstspezifische Anforderungen an die Informationssicherheit.

Zwischenfazit: Diese Anforderungen F1 bis F15 stellen nicht nur ein übergeordnetes Funktionsspektrum eines SLM-unterstützenden Managementsystems dar, sondern können auch ganz allgemein als Anforderungen an ein funktionierendes Service-Level-Management verstanden werden – ganz unabhängig von jeglichen Automatisierungsaspekten.

Abbildung 3.10 beschließt die Ableitung der allgemeinen Anforderungen mit einer groben Einordnung. Während die Aufgabenbereiche, die durch die Anforderungen F6 bis F11 gegeben sind, typischerweise direkt mit Service-Level-Management in Verbindung gebracht werden, stellen die Anforderungen F1 bis F5 das Fundament hierfür dar. Im Gegensatz dazu erweitern F12 bis F15 den klassischen funktionalen Betrachtungs- und Wirkungsbereich von Service-Level-Management um wichtige Aspekte, die das Verhältnis zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen zum Ausdruck bringen.

3.2.4 Auswertung der allgemeinen Funktionsanforderungen

Ziel dieser Arbeit ist es, alle aufgeführten allgemeinen funktionalen Anforderungen (F1 bis F15) zu adressieren. Somit geht die Arbeit von einem erweiterten funktionalen Betrachtungsbereich (*Scope*) für Service-Level-Management

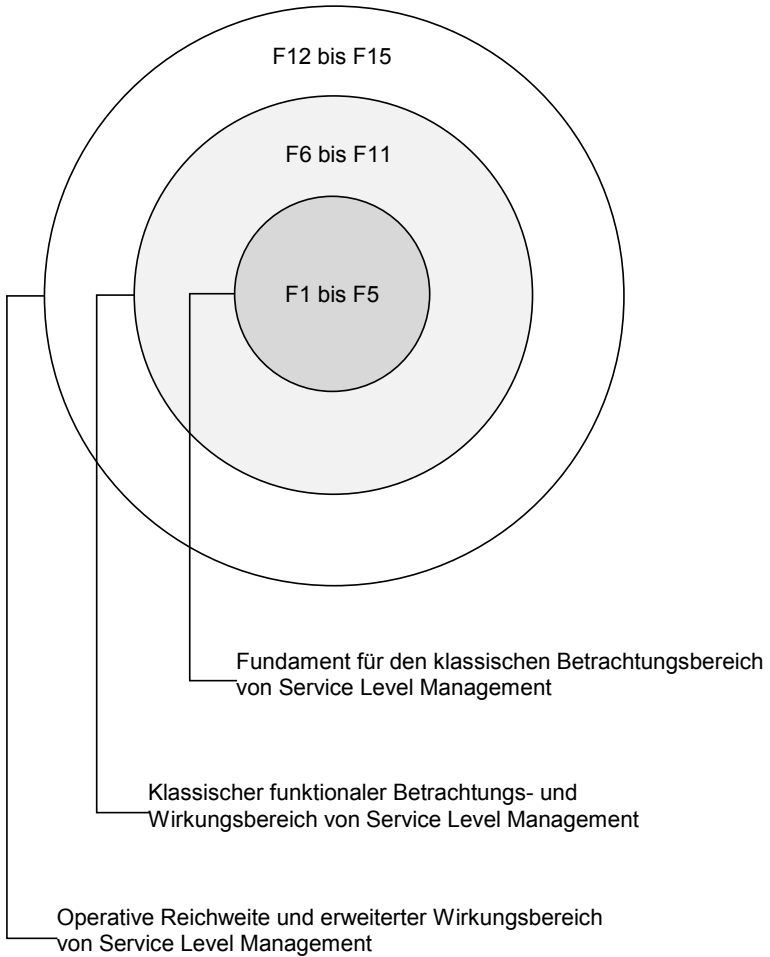


Abbildung 3.10: Einordnung des ermittelten Funktionsspektrums

aus. Allerdings können aus Gründen des Umfangs und der Komplexität der damit verbundenen Problemstellungen nicht alle Teilaspekte in der gleichen Ausführlichkeit behandelt werden.

Generell gilt deswegen: Speziell im äußeren Ring der Abbildung 3.10 (Anforderungen F12 bis F15) werden im Rahmen des Modellansatzes nur einige ausgewählte Aspekte umgesetzt. Zudem gibt die Tabelle 3.1 zusammenfassend wieder, welche Bedeutung die einzelnen Anforderungen in den Szenarien haben. Dabei wird unterschieden zwischen:

- ✓ : Die Anforderung ist eine Kernanforderung für dieses Szenario
- (✓) : Die Anforderung ist eine relevante, aber optionale Anforderung für dieses Szenario
- × : Die Anforderung ist für das Szenario nicht relevant

Anforderung	Szenario 1 (ITG)	Szenario 2 (LRZ)	Szenario 3 (FMG)
F1	✓	✓	✓
F2	✓	✓	✓
F3	✓	✓	✓
F4	✓	(✓)	✓
F5	×	✓	✓
F6	✓	✓	✓
F7	✓	✓	✓
F8	(✓)	(✓)	✓
F9	✓	✓	✓
F10	✓	✓	✓
F11	✓	✓	✓
F12	✓	✓	✓
F13	×	(✓)	✓
F14	×	✓	✓
F15	×	✓	×

Tabelle 3.1: Zuordnung der funktionalen Anforderungen zu den Szenarien

Auch diese Kategorisierung gibt einen Hinweis darauf, wie stark die einzelnen Anforderungen jeweils zu gewichten sind.

3.2.5 Zusammenfassung und nächste Schritte

Die 15 abgeleiteten Funktionsanforderungen umreißen grob das *funktionale Spektrum* eines IT-gestützten Managementsystems für Service-Level-Management. Alle Anforderungen wurden auf Basis der SLM-Zieldefinition aus den vorgestellten Szenarien abgeleitet, und jede Anforderung ist somit in mindestens einem der vorgestellten Szenarien relevant.

Um dieses Bild weiter zu verfeinern, werden im nächsten Abschnitt *konkrete Anwendungsfälle* definiert, die die Anforderungen F1 bis F15 auf eine feinergranulare Sichtweise herunterbrechen und die somit als Primäranwendungsfälle eines SLM-unterstützenden Managementsystems fungieren. Um die Menge der Primäranwendungsfälle vollständig zu erfassen, wird nicht nur auf die funktionalen Anforderungen F1 bis F15 zugegriffen, sondern zusätzlich berücksichtigt, welche Phasen des Dienstlebenszyklus jede dieser Anforderungen berührt. Die dazu erforderliche Abbildung von Anforderungen auf Lebenszyklusphasen wurde zu diesem Zweck bereits vorgenommen.

Anhand von ausgewählten Beispielen wird der Bezug der Primäranwendungsfälle zu den Szenarien hergestellt. Dies schlägt eine Brücke zurück zum Ausgangspunkt und soll vor allem darstellen, dass die Primäranwendungsfälle plausibel sind.

3.3 Primäranwendungsfälle

Wie in Abbildung 3.11 dargestellt, wird das konkrete SLM-unterstützende Managementsystem zunächst als *Black Box* angesehen. Es wird also angenommen, dass ein IT-gestütztes Managementsystem realisierbar ist – ohne Kenntnis über das Design und die Implementierungsdetails dieses Systems. Entscheidend ist, dass die nachfolgend dargestellten Primäranwendungsfälle durch das Managementsystem unterstützt werden. Mit anderen Worten: Die Anwendungsfälle beschreiben, *was* das IT-gestützte Managementsystem können muss, aber nicht *wie* diese Merkmale zu realisieren sind, leisten somit also einen wichtigen und zentralen Beitrag zur funktionalen Anforderungsanalyse.

Für die Beschreibung der Primäranwendungsfälle wird auf die Terminologie Bezug genommen, die in Abschnitt 2.3 (Begriffsbildung) eingeführt wurde. Insbesondere wird das Domänenmodell mit Differenzierung zwischen den Organisationsdomänen *Provider* und *Kunde* mit Erweiterung um die Domäne *Lieferant* zugrunde gelegt. Die Lieferanten- oder auch Sub-Provider-Domäne ist kein expliziter Teil der eingeführten Ausprägung des MNM-Dienstmodells,

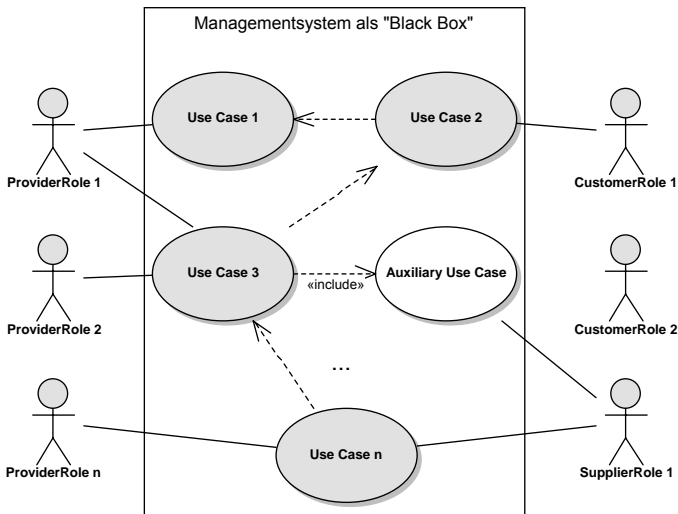


Abbildung 3.11: Das Managementsystem als *Black Box*

wird aber im Zusammenhang mit verketteter Dienstbringung durch eine Methodik zur rekursiven Anwendung des Modells von Garschhammer et al. adressiert [27].

Wie werden auf Basis des Funktionsspektrums Anwendungsfälle abgeleitet? Ein sehr nützliches Werkzeug ist an dieser Stelle – wie bereits zuvor erwähnt – der generische Dienstlebenszyklus, der ja bereits im Kontext der einzelnen Funktionsanforderungen betrachtet wurde. Das erweist sich nun erneut als hilfreich. Für jeden Bestandteil des Funktionsspektrums, d.h. für jede der Anforderungen F1 bis F15, wurde bereits analysiert, welche Phasen des Dienstlebenszyklus in welcher Weise durch die jeweilige Anforderung tangiert werden. Dadurch wurde jede Anforderung aus dem Funktionsspektrum von verschiedenen Seiten „beleuchtet“. Durch die Primäranwendungsfälle muss gewährleistet sein, dass alle auf diese Weise identifizierten Facetten jeder Anforderung vom Managementsystem abgedeckt werden.

Im einfachsten Fall ergibt sich damit aus einer Anforderung aus dem Funktionsspektrum genau ein Primäranwendungsfall. In den meisten Fällen reicht dies jedoch nicht aus. Tabelle 3.2 zeigt, welche realisierenden Primäranwendungsfälle sich ausgehend von den funktionalen Anforderungen F1 bis F15

ergeben. Im nächsten Abschnitt werden diese Anwendungsfälle definiert.

<i>Anforderung</i>	<i>Realisierende Anwendungsfälle</i>
F1	1, 2, 3
F2	4, 5, 6
F3	16, 17, 18
F4	7, 8, 9, 10, 11, 12
F5	13, 14, 15
F6	19, 20, 21
F7	22, 23, 24
F8	25, 26, 27
F9	28
F10	29
F11	30, 31, 32
F12	33, 34
F13	35
F14	36, 37
F15	38

Tabelle 3.2: Zuordnung zwischen funktionalen Anforderungen und Primäranwendungsfällen

3.3.1 Definition der Anwendungsfälle

Zu einigen der im Folgenden dargestellten Anwendungsfälle sind Beispiele angegeben, die Bezug auf mindestens eines der Szenarien nehmen. Zu allen Anwendungsfällen werden – jeweils ohne Anspruch auf Vollständigkeit – mögliche Auslöser sowie die beteiligten Organisationsdomänen aufgelistet. Sofern erforderlich, werden ergänzende Informationen gegeben, die diesen Anwendungsfall vertiefend charakterisieren.

USE CASE 1: CREATE NEW SERVICE Der Provider erweitert sein Dienstportfolio um einen neuen Dienst.

Beispiel/Szenario 2: Das LRZ nimmt einen neuen Dienst „Video Conferencing“ in sein Dienstportfolio auf. Funktionalität, Einsatzbereiche, Einschränkungen und Voraussetzungen zur Nutzung des Dienstes werden dokumentiert. Der Dienst steht den Kunden zu diesem Zeitpunkt noch nicht zur Verfügung,

die technischen Voraussetzungen hierfür sind aber prinzipiell erfüllt. „Video Conferencing“ kann von Kunden des LRZ erst beauftragt und von Anwendern genutzt werden, nachdem der Dienst auch in die jeweiligen Dienstkataloge aufgenommen wurde.

- Möglicher Auslöser: Nachfrage durch einen Kunden
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Das Dienstportfolio wird als Verzeichnis aller Dienste vom Provider geführt und kann sowohl Informationen enthalten, die nur vom Provider selbst benötigt werden (zum Beispiel technische Dienstspezifikationen, Historie eines Dienstes, Referenz auf die Konfigurationsinformationen aus einer Konfigurationsmanagementdatenbank) als auch solche, die zusätzlich für den Kunden relevant sein können (wie zum Beispiel nicht-technische Beschreibungen, messbare Qualitätsparameter und Messmethoden sowie mögliche Service-Levels). Kunden können auf die für sie sichtbar gemachten Informationen über Dienstkataloge zugreifen, die Teilsichten auf das Portfolio realisieren.

USE CASE 2: CHANGE SERVICE Der Provider ändert die Spezifikation eines Dienstes aus seinem Portfolio.

Beispiel/Szenario 1: Das Informationssystem OFFICE wird um eine Software zur Präsentationserstellung ergänzt. Die im Portfolio enthaltenen Beschreibungen und Spezifikationen des Dienstes werden im Hinblick auf diese Funktionserweiterung angepasst.

- Mögliche Auslöser:
 - Geänderte Kundenanforderungen an die Funktionalität eines Dienstes
 - Geänderte Provider-seitige Randbedingungen
 - Hersteller-seitige Änderungen an einem gelieferten Produkt oder Subdienst
- Berührte Organisationsdomäne: Provider

- Ergänzende Informationen: Sofern der Dienst von Kunden beauftragt ist, werden diese über die Änderung informiert. Im Falle von Spezifikationsänderungen (insbesondere bei Einschränkungen der Funktionalität eines Dienstes), die im Widerspruch zu Beschreibungen aus aktiven Dienstvereinbarungen stehen, müssen zuvor bzw. zeitgleich die entsprechenden SLAs unter Einbeziehung der Kunden angepasst werden.

USE CASE 3: DELETE SERVICE Der Provider entfernt einen Dienst aus seinem Portfolio.

Beispiel/Szenario 2: Der Dienst „Video Conferencing“ wird nicht mehr bereitgestellt und somit aus dem Dienstportfolio gelöscht.

- Mögliche Auslöser:
 - Geänderte Kundenanforderungen (Dienst nicht mehr benötigt)
 - Dienst technisch nicht mehr realisierbar oder steht im Widerspruch zu Qualitäts-/Sicherheitsrichtlinien des Providers
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Der Dienst darf zum Zeitpunkt der Außerbetriebnahme in keinem Dienstkatalog mehr enthalten und von keinem Kunden mehr beauftragt sein. Entsprechend sind bestehende Dienstkataloge und -vereinbarungen zuvor zu prüfen und gegebenenfalls anzupassen.

USE CASE 4: CREATE NEW CUSTOMER Der Provider fügt einen neuen Kunden zu seinem Kundenstamm hinzu.

Beispiel/Szenario 3: Eine neue Fluggesellschaft erhält An- und Abflugrechte auf dem Flughafen München. Entsprechend wird sie künftig neben den relevanten Business Services auch IT-Dienste in Anspruch nehmen und somit als neuer Dienstnehmer gegenüber dem IT-Provider auftreten. Dieser legt einen entsprechenden Kundendatensatz als Basis für Dienstkataloge für und Dienstvereinbarungen mit dieser Fluggesellschaft an.

- Möglicher Auslöser: Neuer Kunde
- Berührte Organisationsdomänen: Provider, Kunde

USE CASE 5: CHANGE CUSTOMER DATA Die zu einem Kunden gespeicherten Daten werden geändert.

(Ohne Beispiel.)

- Möglicher Auslöser: Geänderte SLM-relevante Kundeninformationen (z.B. Standortinformationen, Zahl der Nutzer, andere organisatorische oder technische Gegebenheiten)
- Berührte Organisationsdomänen: Provider, Kunde

USE CASE 6: DELETE CUSTOMER Ein Kunde wird aus dem Kundenstamm gelöscht.

(Ohne Beispiel.)

- Möglicher Auslöser: Wegfall eines Kunden
- Berührte Organisationsdomänen: Provider, Kunde

USE CASE 7: INITIALIZE SERVICE CATALOG Der Provider definiert einen Dienstkatalog als spezifische Sicht auf das Dienstportfolio.

Beispiel/Szenario 2: Einem Kunden des LRZ soll ein Dienstkatalog zur Verfügung gestellt werden, der alle Dienste enthält, die von ihm beauftragt werden können (z.B. drahtloser Netzzugang für die Lehrstuhlmitarbeiter, Hosting der Lehrstuhl-Webseiten, Terminal-Services, usw.). Zu diesem Zweck wird ein Dienstkatalog erstellt, der relevante Randbedingungen und Kundeninformationen berücksichtigt. So können beispielsweise bedingt durch den Standort des Kunden bestimmte Dienste oder Dienstmerkmale ausgeschlossen sein.

- Möglicher Auslöser: Neuer Kunde wurde angelegt (vgl. Anwendungsfall 4)
- Berührte Organisationsdomänen: Provider
- Ergänzende Informationen: Es muss ein Dienstportfolio existieren, das die Grundlage für jeden Dienstkatalog bildet. Werden Dienste durch einen Kunden beauftragt, so wird diese Beauftragung immer auf den Dienstkatalog bezogen. Der Geltungsbereich (Adressaten) jedes Dienstkatalogs (gültig für welche Kunden?) sowie sein Betrachtungsbereich

(welcher Teil des Portfolios?) müssen festgelegt werden, damit der Dienstkatalog für diesen Zweck eingesetzt werden kann. Diese beiden Aspekte werden durch die nachgelagerten Anwendungsfälle 8 und 11 abgedeckt.

USE CASE 8: ASSIGN CATALOG TO CUSTOMER Ein Dienstkatalog wird einem Kunden zugewiesen und hat somit für diesen Kunden Gültigkeit.

Beispiel/Szenario 2: Ein durch das LRZ initialisierter Dienstkatalog wird dem Kunden „Institut für Informatik“ zugewiesen.

- Möglicher Auslöser: Initialisierung eines Dienstkatalogs (vgl. Anwendungsfall 7)
- Berührte Organisationsdomänen: Provider, Kunde (nur passiv/informiert)
- Ergänzende Informationen: Ein Dienstkatalog kann einem oder mehreren Kunden zugewiesen werden. Er besitzt im zweiten Fall für jeden zugewiesenen Kunden Gültigkeit.

USE CASE 9: DISSOCIATE CATALOG FROM CUSTOMER Eine Zuweisung zwischen einem Dienstkatalog und einem Kunden wird aufgehoben.

(Ohne Beispiel.)

- Mögliche Auslöser:
 - Wegfall eines Kunden
 - Wegfall von Diensten
 - Konsolidierung von Dienstkatalogen
- Berührte Organisationsdomänen: Provider, Kunde (nur passiv/informiert)

USE CASE 10: DELETE SERVICE CATALOG Ein bestehender Dienstkatalog wird gelöscht.

(Ohne Beispiel)

- Möglicher Auslöser: Wegfall eines Kunden
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Es darf keine Dienstvereinbarungen geben, die auf Dienste aus dem zu löschenden Dienstkatalog referenzieren. Andernfalls müssen diese zuvor aufgehoben werden (siehe Anwendungsfall 21).

USE CASE 11: ADD SERVICE TO CATALOG Ein Dienst aus dem Portfolio wird in einen oder mehrere Dienstkataloge aufgenommen.

Beispiel/Szenario 2: Der Dienst „Video Conferencing“ aus dem Portfolio wird in den Dienstkatalog aufgenommen, der unter anderem für den Kunden „Institut für Informatik“ Gültigkeit hat, wobei die anwendbaren Service-Levels in dienstspezifischer Ausprägung definiert werden. Die dienstspezifische Adaption der Support-Level-Profile kann aus Abweichungen von den und Verfeinerungen der generellen Profile sowie zusätzlichen dienstspezifischen Qualitätsmerkmalen bestehen. Zudem werden die Kosten und/oder Randbedingungen für die jeweiligen Service-Levels festgelegt.

- Mögliche Auslöser:
 - Initialisierung eines Dienstkatalogs
 - Neuer Dienst verfügbar
- Berührte Organisationsdomänen: Provider, Kunde (nur passiv/informiert)
- Ergänzende Informationen: Der Dienst muss zum Dienstportfolio gehören, bevor er in einen Dienstkatalog überführt werden kann.

USE CASE 12: REMOVE SERVICE FROM CATALOG Ein Dienst wird aus einem Dienstkatalog entfernt.

(Ohne Beispiel.)

- Mögliche Auslöser:
 - Keine Nachfrage nach dem Dienst

- Dienst wurde substituiert
- Dienst nicht (mehr) realisierbar
- Berührte Organisationsdomänen: Provider, Kunde (nur passiv/informiert)
- Ergänzende Informationen: Ein Dienst kann nur aus dem Dienstkatalog entfernt werden, wenn aus aktiven Dienstvereinbarungen nicht mehr auf ihn referenziert wird.

USE CASE 13: CREATE SUPPORT LEVEL PROFILE Ein allgemeines, dienstunabhängiges Support-Level-Profil wird definiert.

Beispiel/Szenario 2: Das LRZ definiert die generellen Support-Level-Profile „Gold“, „Silber“ und „Bronze“, die sich im Wesentlichen in der maximalen Wiederherstellungszeit im Fehlerfall, den Support- und Reaktionszeiten sowie den damit verbundenen Kosten unterscheiden.

- Möglicher Auslöser: Initialisierung des Dienstportfolios
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Die durch diesen Anwendungsfall definierten Support-Level-Profile können auf Dienstkatalog-, Dienst- oder Dienstvereinbarungsebene angepasst oder ersetzt werden.

USE CASE 14: CHANGE SUPPORT LEVEL PROFILE Der Provider nimmt Änderungen an einem bestehenden dienstunabhängigen Support-Level-Profil.

Beispiel/Szenario 2: Die Supportzeiten im Support-Level-Profil „Gold“ werden auf das Wochenende ausgeweitet.

- Möglicher Auslöser: Geänderte Anforderungen an den Support und die Ausgestaltung der verschiedenen Support-Levels
- Berührte Organisationsdomänen: Provider, Kunde (nur passiv/informiert)
- Ergänzende Informationen: Kunden müssen informiert werden. Im Falle von aktiven Dienstvereinbarungen über das betreffende Support-Level müssen zeitgleich die SLAs unter Zustimmung des Kunden angepasst werden (siehe Anwendungsfall 20).

USE CASE 15: DELETE SUPPORT LEVEL PROFILE Der Provider löscht ein bestehendes dienstunabhängiges Support-Level-Profil.

Beispiel/Szenario 2: Das Support-Level-Profil „Platin“ wird gelöscht und steht nicht mehr zur Verfügung.

- Möglicher Auslöser: Geänderte Randbedingungen für den Support
- Berührte Organisationsdomänen: Provider, Kunde (nur passiv/informiert)
- Ergänzende Informationen: Kunden müssen informiert werden. Im Falle von aktiven Dienstvereinbarungen über das betreffende, zu löschende Support-Level, müssen zeitgleich die SLAs unter Zustimmung des Kunden angepasst werden (siehe Anwendungsfall 20).

USE CASE 16: CREATE DELIVERY PARTY Der Provider fügt eine interne IT-Fachabteilung oder einen externen Lieferanten/Sub-Provider zur Liste der an der Realisierung von Diensten beteiligten Parteien hinzu.

Beispiel/Szenario 3: Ein externer Telekommunikationsdienstleister wird als Lieferant gepflegt.

- Mögliche Auslöser:
 - Initialisierung der *Service Delivery*-Parteien
 - Umstrukturierung in der IT-Organisation
 - Neuer externer Lieferant, von dem (Sub-)Dienste bezogen werden (sollen)
- Berührte Organisationsdomänen: Provider, ggf. Lieferant

USE CASE 17: CHANGE DELIVERY PARTY DATA Die zu einer internen IT-Fachabteilung oder zu einem externen Lieferanten gespeicherten Daten werden geändert.

(Ohne Beispiel.)

- Mögliche Auslöser:

- Änderung von Kontaktdaten
 - Geographische Veränderungen
 - Organisatorische Umstrukturierung
- Berührte Organisationsdomänen: Provider, ggf. Lieferant

USE CASE 18: DELETE DELIVERY PARTY Der zu einer internen IT-Fachabteilung oder zu einem externen Lieferanten bestehende Datensatz wird gelöscht.

(Ohne Beispiel.)

- Mögliche Auslöser:
 - Wegfall einer IT-Fachabteilung oder eines Lieferanten
 - Organisatorische Umstrukturierung
- Berührte Organisationsdomänen: Provider, ggf. Lieferant

USE CASE 19: CLOSE SLA Ein Kunde beauftragt einen Dienst aus einem für ihn gültigen Dienstkatalog. Eine entsprechende Dienstvereinbarung wird zwischen Kunde und Provider geschlossen.

Beispiel/Szenario 1: Der Dienst „Webhosting“ wird von einem Kunden des LRZ, beispielsweise einem Lehrstuhl der Technischen Universität, beauftragt. Der Kunde wählt neben dem Dienst und möglichen Varianten auch das gewünschte Service-Level aus.

- Möglicher Auslöser: Kundenanforderung an den Dienst
- Berührte Organisationsdomänen: Provider, Kunde
- Ergänzende Informationen: Jede Dienst-Beauftragung impliziert die Vereinbarung einer für diesen Dienst gültigen Dienstvereinbarung (vgl. Grundannahme 1).

USE CASE 20: REVISE SLA Eine bestehende Dienstvereinbarung wird geändert.

Beispiel/Szenario 1: Die Verfügbarkeitsanforderungen an den beauftragten Dienst „Webhosting“ sollen auf Wunsch des Kunden neu verhandelt werden. Entsprechende in der Dienstvereinbarung festgelegte Service-Level-Ziele müssen in geänderter Version neu vereinbart werden.

- Mögliche Auslöser:
 - Geänderte Kundenanforderungen an die Dienstleistung
 - Geänderte Provider-seitige Randbedingungen der Dienstleistung
- Berührte Organisationsdomänen: Provider, Kunde
- Ergänzende Informationen: Nicht zwangsläufig muss, wie im genannten Beispiel, der Kunde den Ausschlag für diesen Anwendungsfall geben. Auch seitens des Providers kann die Notwendigkeit entstehen, eine bestehende Dienstvereinbarung nach Rücksprache mit dem Kunden zu ändern.

USE CASE 21: RESIGN SLA Eine Dienstvereinbarung wird aufgehoben und bei Bedarf archiviert.

Beispiel/Szenario 1: Der Service „Webhosting“ wird vom Kunden gekündigt.

- Möglicher Auslöser:
 - Ein Dienst wird vom Kunden zur Unterstützung seiner Geschäftsprozesse nicht mehr benötigt.
 - Ein Dienst kann nicht mehr erbracht werden.
- Berührte Organisationsdomänen: Provider, Kunde
- Ergänzende Informationen: Eventuell werden bestimmte OLAs oder UCs nach Aufhebung eines SLAs nicht mehr benötigt.

USE CASE 22: CLOSE OLA Eine operative Vereinbarung über die Erbringung eines Subdienstes wird zwischen dem Provider und einer internen IT-Fachabteilung geschlossen.

Beispiel/Szenario 2: Um die einem Kunden für einen Dienst zugesicherten Service-Level-Ziele einzuhalten, werden operative Vereinbarungen mit der den Dienst bereitstellenden IT-Abteilung sowie der Support-Abteilung (Service Desk) abgeschlossen, die die Anforderungen aus den Dienstvereinbarungen abbilden.

- Möglicher Auslöser: Eine Dienstvereinbarung soll abgeschlossen werden.
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Operative Vereinbarungen sollten zur Absicherung von in Dienstvereinbarungen garantierten Service-Levels geschlossen werden.

USE CASE 23: REVISE OLA Eine operative Vereinbarung wird geändert.

(Ohne Beispiel.)

- Mögliche Auslöser:
 - Eine Dienstvereinbarung, die durch eine operative Vereinbarung untermauert wird, soll geändert werden.
 - Eine bestehende operative Vereinbarung erweist sich als nicht ausreichend.
 - Ein Konflikt wurde festgestellt (vgl. Anwendungsfall 28).
- Berührte Organisationsdomäne: Provider

USE CASE 24: RESIGN OLA Eine operative Vereinbarung wird aufgehoben.

(Ohne Beispiel.)

- Möglicher Auslöser: Alle Dienstvereinbarungen, die auf den in der betroffenen operativen Vereinbarung zugesicherten Leistungen aufbauen, wurden aufgehoben.
- Berührte Organisationsdomäne: Provider

USE CASE 25: CLOSE UC Ein Lieferantenvertrag über die Erbringung eines Subdienstes wird zwischen dem Provider und einem externen Lieferanten geschlossen.

Beispiel/Szenario 1: Um die für einen Dienst an einen Kunden zugesicherten Service-Levels einzuhalten, wird ein Vertrag (UC) mit einem externen Lieferanten abgeschlossen, der Subdienste oder Teilsysteme für diesen Dienst liefert.

- Mögliche Auslöser:
 - Eine neue Dienstvereinbarung soll abgeschlossen werden.
 - Eine bestehende Dienstvereinbarung soll geändert werden.
- Berührte Organisationsdomänen: Provider, Lieferant
- Ergänzende Informationen: UCs sollten zur Absicherung von in SLAs garantierten Service-Levels geschlossen werden.

USE CASE 26: REVISE UC Ein Lieferantenvertrag (UC) wird geändert.

(Ohne Beispiel.)

- Mögliche Auslöser:
 - Eine Dienstvereinbarung, die durch einen Lieferantenvertrag untermauert wird, soll geändert werden.
 - Ein bestehender Lieferantenvertrag erweist sich als nicht ausreichend.
 - Ein Konflikt wurde festgestellt (vgl. Anwendungsfall 28).
- Berührte Organisationsdomänen: Provider, Lieferant

USE CASE 27: RESIGN UC Ein Lieferantenvertrag (UC) wird aufgehoben.

(Ohne Beispiel.)

- Möglicher Auslöser: Alle SLAs, die auf den im betroffenen Lieferantenvertrag zugesicherten Leistungen aufbauen, wurden aufgehoben.

- Berührte Organisationsdomänen: Provider, Lieferant

USE CASE 28: ANALYZE AGREEMENTS Der Provider analysiert Abhängigkeiten zwischen Dienstvereinbarungen, operativen Vereinbarungen und Lieferantenverträgen.

Beispiel/Szenario 1: Nach der Änderung eines SLAs mit einem Kunden überprüft der Provider, ob die bestehenden OLAs und UCs ausreichen, um die Einhaltung der Service-Level-Ziele garantieren zu können.

- Mögliche Auslöser: Ein SLA, OLA oder UC wurde erstellt, geändert oder aufgehoben.
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Dienstvereinbarungen, operative Vereinbarungen und Lieferantenvereinbarungen stehen in engem Zusammenhang. Insbesondere sollen OLAs und UCs zur Realisierung der Service-Level-Ziele aus den SLAs beitragen, indem sie die beteiligten Parteien zur Schaffung der erforderlichen Randbedingungen zur zielgerechten Dienstleistung verpflichten. Konflikte oder Unzulänglichkeiten in den komplexen Zusammenhängen zwischen OLAs, UCs und SLAs – gegebenenfalls auch über mehrere Stufen – sollen mit Hilfe eines SLM-unterstützten Managementsystems erkannt werden können. Das Ergebnis der Überprüfung kann zu Änderungen an bestehenden SLAs, OLAs und UCs führen.

USE CASE 29: MEASURE AND RECORD Die in den Vereinbarungen festgelegten Leistungsparameter werden gemessen, mit den Zielwerten verglichen und entsprechende Datensätze und Aufzeichnungen generiert.

(Ohne Beispiel.)

- Möglicher Auslöser: Die mit diesem Anwendungsfall verbundenen Aktivitäten werden im Normalfall nicht ereignisgesteuert ausgelöst, sondern finden während der gesamten Betriebsphase jedes Dienstes in der in den SLAs definierten und vereinbarten Art und Weise statt, also zum Beispiel permanent, stichprobenartig oder in bestimmten Zeitintervallen.
- Berührte Organisationsdomäne: Provider

USE CASE 30: CREATE INTERNAL REPORT Auf Basis der Mess-Datensätze und Aufzeichnungen werden interne Berichte generiert.

(Ohne Beispiel.)

- Möglicher Auslöser: Berichte werden im Normalfall nicht ereignisgesteuert, sondern periodisch erstellt, zum Beispiel monatlich, quartalsweise oder jährlich.
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Interne Berichte adressieren primär die Entscheider in und „über“ den IT-Fachabteilungen wie zum Beispiel Fachgruppenleiter, Bereichsleiter oder den *Chief Information Officer* (CIO). Von entscheidender Wichtigkeit ist, dass die Berichte zur operativen Verbesserung der Dienste und der Dienstmanagementprozesse einen zielführenden Beitrag leisten und an ein entsprechendes Verbesserungsprogramm gekoppelt sind, wie es unter anderem in ISO/IEC 20000 gefordert wird (*Service Improvement Plan/Program*).

USE CASE 31: CREATE EXTERNAL REPORT Auf Basis der Mess-Datensätze und Aufzeichnungen werden Berichte für einen Kunden generiert.

(Ohne Beispiel.)

- Möglicher Auslöser: wie in Anwendungsfall 30
- Berührte Organisationsdomänen: Provider, Kunde

USE CASE 32: CREATE SLA VIOLATION NOTIFICATION Auf Basis von Mess-Datensätzen und Aufzeichnungen im Abgleich mit bestehenden Dienstvereinbarungen und den enthaltenen Service-Level-Zielen wird festgestellt, dass ein SLA nicht eingehalten wird oder in naher Zukunft nicht eingehalten werden kann. Eine Benachrichtigung des betreffenden Kunden wird generiert.

(Ohne Beispiel.)

- Möglicher Auslöser: Tatsächliche oder abzusehende Verletzung einer Dienstvereinbarung mit maßgeblicher Auswirkung für den betroffenen Kunden

- Berührte Organisationsdomänen: Provider, Kunde

USE CASE 33: CALCULATE SLA-BASED INCIDENT URGENCY Die Dringlichkeit einer Störung eines IT-Dienstes wird als Basis für die Festsetzung der Priorität auf der Grundlage der in den Dienstvereinbarungen über den gestörten Dienst vereinbarten Service-Level-Ziele und der assoziierten Support-Levels ermittelt.

(Ohne Beispiel.)

- Möglicher Auslöser: Eine Störung wurde von einem Anwender gemeldet oder durch ein Überwachungssystem festgestellt. Zur Ablaufkoordination der Entstörung soll SLA-basiert eine Priorität festgesetzt werden.
- Berührte Organisationsdomäne: Provider

USE CASE 34: DETERMINE RESOLUTION CONSTRAINTS Die für die Wiederherstellung eines gestörten IT-Dienstes relevanten Randbedingungen werden auf Basis von Dienstvereinbarungen und assoziierten Support-Levels ermittelt und beschrieben. Hierunter fallen vor allem Reaktionszeiten und maximale Wiederherstellungszeiten.

(Ohne Beispiel.)

- Möglicher Auslöser: Die Lösung einer Störung und Wiederherstellung des gestörten Dienstes soll geplant und implementiert werden.
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Während das Ziel des Anwendungsfalls 32 die Ermittlung eines diskreten numerischen Wertes ist, der die relative Dringlichkeit einer Störung zum Ausdruck bringt (Quantifizierung), zielt dieser Anwendungsfall darauf ab, die genannten Randbedingungen qualitativ und quantitativ darzustellen. Beide Anwendungsfälle können dabei auf den teilweise identischen Informationen beruhen.

USE CASE 35: CALCULATE SLA-BASED COSTS FOR SERVICE USAGE Die für den Betrieb eines Dienstes und seine Bereitstellung an einen Kunden zu verrechnenden Kosten werden auf Basis der in der betreffenden Dienstvereinbarung festgelegten Abrechnungsinformationen (Preise, Abrechnungseinheiten, etc.) sowie des Nutzungsverhaltens des Kunden berechnet.

(Ohne Beispiel.)

- Mögliche Auslöser:
 - Ende einer Abrechnungsperiode erreicht
 - Anfrage einer (Zwischen-)Abrechnung
 - Aufhebung einer Dienstvereinbarung
- Berührte Organisationsdomänen: Provider, Kunde

USE CASE 36: CALCULATE SLA-BASED CHANGE CATEGORY Die Kategorie einer geplanten Änderung an einem Dienst oder einer seiner Komponenten wird auf der Grundlage der bestehenden Dienstvereinbarungen zu diesem Dienst ermittelt.

(Ohne Beispiel.)

- Möglicher Auslöser: Ein Änderungsantrag (*Request for Change*) soll im Hinblick auf die Kategorie der geplanten Änderung bewertet werden.
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Änderungen an IT-Diensten und ihrer technischen Infrastruktur durchlaufen in der Regel einen formalen Prozess zum Änderungsmanagement, der zur Bewertung und Kategorisierung einer (geplanten) Änderung auf Informationen aus Dienstvereinbarungen angewiesen ist. Die Kategorie einer Änderung ergibt sich aus ihrer Priorität und ihrem Risiko. Zur Bestimmung beider Komponenten müssen SLAs hinreichend aussagekräftige Informationen beinhalten.

USE CASE 37: DETERMINE TRANSITION CONSTRAINTS Die transitionsrelevanten Randbedingungen der Bereitstellung eines neuen oder geänderten IT-Dienstes werden auf Basis bestehender Dienstvereinbarungen bestimmt.

(Ohne Beispiel.)

- Möglicher Auslöser: Release-Planung für einen Dienst
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: Bei der Bereitstellung eines Dienstes müssen vielfältige Randbedingungen Berücksichtigung finden, von denen sich einige aus Dienstvereinbarungen ergeben können bzw. für die Dienstvereinbarungen sogar die Grundlage bilden. Ein Beispiel sind festgelegte

Wartungsfenster, die zur Ausrollung von Diensten vorgesehen sind, um eine möglichst geringe Beeinträchtigung der Geschäftsprozesse zu erzielen.

USE CASE 38: DETERMINE SERVICE SECURITY CONSTRAINTS Die sicherheitsrelevanten Randbedingungen der Bereitstellung eines neuen oder geänderten IT-Dienstes werden auf Basis bestehender Dienstvereinbarungen bestimmt.

(Ohne Beispiel.)

- Möglicher Auslöser: Planung der informationssicherheitstechnischen Dienstkonfiguration
- Berührte Organisationsdomäne: Provider
- Ergänzende Informationen: vgl. Anwendungsfall 36; Beispiele für sicherheitsrelevante Randbedingungen sind die mit dem Kunden vereinbarten Anforderungen an Authentifizierung oder Datenintegrität.

3.3.2 Zusammenhänge zwischen den Anwendungsfällen

Die Abbildungen 3.12 und 3.13 illustrieren überblicksartig die wichtigsten Abhängigkeiten zwischen den identifizierten Anwendungsfällen. Jeder gestrichelte, gerichtete Pfeil steht für eine Abhängigkeit eines Anwendungsfalls von einem anderen, wobei folgende Farbsemantik Anwendung findet:

- Grauer Pfeil: Triviale Abhängigkeit zwischen eng verwandten Anwendungsfällen; häufig: Änderung oder Löschung eines Objekts hängt ab von dessen vorheriger Erstellung.
- Schwarzer Pfeil: Nicht-triviale Abhängigkeit; häufig: Erstellung, Änderung oder Löschung eines Objekts hängt ab von vorheriger Erstellung Änderung oder Löschung *eines anderen* Objekts.

Tabelle 3.3 beschreibt und begründet jede der dargestellten Abhängigkeiten abschließend.

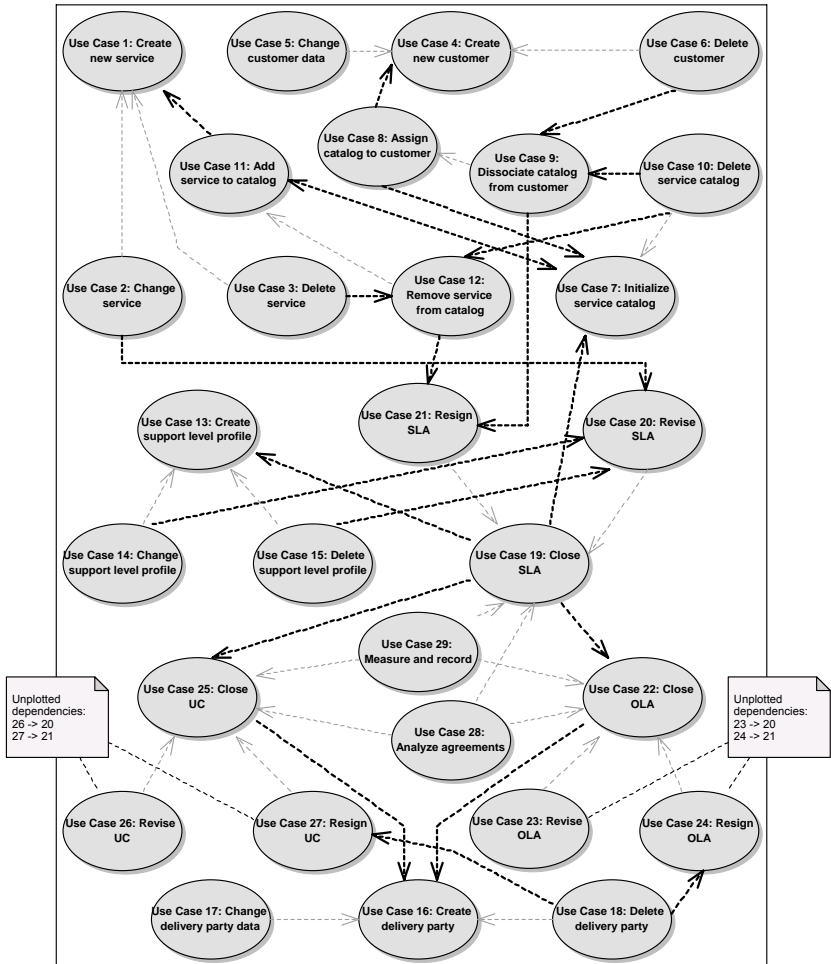


Abbildung 3.12: Überblick über die Primäranwendungsfälle 1 bis 29 und ihre wechselseitigen Abhängigkeiten

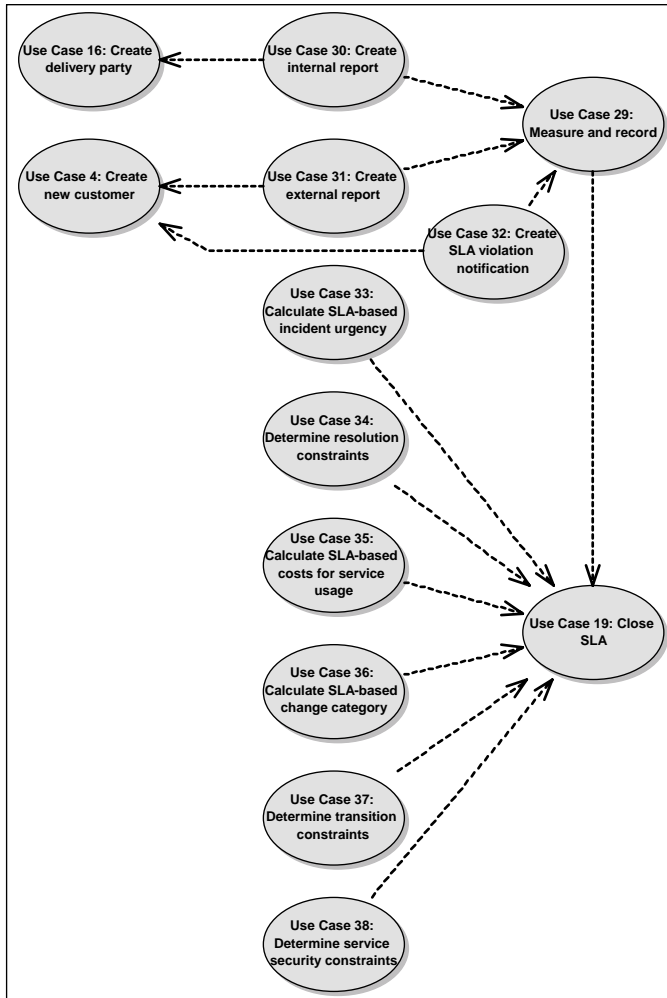


Abbildung 3.13: Überblick über die Primäranwendungsfälle 30 bis 38 und ihre wechselseitigen Abhängigkeiten

<i>Von Use Case</i>	<i>Nach Use Case</i>	<i>Erklärung</i>
2 (<i>Change service</i>)	1 (<i>Create new service</i>)	Um einen Dienst zu ändern, muss dieser zuvor erstellt worden sein.
2 (<i>Change service</i>)	20 (<i>Revise SLA</i>)	Wird ein Dienst verändert, so müssen SLAs, die sich auf diesen Dienst beziehen, gegebenenfalls angepasst werden.
3 (<i>Delete service</i>)	1 (<i>Create new service</i>)	Um einen Dienst zu löschen, muss dieser zuvor erstellt worden sein.
3 (<i>Delete service</i>)	12 (<i>Remove service from catalog</i>)	Bevor ein Dienst stillgelegt und aus dem Portfolio entfernt werden kann, muss er aus allen Dienstkatalogen entfernt werden.
5 (<i>Change customer data</i>)	4 (<i>Create new customer</i>)	Bevor ein Kundendatensatz geändert werden kann, muss dieser angelegt worden sein.
6 (<i>Delete customer</i>)	4 (<i>Create new customer</i>)	Bevor ein Kundendatensatz gelöscht werden kann, muss dieser angelegt worden sein.
6 (<i>Delete customer</i>)	9 (<i>Dissociate catalog from customer</i>)	Bevor ein Kundendatensatz gelöscht werden kann, muss sichergestellt sein, dass es keine Zuweisungen zwischen diesem Kunden und bestehenden Dienstkatalogen mehr gibt. Falls doch, müssen entsprechende Zuweisungen zuerst aufgehoben werden.
8 (<i>Assign catalog to customer</i>)	4 (<i>Create new customer</i>)	Damit ein Dienstkatalog einem Kunden zugewiesen werden kann, muss der Kunde zuvor angelegt worden sein.

8 (<i>Assign catalog to customer</i>)	7 (<i>Initialize service catalog</i>)	Um einen Kunden festzulegen, für den ein Dienstkatalog gültig ist, muss der entsprechende Katalog zuvor erstellt worden sein.
9 (<i>Dissociate catalog from customer</i>)	8 (<i>Assign catalog to customer</i>)	Nur eine bestehende Zuweisung zwischen einem Dienstkatalog und einem Kunden kann auch wieder aufgehoben werden.
9 (<i>Dissociate catalog from customer</i>)	21 (<i>Resign SLA</i>)	Soll ein Dienstkatalog nicht länger Gültigkeit für einen Kunden besitzen, so dürfen auch keine SLAs mehr bestehen, die Dienste aus diesem Katalog referenzieren. Somit müssen vor der Aufhebung einer Zuweisung zwischen Dienstkatalog und Kunde alle bestehenden SLAs aufgehoben werden.
10 (<i>Delete service catalog</i>)	7 (<i>Initialize service catalog</i>)	Nur ein bestehender Dienstkatalog kann auch gelöscht werden.
10 (<i>Delete service catalog</i>)	9 (<i>Dissociate catalog from customer</i>)	Ein Dienstkatalog kann nur gelöscht werden, wenn er für keinen Kunden mehr gültig ist. Somit müssen zuvor alle entsprechenden Zuweisungen zwischen Kunden und diesem Dienstkatalog aufgehoben werden.
10 (<i>Delete service catalog</i>)	12 (<i>Remove service from catalog</i>)	Ein Dienstkatalog kann nur gelöscht werden, wenn er keine Dienste mehr enthält bzw. referenziert.

11 (<i>Add service to catalog</i>)	1 (<i>Create new service</i>)	Ein Dienst muss zunächst im Portfolio angelegt werden, um zu einem Dienstkatalog hinzugefügt zu werden.
11 (<i>Add service to catalog</i>)	7 (<i>Initialize service catalog</i>)	Zur Initialisierung eines Dienstkatalogs gehört das Hinzufügen einer initialen Menge an Diensten.
12 (<i>Remove service from catalog</i>)	11 (<i>Add service to catalog</i>)	Um einen Dienst aus einem Dienstkatalog zu entfernen, muss dieser zuvor hinzugefügt worden sein.
12 (<i>Remove service from catalog</i>)	21 (<i>Resign SLA</i>)	Bevor ein Dienst aus einem Dienstkatalog entfernt werden kann, müssen SLAs von Kunden, für die dieser Dienstkatalog gültig ist und die sich auf den zu löschenden Dienst beziehen, aufgelöst werden.
14 (<i>Change support level profile</i>)	13 (<i>Create support level profile</i>)	Um ein Support-Level-Profil zu ändern, muss dieses zuvor angelegt worden sein.
14 (<i>Change support level profile</i>)	20 (<i>Revise SLA</i>)	Wird ein Support-Level-Profil geändert, so müssen SLAs, die dieses Profil referenzieren, gegebenenfalls angepasst werden.
15 (<i>Delete support level profile</i>)	13 (<i>Create support level profile</i>)	Um ein Support-Level-Profil zu löschen, muss dieses zuvor angelegt worden sein.
15 (<i>Delete support level profile</i>)	20 (<i>Revise SLA</i>)	Wird ein Support-Level-Profil gelöscht, so müssen SLAs, die dieses Profil bislang referenzieren, angepasst werden.

17 (<i>Change delivery party data</i>)	16 (<i>Create delivery party</i>)	Damit die zu einer an der Dienstleistung beteiligten Partei gespeicherten Informationen verändert werden können, muss diese zuvor angelegt worden sein.
18 (<i>Change delivery party data</i>)	16 (<i>Create delivery party</i>)	Damit eine an der Dienstleistung beteiligte Partei aus der zugehörigen Informationsbasis gelöscht werden kann, muss diese zuvor angelegt worden sein.
18 (<i>Change delivery party data</i>)	24 (<i>Resign OLA</i>)	Bevor eine an der Dienstleistung beteiligte Partei aus der zugehörigen Informationsbasis gelöscht werden kann, müssen im Falle eines internen Dienstleistungsbereitstellers alle mit ihm geschlossenen OLAs aufgehoben worden sein.
18 (<i>Change delivery party data</i>)	27 (<i>Resign UC</i>)	Bevor eine an der Dienstleistung beteiligte Partei aus der zugehörigen Informationsbasis gelöscht werden kann, müssen im Falle eines externen Lieferanten alle mit ihm geschlossenen UCs aufgehoben worden sein.
19 (<i>Close SLA</i>)	4 (<i>Create new customer</i>)	Jedes SLA wird vom Provider mit einem bestehenden Kunden geschlossen.
19 (<i>Close SLA</i>)	7 (<i>Initialize service catalog</i>)	Jedes SLA zwischen Kunde und Provider bezieht sich auf einen im gültigen Dienstkatalog enthaltenen Dienst; entsprechend ist ein Dienstkatalog Voraussetzung für den Abschluss von SLAs.

19 (<i>Close SLA</i>)	13 (<i>Create support level profile</i>)	Jedes SLA referenziert ein Support-Level-Profil, das zuvor angelegt worden sein muss.
19 (<i>Close SLA</i>)	22 (<i>Close OLA</i>)	Ein SLA über einen Dienst muss durch entsprechende operative Vereinbarungen untermauert werden, um die Service-Level-Ziele auf ein belastbares Fundament zu stellen.
19 (<i>Close SLA</i>)	25 (<i>Close UC</i>)	Ein SLA über einen Dienst muss durch entsprechende Lieferantenverträge untermauert werden, um die Service-Level-Ziele auf ein belastbares Fundament zu stellen, sofern der Dienst ganz oder teilweise von externen Sub-Providern erbracht wird oder von extern erbrachten Subdiensten abhängt.
20 (<i>Revise SLA</i>)	19 (<i>Close SLA</i>)	Nur zuvor abgeschlossene SLAs können geändert werden.
21 (<i>Resign SLA</i>)	19 (<i>Close SLA</i>)	Nur zuvor abgeschlossene SLAs können aufgehoben werden.
22 (<i>Close OLA</i>)	16 (<i>Create delivery party</i>)	OLAs werden mit internen IT-Fachabteilungen geschlossen, die zuvor als an der Diensterbringung/-realisierung beteiligte Parteien angelegt werden müssen.
23 (<i>Revise OLA</i>)	22 (<i>Close OLA</i>)	Nur zuvor abgeschlossene operative Vereinbarungen können geändert werden.

24 (<i>Resign OLA</i>)	22 (<i>Close OLA</i>)	Nur zuvor abgeschlossene operative Vereinbarungen können aufgehoben werden.
25 (<i>Close UC</i>)	16 (<i>Create delivery party</i>)	UCs werden mit externen Zulieferern geschlossen, die zuvor als an der Dienstleistung/-realisierung beteiligte Parteien angelegt werden müssen.
26 (<i>Revise UC</i>)	25 (<i>Close UC</i>)	Nur zuvor abgeschlossene Lieferantenverträge können geändert werden.
27 (<i>Resign UC</i>)	25 (<i>Close UC</i>)	Nur zuvor abgeschlossene Lieferantenverträge können aufgehoben werden.
28 (<i>Analyze agreements</i>)	19/22/25 (<i>Close SLA/OLA/UC</i>)	SLAs, OLAs und UCs müssen existieren, bevor Zusammenhänge zwischen ihnen analysiert und Konflikte erkannt werden können.
29 (<i>Measure and record</i>)	19/22/25 (<i>Close SLA/OLA/UC</i>)	Die in den SLAs, operativen Vereinbarungen und Lieferantenvereinbarungen getroffenen Zielvereinbarungen müssen gemessen und die Ergebnisse aufgezeichnet werden.
30 (<i>Create internal report</i>)	16 (<i>Create delivery party</i>)	Interne Berichte können interne an der Dienstleistung beteiligte Parteien als Adressaten haben.
30 (<i>Create internal report</i>)	29 (<i>Measure and record</i>)	Jeder interne Bericht basiert auf zuvor erhobenen Messdaten und Aufzeichnungen.
31 (<i>Create external report</i>)	4 (<i>Create new customer</i>)	Jeder externe Bericht hat einen oder mehrere Kunden als Adressaten.
31 (<i>Create external report</i>)	29 (<i>Measure and record</i>)	Jeder externe Bericht basiert auf zuvor erhobenen Messdaten und Aufzeichnungen.

32 (<i>Create SLA violation notification</i>)	4 (<i>Create new customer</i>)	Jede Benachrichtigung über eine SLA-Verletzung hat den Kunden als Adressaten, dessen SLA verletzt wurde oder verletzt zu werden droht.
32 (<i>Create SLA violation notification</i>)	29 (<i>Measure and record</i>)	Um SLA-Verletzungen festzustellen, müssen erhobene Messdaten und Aufzeichnungen mit den Zielen aus bestehenden SLAs abgeglichen werden.
33 (<i>Calculate SLA-based incident urgency</i>)	19 (<i>Close SLA</i>)	Um die Dringlichkeit als Komponente der Priorität einer Störung auf Basis gültiger SLAs ermitteln zu können, müssen entsprechende SLAs zuvor abgeschlossen worden sein.
34 (<i>Determine resolution constraints</i>)	19 (<i>Close SLA</i>)	Zu den Entstörungsbedingungen gehören Reaktions- und Wiederherstellungszeiten sowie vereinbarte Wartungsfenster, die in entsprechenden SLAs festgehalten werden müssen.
35 (<i>Calculate SLA-based costs for service usage</i>)	19 (<i>Close SLA</i>)	Um Kosten für die Nutzung von Diensten an Kunden weiter zu verrechnen, müssen zuvor SLAs abgeschlossen worden sein, die die erforderlichen Abrechnungsinformationen enthalten.
36 (<i>Calculate SLA-based change category</i>)	19 (<i>Close SLA</i>)	Die Kategorie einer nachgefragten und/oder erforderlichen Änderung kann auf Basis in SLAs festgehaltener, das <i>Change Management</i> betreffende Informationen bestimmt werden.

37 (<i>Determine transition constraints</i>)	19 (<i>Close SLA</i>)	Transitions-Bedingungen ergeben sich analog zu den Entstörungs-Bedingungen für die Transition neuer oder geänderter Dienste in die Live-Umgebung aus in SLAs erfolgten Festlegungen.
38 (<i>Determine service security constraints</i>)	19 (<i>Close SLA</i>)	SLAs müssen bestehen und Kundenanforderungen an die Informationssicherheit widerspiegeln, aus denen die sicherheitstechnischen Randbedingungen der Dienstbringung bestimmt werden können.

Tabelle 3.3: Beschreibung der Abhängigkeiten zwischen den Anwendungsfällen

3.4 Nicht-funktionale Anforderungen

Bisher wurden in diesem Kapitel ausschließlich funktionale Anforderungen ans Service-Level-Management betrachtet, die aus den Szenarien abgeleitet und in Anwendungsfälle für ein potenzielles Managementsystem überführt wurden. Diese Anwendungsfälle beschreiben schon recht detailliert den gewünschten Funktionskatalog eines solchen Managementsystems und stellen somit zugleich die Kernanforderungen an die Managementarchitektur dar, deren Ziel es ja ist, die Entwicklung eines konkreten Managementsystems zu erleichtern und zu unterstützen.

Doch neben den funktionalen Anforderungen ergeben sich für die *SLM-Architektur* (nicht für ein *Managementsystem!*) noch einige zusätzliche nicht-funktionale Anforderungen. Sie werden im Folgenden kurz eingeführt.

NF1: ALLGEMEINGÜLTIGKEIT

Die Forderung nach Allgemeingültigkeit der Architektur bezieht sich darauf, dass beim Architekturentwurf möglichst keine (unnötig) einschränkenden Annahmen getroffen werden sollen, die die Anwendbarkeit der Architektur in bestimmten Szenarien gefährden.

Im Abschnitt 3.2.1.2 wurden drei Grundannahmen getroffen, die weniger eine Einschränkung der Allgemeingültigkeit als vielmehr eine Präzisierung bestimmter elementarer Zusammenhänge im Service-Level-Management zum Ziel haben.

Durch die Auswahl verschiedener Szenarien und die morphologische Analyse wurde außerdem sichergestellt, dass die Quellen für Anforderungen und Anwendungsfälle so gewählt sind, dass keine wesentlichen Szenario-Charakteristika für eine umfassende Architektur fehlen.

NF2: PLATTFORMUNABHÄNGIGKEIT

Die Anforderung der Plattformunabhängigkeit steht in engem Zusammenhang mit der Forderung nach Allgemeingültigkeit. Während NF1 allerdings darauf abzielt, dass die den Anforderungen und Primäranwendungsfällen zugrunde liegenden Szenarien umfassend genug gewählt sind, bezieht sich die Forderung nach Plattformunabhängigkeit auf die Teilmodelle und Modellelemente, die im Rahmen der Prozess- und Systemsicht der Architektur entwickelt werden. Es dürfen keine impliziten oder expliziten einschränkenden Annahmen über Plattformen oder Implementierungstechniken für die Implementierung des Zielsystems gemacht werden.

Zielführend hierfür ist der Einsatz anerkannter und plattformunspezifischer Modellierungswerkzeuge wie die *Unified Modeling Language* (UML) [32, 33, 34] und die *Extensible Markup Language* (XML) [35], wie sie im Rahmen der Kapitel 5 und 6 verwendet werden.

NF3: ANPASSBARKEIT/FLEXIBILITÄT

Wann immer im Rahmen der Architektur konkrete Modellelemente entwickelt und angegeben werden – beispielsweise im Kontext des Informationsmodells der System-Sicht – soll gewährleistet sein, dass diese Modelle in sinnvoller Weise verändert oder ergänzt werden können. Im Falle des Informationsmodells würde das beispielsweise die Hinzunahme weiterer Attribute zu einer UML-Klasse, die ein Informationsobjekt (Artefakt) repräsentiert, bedeuten.

Die Architektur muss daher eine Methodik zur Modifikation von Teilmodellen oder Modellelementen mit sich bringen, die vor allem die Auswirkung konkreter Änderungen auf andere Modellelemente berücksichtigt. Dies wird im Kapitel 7 adressiert.

NF4: ERWEITERBARKEIT

Die Forderung nach Erweiterbarkeit der Architektur bezieht sich vor allem auf die Möglichkeit, zusätzliche Primäranwendungsfälle (in Ergänzung zu den Anwendungsfällen aus dem funktionalen Teil des Anforderungskatalogs) für das Design eines konkreten Managementsystems berücksichtigen zu können. Vor dem Hintergrund, dass die für die funktionale Anforderungsanalyse gewählten Szenarien keine absolute Vollständigkeit garantieren können, ist der Erweiterbarkeitsaspekt besonders wichtig.

Die Architektur muss eine Methodik zur Selbst-Erweiterung auf Basis neuer Anwendungsfälle mitliefern. Eine solche Methodik wird im Kapitel 7 vorgestellt.

NF5: FEHLERFREIHEIT UND KONSISTENZ DER TEILMODELLE

Einzelne Teilmodelle sollen für sich genommen fehlerfrei und auf allen Ebenen untereinander konsistent und widerspruchsfrei sein. Das betrifft unterschiedliche Elemente eines Teilmodells, verschiedene Teilmodelle einer Sicht sowie Teilmodelle unterschiedlicher Sichten (Prozess- vs. System-Sicht). Man kann in diesem Zusammenhang syntaktische und semantische Fehlerfreiheit unterscheiden.

- Syntaktische Fehlerfreiheit und Konsistenz der Teilmodelle: Durch den Einsatz eines professionellen Modellierungswerkzeugs kann eine Vielzahl potenzieller syntaktischer Fehler vermieden werden, wie zum Beispiel die fehlerhafte oder deplatzierte Verwendung bestimmter syntaktischer Konstrukte oder Namenskonflikte zwischen verschiedenen Teilmodellen eines Gesamtsystems.
- Semantische Fehlerfreiheit und Konsistenz der Teilmodelle: Semantische Fehlerfreiheit ist generell nicht garantierbar. Den wichtigsten Beitrag zur Sicherstellung robuster Modelle mit geringer Fehleranfälligkeit leistet im Rahmen dieser Arbeit das klare methodische Vorgehen mit dem in Kapitel 2 eingeführten Ordnungsrahmen, welcher den Problembereich auf unterschiedliche Sichten und Teilmodelle herunterbricht und damit hilft, die Komplexität zu reduzieren und beherrschbar zu machen, was letztlich die Gefahr von Fehlern eindämmt.

NF6: EINHALTUNG DER GRUNDSÄTZE ORDNUNGSMÄSSIGER MODELLIERUNG

Die Grundsätze ordnungsmäßiger Modellierung (GOM) nach Becker et al. [36, 37] stellen einen generischen methodischen Ordnungsrahmen für die Erstellung betrieblicher Informationsmodelle dar. Sie ergeben in der Summe eine hilfreiche „Checkliste“, die – angewandt auf die einzelnen Architekturteilmodelle – jeden Teil der Gesamtlösung im Hinblick auf die folgenden Aspekte kritisch durchleuchtet:

- Grundsatz der Richtigkeit: Passt das Modell zur Realität bzw. erfüllt es tatsächliche/reale Anforderungen?
- Grundsatz der Relevanz: Ist das Modell für die Erfüllung der Anforderungen relevant, d.h. wird es (in der gegebenen Ausprägung) benötigt?
- Grundsatz der Wirtschaftlichkeit: Ist der Detaillierungsgrad des Modells sinnvoll gewählt?
- Grundsatz der Klarheit: Ist das Modell in seiner Ausprägung verständlich, nachvollziehbar und anschaulich?
- Grundsatz der Vergleichbarkeit: Lässt sich das Modell in der gewählten Notation mit verwandten oder in Beziehung stehenden Modellen in anderer Notation vergleichen?
- Grundsatz des systematischen Aufbaus: Wird bei der Modellierung in unterschiedlichen Sichten die sichtenübergreifende Konsistenz gewährleistet?

3.5 Zusammenfassung und nächste Schritte

Durch die identifizierten Primäranwendungsfälle sind nun konkrete Anforderungen an ein Managementsystem für den Bereich Service-Level-Management bekannt. Das Fundament für die gewünschte Managementarchitektur ist damit geschaffen.

Alle weiteren Schritte im Architekturentwurf stehen in sehr engem Zusammenhang zu diesen Primäranwendungsfällen. So wird in Kapitel 5 zu jedem Primäranwendungsfall ein Referenz-Workflow definiert, der den Kontrollfluss innerhalb des Anwendungsfalls herausstellt. Auch die Informationsartefakte werden in Kapitel 5 durch Anwendungsfall-Analyse identifiziert. Darüber hinaus liefern die Anwendungsfälle Anhaltspunkte für Kommunikationsbeziehungen innerhalb des Service-Level-Managements sowie zwischen Service-Level-Management und anderen Disziplinen des Dienstmanagements.

Kapitel 4

Umfeld und verwandte Arbeiten

Inhalt dieses Kapitels

4.1 Einführung, Überblick und Bewertungsschema	132
4.1.1 Überblick	132
4.1.2 Analyse- und Bewertungsschema	135
4.2 Grundlegende Arbeiten	135
4.2.1 Dienstmanagement-Rahmenwerk (Dreo)	135
4.2.2 SLM für Unternehmensnetze (Lewis)	139
4.3 Prozessrahmenwerke	141
4.3.1 IT Infrastructure Library (OGC)	141
4.3.2 ISO/IEC 20000	145
4.3.3 Enhanced Telecom Operations Map (TMF)	147
4.4 Modellbasierte Ansätze	148
4.4.1 Shared Information/Data Model (TMF)	148
4.4.2 Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement (Brenner)	154
4.4.3 Modellbasiertes SLM (Debusmann)	155
4.5 Sonstige Ansätze	156
4.5.1 NGOSS SLA-Management-Handbuch (TMF)	156
4.5.2 SLA-Entwurf auf Basis von Dienstprozessen (Schmidt)	162
4.5.3 Weitere Veröffentlichungen im Kurzüberblick	164
4.6 Zusammenfassung und abschließende Bewertung	165

In diesem Kapitel werden Arbeiten und Ansätze vorgestellt, die mit den Fragestellungen dieser Dissertation thematisch verwandt sind. Da die Zahl der Publikationen im Themenbereich IT-Service-Management über die letzten Jahre stark zugenommen hat, beschränkt sich dieses Kapitel im Wesentlichen auf

diejenigen Werke, die für diese Arbeit eine hohe Relevanz haben, weil sie entweder maßgeblich zur *Begriffsbildung* beitragen, einen *methodisch ähnlichen Weg* gehen oder *konzeptionell in Teile des Lösungsansatzes* einfließen.

Darüber hinaus verfolgt dieses Kapitel das Ziel einer *inhaltlichen Abgrenzung* zu anderen Arbeiten, insbesondere aus den Gebieten Dienstmanagement und Service-Level-Management. Es wird dargestellt, wie sich diese Arbeit in das Spektrum der bisherigen Veröffentlichungen zum Schwerpunktthema Service-Level-Management einordnen lässt, wo sich Schnittmengen ergeben und in welchen Fällen gegebenenfalls auch inhaltliche Konflikte oder Unverträglichkeiten erkennbar sind.

4.1 Einführung, Überblick und Bewertungsschema

Bevor sich dieses Kapitel der detaillierten Analyse anderer Arbeiten aus dem großen Umfeld *Dienstmanagement und Service-Level-Management* widmet, versucht dieser Abschnitt den Betrachtungsbereich sinnvoll zu strukturieren und erste Bezüge zur vorliegenden Arbeit herzustellen. In Abschnitt 4.1.1 wird zu diesem Zweck eine tabellarische Übersicht über alle betrachteten Ansätze entwickelt. Abschnitt 4.1.2 präsentiert im Anschluss daran ein Analyse- und Bewertungsschema, das es ermöglicht, die zuvor nur grob klassifizierten Ansätze noch detaillierter im Kontext von Zielsetzung und Vorgehensweise dieser Arbeit zu analysieren.

4.1.1 Überblick

Tabelle 4.1 gibt einen ersten Überblick über die im Rahmen dieses Kapitels adressierten Arbeiten und Ansätze, die in vier Kategorien eingeteilt werden: *Grundlegende Arbeiten* zum Thema Service-Level-Management, *Prozessrahmenwerke*, *Modellbasierte Ansätze* und *Sonstige Arbeiten*. Bei den grundlegenden Arbeiten handelt es sich um Bücher oder Forschungsarbeiten, die sich dem Thema Service-Level-Management aus einer unbeeinflussten Perspektive nähern und dabei nicht zur zweiten Kategorie, den Prozessrahmenwerken, zu zählen sind.

Die Prozessrahmenwerke haben gemeinsam, dass sie Prozesse für das Management von IT-Diensten spezifizieren. Die Disziplin bzw. der Prozess Service-Level-Management nimmt in allen diesen Rahmenwerken eine zentrale Stellung ein. Der Fokus der unterschiedlichen Rahmenwerke und auch die Art der Darstellung der Prozesse variiert zum Teil sehr stark, weshalb es als wichtig

anzusehen ist, sich gerade bei der wissenschaftlichen Auseinandersetzung nicht nur auf ein bestimmtes Rahmenwerk zu beschränken.

Die modellbasierten Ansätze verwenden jeweils einen bestimmten Modellierungsansatz, um sich mit ausgewählten Aspekten des Dienst- oder Service-Level-Managements zu befassen. Auch die in dieser Kategorie vorgestellten Arbeiten unterscheiden sich zum Teil stark in Zielsetzung und Herangehensweise. Arbeiten, die nicht oder nicht eindeutig einer der drei qualifizierten Kategorien zugeordnet werden können, die aber für die vorliegende Arbeit relevant sind, werden im letzten Teilkapitel vorgestellt und beschließen die Umfeld-Analyse.

Tabelle 4.1 zeigt, dass jeder der im Folgenden vorgestellten Ansätze in mindestens einem der zu Beginn genannten Punkte für diese Arbeit relevant ist. Zur Begriffsbildung, die im Rahmen dieser Arbeit zu einem initialen Teil bereits in Kapitel 2.3 erfolgte, wurden die Veröffentlichungen herangezogen, die aktuell eine besonders hohe Verbreitung in der Praxis erreicht haben – wie im Fall der *IT Infrastructure Library* (ITIL) –, normativen Charakter haben – wie im Fall des Internationalen Standards für IT-Service-Management ISO/IEC 20000 –, oder die in der Forschungslandschaft wichtige theoretische Grundsteine gelegt haben – wie das Dienstmanagement-Rahmenwerk von Dreo für den Bereich des Dienstmanagements allgemein oder das Buch von Lewis für das Themengebiet Service-Level-Management.

Methodisch geht diese Arbeit mit dem in Kapitel 2 gewählten Konzept zwar einen eigenen Weg, orientiert sich jedoch in Teilaspekten an den Arbeiten von Dreo und Brenner.

Beiträge zu den Lösungskonzepten, die im Rahmen dieser Arbeit entwickelt und in den nachfolgenden Kernkapiteln vorgestellt werden, leisten direkt oder indirekt (eingeklammertes Häkchen in der Tabelle) viele der hier genannten Ansätze. Ein direkter Beitrag liegt vor, wenn Konzepte aus dem jeweiligen Ansatz in die Anforderungsanalyse oder in eines der späteren Kapitel einfließen oder aber verwendet werden können, um eine Lücke zu schließen, die diese Arbeit (bewusst) offen lässt.

Die Abgrenzung der mit dieser Arbeit gelieferten Ergebnisse und Lösungen von denen anderer Ansätze ist insbesondere im Forschungsumfeld und dort speziell bei den modellbasierten Ansätzen wichtig. Dabei soll aufgezeigt werden, inwieweit sich diese Arbeit von den anderen unterscheidet und entweder unterschiedliche Problemstellungen adressiert oder die gleichen Problemstellungen auf andere, neue Art und Weise zu lösen versucht.

<i>Kategorie</i>	<i>Autor/Titel</i>	<i>Relevanz für diese Arbeit</i>			
		<i>Begriffsbildung</i>	<i>Methodik</i>	<i>Lösungen</i>	<i>Abgrenzung</i>
Grundlegende Arbeiten	Dreo	✓	✓	✓	✓
	Lewis	✓		✓	✓
Prozess- rahmenwerke	ITIL	✓		✓	✓
	ISO/IEC 20000	✓		✓	✓
	eTOM			✓	✓
Modellbasierte Ansätze	SID			✓	✓
	Brenner		✓		✓
	Debusmann				✓
Sonstige Arbeiten	TMF SLA Mgmt.				✓
	Schmidt			✓	✓

Tabelle 4.1: Überblick der in diesem Kapitel vorgestellten Arbeiten und ihren Bezug zu dieser Dissertation

4.1.2 Analyse- und Bewertungsschema

Die Ergebnisse der funktionalen Anforderungsanalyse in Kapitel 3 liefern Maßgaben zur Entwicklung der Managementarchitektur. Dazu wurden die 15 Funktionsanforderungen an ein SLM-spezifisches Managementsystem bereits zu konkreten Anwendungsfällen verfeinert. Gleichzeitig liefern einige der in diesem Kapitel vorgestellten Arbeiten Beiträge zur Managementarchitektur, die in den Kontext einer oder mehrerer Funktionsanforderungen fallen. Mit anderen Worten: Die Erfüllung bestimmter funktionaler Anforderungen durch die Managementarchitektur kann unter Zuhilfenahme der hier präsentierten Ansätze leichter erreicht werden, da bereits Teillösungen für bestimmte Aspekte existieren.

Tabelle 4.2 illustriert das verwendete Analyse- und Bewertungsschema, das für jeden Ansatz mit Einfluss auf die Architekturentwicklung zur Anwendung kommt und in diesem Kontext auch zwischen Aspekten einer Prozess-Sicht und solchen einer System-Sicht differenziert. Damit wird eines der wesentlichen methodischen Merkmale des Ordnungsrahmens der Architektur aufgegriffen (vgl. Kapitel 2).

Es ist das Ziel dieses Analyse- und Bewertungsschemas, für jede Arbeit, die gemäß Tabelle 4.1 Beiträge zur Lösung von (Teil-)Fragestellungen der vorliegenden Arbeit liefert, in qualifizierter und strukturierter Weise zu untersuchen und darzustellen, wie groß ihr Beitrag im Hinblick auf die einzelnen funktionalen Anforderungen jeweils ist.

Es folgt ein Ein- und Überblick über die relevanten verwandten Arbeiten in den bereits genannten Kategorien *Grundlegende Arbeiten*, *Prozessrahmenwerke*, *Modellbasierte Ansätze* und *Andere Ansätze*.

4.2 Grundlegende Arbeiten

In diesem Abschnitt wird eine Auswahl verschiedenster Veröffentlichungen zum Themengebiet Dienst- und Service-Level-Management vorgestellt, die hilft, ein grundlegendes Verständnis über Zusammenhänge und Herausforderungen in dieser Disziplin aus verschiedensten Perspektiven zu gewinnen.

4.2.1 Dienstmanagement-Rahmenwerk (Dreo)

Als erste Arbeit der Kategorie *Grundlegende Arbeiten* wird das generische Dienstmanagement-Rahmenwerk von Dreo [4, 38] vorgestellt, das schon im

<i>Ebene</i>	<i>Symbol</i>	<i>Bedeutung</i>
Prozess	□□□	Der betrachtete Aspekt liegt thematisch außerhalb des Betrachtungsbereichs des analysierten Ansatzes.
	■□□	Der Ansatz beschäftigt sich mit Aspekten, die thematisch in den Kontext des Kriteriums fallen, liefert dabei allerdings nur sehr rudimentäre Beiträge für ein formales Prozessmodell.
	■ ■ □	Der Ansatz liefert im Hinblick auf das betrachtete Kriterium wesentliche Ansätze oder Teile eines formalen Prozessmodells (zum Beispiel Workflow-Definitionen, Teile eines Informationsmodells, Rollendefinitionen, etc.).
	■ ■ ■	Der Ansatz liefert im Hinblick auf das betrachtete Kriterium ein vollständiges Prozessmodell, das mindestens ein rudimentäres Rollenmodell, eine formale Workflow-Definition oder eine informelle Ablaufbeschreibung sowie eine Beschreibung von Informationsobjekten und ihren Zusammenhängen enthält.
System	□□□	Aspekte der Realisierung der betrachteten funktionalen Anforderung durch ein IT-gestütztes Managementsystem fallen nicht in den Betrachtungsbereich des analysierten Ansatzes.
	■□□	Der Ansatz beschäftigt sich mit Systementwurfsaspekten, die thematisch in den Kontext des Kriteriums fallen, liefert dabei allerdings nur sehr rudimentäre und spezifische Beiträge zum Ziel Automatisierung.
	■ ■ □	Der Ansatz liefert im Hinblick auf das betrachtete Kriterium einen maßgeblichen Beitrag mit Blick auf Automatisierungsaspekte.
	■ ■ ■	Der Ansatz liefert im Hinblick auf das betrachtete Kriterium einen vollständigen und generischen Systementwurf.

Tabelle 4.2: Schema und Symbolik zur Analyse und Bewertung des Beitrags zur Architekturentwicklung durch einen Lösungsansatz

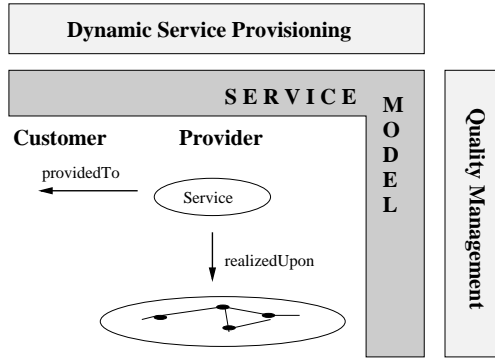


Abbildung 4.1: Kernbestandteile des Dienstmanagement-Rahmenwerks von Dreo

Kapitel 1 kurz angesprochen wurde. Obwohl diese Arbeit wichtige Aspekte des Service-Level-Managements beinhaltet, liegt ihr Hauptaugenmerk auf der ganzheitlichen Betrachtung des Dienstmanagements, die mit der Vision einer integrierten Dienstmanagementplattform verbunden wird.

4.2.1.1 Überblick

Das entwickelte Dienstmanagement-Rahmenwerk besteht aus drei Kernbestandteilen (vgl. Abbildung 4.1):

1. Statisches Dienstmodell (*Service Model*)
2. Modell der dynamischen Diensterbringung (*Dynamic Service Provisioning*)
3. Modell zum Dienstqualitätsmanagement (*Quality Management*)

Alle drei Module bilden eine ganz wesentliche konzeptionelle Basis für die im Rahmen dieser Arbeit zu entwickelnde Managementarchitektur für Service-Level-Management. Bereits im Kapitel 2 wurde auf das MNM-Dienstmodell Bezug genommen, um ein grundlegendes und technisch-fundiertes Verständnis über IT-Dienste aufzubauen. Dieses Dienstmodell stellt auch die Grundlage für Dreos Rahmenwerk und ihre Erweiterungen im Rahmen des statischen Dienstmodells dar. Der Dienstlebenszyklus, der ein Teil des Modells der dynamischen

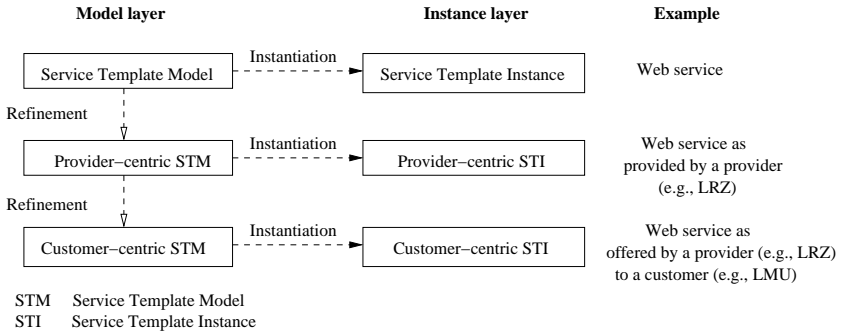


Abbildung 4.2: Anwendung von Dienst-Template-Modellen nach Dreo [4]

Diensterbringung ist, wurde in Kapitel 3 instrumentiert, um die ermittelten funktionalen Anforderungen auf die verschiedenen Phasen der Diensterbringung abzubilden.

Abbildung 4.2 illustriert die Anwendung verschiedener Arten von Dienst-Template-Modellen, die eine zentrale Komponente des statischen Dienstmodells im Kontext des Dienstmanagement-Rahmenwerks darstellen. Ausgehend vom MNM-Dienstmodell als eine Art Meta-Modell liefert die Arbeit von Dreo ein allgemeines *Service Template Model* (STM), welches zunächst in ein Provider-bezogenes STM verfeinert und zuletzt in ein kundenbezogenes STM überführt wird. Auf dieser Ebene, der Modellebene, werden weder konkrete Dienste, noch konkrete Provider oder Kunden betrachtet. Die drei verschiedenen STM-Modellvarianten spezifizieren vielmehr allgemein, welche managementrelevanten Attribute benötigt werden, um einen spezifischen Dienst oder eine bestimmte Dienst-Art (wie zum Beispiel *Web Service*) zunächst provider- und kundenunabhängig zu beschreiben, welche *zusätzlich* benötigt werden, um den Dienst in seiner Realisierung durch einen bestimmten Provider zu beschreiben, und welche Attribute schließlich hinzukommen, wenn eine Dienstinstanz von einem Provider an einem bestimmten Kunden geliefert wird.

Ein Dienstportfolio enthält Instanzen des Provider-bezogenen STMs, ein Dienstkatalog Instanzen des Kunden-bezogenen STMs, und eine Dienstvereinbarung (SLA) referenziert die kundenspezifischen STM-Instanzen eines Dienstkatalogs. Dies zeigt deutlich, wie gut sich das Rahmenwerk von Dreo auf die Domäne des Service-Level-Managements anwenden lässt. Daher beeinflusst es auch die Architekturentwicklung in einigen Punkten maßgeblich.

Anforderung	Erfüllungsgrad	
	Prozess	System
F1 (Dienstinf.)	■ ■ □	■ ■ □
F2 (Kunden)	■ □ □	■ □ □
F3 (FA/Lief.)	■ □ □	■ □ □
F4 (Dienstkat.)	■ ■ □	■ ■ □
F5 (Supp.-Level)	□ □ □	□ □ □
F6 (SLAs)	■ ■ □	■ ■ □
F7 (OLAs)	■ □ □	■ □ □
F8 (UCs)	■ □ □	■ □ □
F9 (Konflikte)	□ □ □	□ □ □
F10 (Messung)	■ □ □	■ □ □
F11 (Berichte)	□ □ □	□ □ □
F12 (Entstörung)	□ □ □	□ □ □
F13 (Verrechnung)	■ □ □	■ □ □
F14 (Transition)	□ □ □	□ □ □
F15 (Sicherheit)	□ □ □	□ □ □

Tabelle 4.3: Bewertung des Lösungspotenzials des Dienstmanagement-Rahmenwerks von Dreo im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

4.2.1.2 Bewertung

Tabelle 4.3 illustriert das Ergebnis der Anwendung des Analyse- und Bewertungsschemas auf das Dienstmanagement-Rahmenwerk von Dreo. Sie macht deutlich, dass die größten Beiträge dieses Rahmenwerks für die vorliegende Arbeit in den Bereichen der Verwaltung von Dienstinformationen (vgl. Anforderung F1), Dienstkatalogen (vgl. Anforderung F4) und Dienstvereinbarungen (vgl. Anforderung F6) liegen.

4.2.2 SLM für Unternehmensnetze (Lewis)

Während Dreos Fokus ein generisches Rahmenwerk für ein ganzheitliches Dienstmanagement war, beschäftigt sich Lewis in seinem Buch *Service Level Management for Enterprise Networks* [6] spezifisch mit der Domäne Service-Level-Management und liefert eine umfassende und durchaus auch wegweisende Auseinandersetzung mit diesem Themengebiet.

4.2.2.1 Überblick

Das Buch gibt eine Einführung in das Thema Service-Level-Management, die zunächst ein intuitives Verständnis über die Notwendigkeit, Ziele und Herausforderungen von SLM aufbaut, und definiert im zweiten Kapitel eine Reihe von Begriffen und Konzepten, die zum Teil auch in dieser Arbeit eine Rolle spielen und gemäß der Definitionen von Lewis verwendet werden, sofern nicht explizit anders festgelegt.

Unter anderem werden folgende Begriffe und Konzepte definiert: *business process*, *enterprise network*, *service*, *service parameter*, *component parameter*, *component-to-service parameter mapping*, *service level*, *service level agreement*, *service level report*

Aufbauend auf den Begriffs- und Konzeptdefinitionen stellt Lewis anschließend eine *SLM-Methodik (SLM Methodology)* und eine *SLM-Architektur (SLM Architecture)* vor. Die Methodik beschäftigt sich mit der Einführung von Service-Level-Management in einem *Enterprise Network* einschließlich der Entwicklung eines *SLM-Systems*. Die Architektur stellt dar, wie Lösungsansätze aus den Bereichen *Künstliche Intelligenz*, *Robotik* und *Data Warehousing* in die Konzeption eines automatisierten SLM-Systems einfließen können. Beide Kapitel – Methodik wie Architektur – beschäftigen sich schwerpunktmäßig mit technischen Aspekten des Service-Level-Managements auf einer konzeptionellen Ebene. Messung und Überwachung (*Monitoring*) von Ressourcen und Diensten stellen ein zentrales Thema in diesem Ansatz dar.

4.2.2.2 Bewertung

Methodik und Architektur von Lewis befassen sich in erster Linie mit der Überwachung der in den bestehenden SLAs zugesicherten Leistungsparameter. Damit unterscheidet sich Lewis' Architektur in ihrer Zielsetzung deutlich von der im Rahmen dieser Arbeit zu entwickelnden Managementarchitektur. Entwurfsprozesse (zum Beispiel von Dienstvereinbarungen oder Diensten) spielen keine Rolle. Ein Informationsmodell ist ebensowenig Gegenstand der Betrachtungen von Lewis wie die Einbettung von Service-Level-Management in ein umfassendes Dienstmanagement.

Wie Tabelle 4.4 zeigt, fließen aufgrund dieser verschiedenen Zielsetzungen nur wenige Inhalte der Methodik und Architektur von Lewis in diese Arbeit ein. Deutlich stärker kann hingegen von den Begriffs- und Konzeptdefinitionen profitiert werden.

Anforderung	Erfüllungsgrad	
	Prozess	System
F1 (Dienstinf.)	□□□	□□□
F2 (Kunden)	□□□	□□□
F3 (FA/Lief.)	□□□	□□□
F4 (Dienstkat.)	□□□	□□□
F5 (Supp.-Level)	□□□	□□□
F6 (SLAs)	■□□	■□□
F7 (OLAs)	□□□	□□□
F8 (UCs)	□□□	□□□
F9 (Konflikte)	□□□	□□□
F10 (Messung)	■□□	■■□
F11 (Berichte)	□□□	■□□
F12 (Entstörung)	□□□	□□□
F13 (Verrechnung)	□□□	□□□
F14 (Transition)	□□□	□□□
F15 (Sicherheit)	□□□	□□□

Tabelle 4.4: Bewertung des Lösungspotenzials der SLM-Methodik und -Architektur von Lewis im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

4.3 Prozessrahmenwerke

Prozessorientierung ist eine der drei Säulen des methodischen Vorgehens dieser Arbeit (vgl. Kapitel 2, Abbildung 2.1). Ansätze zur Anwendung dieses Grundprinzips im Management von IT-Diensten haben ihren Ursprung primär in der industriellen Praxis. Das gilt auch und ganz besonders für den heute am weitesten verbreiteten Leitfaden zur Implementierung von Dienstmanagementprozessen, die *IT Infrastructure Library* (ITIL), die nun als erstes vorgestellt wird.

4.3.1 IT Infrastructure Library (OGC)

Die *IT Infrastructure Library* (ITIL) ist eine Sammlung von Büchern mit Empfehlungen (*Best Practices*) für ein prozessorientiertes Dienstmanagement. Verantwortlich für Inhalte und Weiterentwicklung dieses Rahmenwerks ist das britische *Office of Government Commerce* (OGC).

4.3.1.1 Überblick

Wenn heute von ITIL die Rede ist, muss zwischen der Version 2, deren Bücher zwischen 2000 und 2006 verfasst und herausgegeben wurden, sowie der Version 3 unterschieden werden, die seit 2007 auf dem Markt ist. Eine klare Differenzierung ist vor allem deshalb wichtig, weil die Unterschiede struktureller und inhaltlicher Art zwischen beiden Versionen weitreichend sind.

Natürlich gibt es aber auch eine Vielzahl an Gemeinsamkeiten: So verfolgen beide Versionen von ITIL das Ziel, die Einführung und Umsetzung von IT-Service-Management in einer IT-Organisation zu fördern und durch entsprechende Empfehlungen zu unterstützen. Dabei nimmt ITIL in der Hauptsache Bezug auf die organisatorischen Herausforderungen in diesem Zusammenhang und formuliert unter anderem die Ziele der einzuführenden Managementprozesse, beschreibt wesentliche Aktivitäten, definiert die zur Prozessausführung benötigten Rollen und gibt – wenn auch in begrenztem Umfang – Beispiele aus der Praxis. Auch Prozessschnittstellen und rudimentäre Anforderungen an die im Rahmen der Prozesse zu pflegenden Informationen finden Erwähnung.

ITIL Version 2 Tabelle 4.5 gibt einen Überblick über die Kernbestandteile von ITIL in der Version 2. Da nicht alle Bücher der *Core Guidance* im gemeinsamen Qualifizierungsprogramm von EXIN und ISEB berücksichtigt wurden, wurde ITIL Version 2 zuletzt hauptsächlich mit den Titeln *Service Support* [10], *Service Delivery* [30] und *Security Management* [39] und den enthaltenen Prozessen in Verbindung gebracht, was aber nicht die volle Bandbreite dieses Rahmenwerks erfasst.

Zusätzlich zu den sieben Kernbestandteilen wurden unter anderem die Bücher *Introduction to ITIL* [44], *Software Asset Management* [45] und *ITIL Small-scale Implementation* [46] herausgebracht. Ersteres gibt einen zusammenfassenden Überblick über die Inhalte der Kernbücher. Der zweite Titel beschäftigt sich mit Aspekten des Managements von Software und Lizenzen, der dritte gibt Empfehlungen zur ITIL-Einführung im kleineren Maßstab.

ITIL Version 3 Der Schritt von ITIL in der Version 2 zur Version 3 war mehr als eine reine Revision oder Erweiterung. Vielmehr wurde der gesamte hinter den eigentlichen Managementprozessen stehende Ansatz weiterentwickelt und zum Teil verändert. Zum für ITIL-Prozesse ab Version 3 relevanten Strukturierungsmittel wurde ein Dienstlebenszyklus, der hier aus den fünf Phasen Strategie (*Service Strategy*), Planung und Entwurf (*Service Design*), Überführung (*Service Transition*), Betrieb (*Service Operation*) und Verbesserung

<i>ITILv2-Band</i>	<i>Inhalte/Beschriebene Prozesse</i>
Service Support [10]	Incident Management, Problem Management, Configuration Management, Change Management, Release Management
Service Delivery [30]	Service Level Management, Availability Management, Capacity Management, IT Service Continuity Management, Financial Management for IT Services
Security Management [39]	Security Management
Application Management [40]	Application Management
Infrastructure Management [41]	ICT Infrastructure Management
Planning to Implement [42]	Empfehlungen zur Einführung eines prozessorientierten Ansatzes im Dienstmanagement
Business Perspective [43]	Erfolgsfaktoren aus Sicht der Geschäftsprozesse

Tabelle 4.5: Überblick über die zentralen Bestandteile von ITIL Version 2

(*Continual Service Improvement*) besteht. Insgesamt verfolgt ITIL Version 3 einen sogenannten *Lifecycle Approach*. Und so besteht die *Core Guidance* von ITIL Version 3 aus fünf Büchern, von denen jedes einen Teil des *Service Lifecycle* beschreibt (vgl. Tabelle 4.6).

ITIL als Konzept-Rahmenwerk Neben den eigentlichen Prozessen gibt es einige Konzepte, die durch ITIL geprägt oder teilweise erstmals explizit benannt wurden. Eines der prominentesten Konzepte ist die *Configuration Management Database* (CMDB), die mittlerweile von nahezu jedem Entwickler unterstützender ITSM-Software aufgegriffen wurde. Hinter der CMDB steht die Idee eines umfassenden Informationssystems zur Speicherung aller managementrelevanter Informationen zu jedem Konfigurationsobjekt (*Configuration Item*, CI) – von Hardware-Komponenten über Softwarelizenzen bis hin zu abstrakten Konfigurationsobjekten wie Diensten. Daneben soll eine CMDB Auskunft über Zusammenhänge und Abhängigkeiten zwischen verschiedenen CIs geben können und somit insbesondere zur Auswirkungsanalyse in operativen Prozessen wie dem Störungs- oder Änderungs-Management eine zentrale Rolle

<i>ITILv3-Band</i>	<i>Inhalte/Beschriebene Prozesse (Auszug)</i>
Service Strategy [2]	Financial Management, Service Portfolio Management, Demand Management, u.a.
Service Design [25]	Service Catalogue Management, Service Level Management, Capacity Management, Availability Management, IT Service Continuity Management, Information Security Management, Supplier Management
Service Transition [47]	Transition Planning and Support, Change Management, Service Asset and Configuration Management, Release and Deployment Management, Service Validation and Testing, Evaluation, Knowledge Management
Service Operation [48]	Event Management, Incident Management, Request Fulfilment, Problem Management, Access Management
Continual Service Improvement [49]	Service Improvement, Service Reporting, Service Measurement, u.a.

Tabelle 4.6: Überblick über die zentralen Bestandteile von ITIL Version 3

einnehmen.

4.3.1.2 Bewertung

Von allen Prozessrahmenwerken ist die *IT Infrastructure Library* (ITIL) das weltweit am weitesten verbreitete. Die Gründe hierfür sind vielfältig. Im Gegensatz zu anderen Ansätzen hat es ITIL vor allem durch die im Vergleich zur ersten Version stark konsolidierte Version 2 geschafft, ein äußerst einfaches, nachvollziehbares und generisch anwendbares Rahmenwerk für IT-Service-Management anzugeben.

Was vor allem im Forschungs- und Entwicklungsumfeld oft als größtes Problem angesehen wird, das die ITIL mit sich bringt, ist zugleich ihre größte Stärke in Anwenderkreisen: die Einfachheit und Oberflächlichkeit, in der *Best Practices*

und Prozesse dargestellt werden. Gerade für die ersten Schritte in Richtung eines prozessorientierten Dienstmanagements und für die Erzielung schneller, sichtbarer Ergebnisse mit einfachen Mitteln hat sich ITIL in der Version 2 bewährt.

Die Lücken, die ITIL offen lässt, treten vor allem bei der Entwicklung unterstützender Software-Tools in Erscheinung und erweisen sich hier als besonders problematisch. Denn der Spielraum, den ITIL für die Implementierung seiner Prozesse in konkreten Betriebsumgebungen lässt, führt in diesem Zusammenhang zu einem Mangel an verwertbaren Anhaltspunkten für die Spezifikation der erforderlichen Systemeigenschaften. An dieser Stelle setzt diese Arbeit an, um für einen wichtigen Teilbereich des Dienstmanagements, nämlich das Service-Level-Management, diese Lücke zu schließen.

Im Ergebnis leistet ITIL wichtige und zielführende Beiträge für die Entwicklung der Managementarchitektur im Rahmen dieser Arbeit. Tabelle 4.7 zeigt, dass besonders die Prozess-Sicht der Architektur von den Inhalten und Empfehlungen aus ITIL profitieren kann. Hier deckt ITIL vor allem eine große Breite ab. Im Hinblick auf generische Automatisierungsaspekte, wie sie auf der Ebene der architekturellen System-Sicht adressiert werden, liefert ITIL nahezu keinen Beitrag.

4.3.2 ISO/IEC 20000

ISO/IEC 20000 ist ein internationaler Standard für IT-Service-Management, der als Nachfolger des britischen Standards BS 15000 [50, 51] stark auf ITIL Version 2 aufbaut. Neben der Möglichkeit für IT-Organisationen, sich gegen ISO/IEC 20000 zertifizieren zu lassen, gibt es auch ein Qualifizierungsprogramm für Personen.

4.3.2.1 Überblick

Der Standard ISO/IEC 20000 besteht, wie in Tabelle 4.8 dargestellt, aus zwei Dokumenten, von denen das erste [52] die Mindestanforderungen spezifiziert, das zweite [31] zusätzliche Empfehlungen enthält. Beide Dokumente sind identisch aufgebaut und strukturiert.

Das Prozess-Rahmenwerk von ISO/IEC 20000 wird in Kapitel 5.4.2 als Ausgangspunkt für das Interprozess-Kommunikationsmodell verwendet. In diesem Kontext werden die enthaltenen Prozesse kurz eingeführt.

<i>Anforderung</i>	<i>Erfüllungsgrad</i>	
	Prozess	System
F1 (Dienstinf.)	■□□	□□□
F2 (Kunden)	■□□	□□□
F3 (FA/Lief.)	■□□	□□□
F4 (Dienstkat.)	■■□	□□□
F5 (Supp.-Level)	■□□	□□□
F6 (SLAs)	■■□	□□□
F7 (OLAs)	■■□	□□□
F8 (UCs)	■■□	□□□
F9 (Konflikte)	□□□	□□□
F10 (Messung)	■□□	□□□
F11 (Berichte)	■□□	□□□
F12 (Entstörung)	■■□	□□□
F13 (Verrechnung)	■□□	□□□
F14 (Transition)	■■□	□□□
F15 (Sicherheit)	■□□	□□□

Tabelle 4.7: Bewertung des Lösungspotenzials von ITIL Version 3 im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

<i>Dokument</i>	<i>Kurztitel</i>	<i>Inhalte (Überblick)</i>
ISO/IEC 20000-1 [52]	Part 1: Specification	Mindestanforderungen an einen IT-Provider, um eine Zertifizierung nach der Norm zu erhalten
ISO/IEC 20000-2 [31]	Part 2: Code of Practice	Zusätzliche Leitlinien und Empfehlungen für Provider und Auditoren, nicht Teil der Anforderungen im Rahmen eines Zertifizierungsaudits

Tabelle 4.8: Bestandteile des Standards ISO/IEC 20000

4.3.2.2 Bewertung

Speziell im Hinblick auf das Prozessrahmenwerk bietet ISO/IEC 20000 im direkten Vergleich mit ITIL deutlich weniger Inhalte. Der Mehrwert von ISO/IEC 20000 besteht sogar eher darin, dass die teilweise sehr umfangreichen Ausführungen aus ITIL „destilliert“ und auf die Kernanforderungen und -empfehlungen reduziert werden. Dies ist für den wichtigsten Einsatzbereich der Norm, die Durchführung von Zertifizierungsaudits, nicht nur hilfreich, sondern geradezu essenziell. Im Kontext dieser Arbeit führt es aber dazu, dass ISO/IEC 20000 nicht mit eigenen Lösungskonzepten in die Architekturentwicklung einfließt, sondern in erster Linie die Begriffsbildung unterstützt. Somit entfällt die Anwendung des Analyse- und Bewertungsschemas.

4.3.3 Enhanced Telecom Operations Map (TMF)

Während ITIL ihren Ursprung im *Data Center*-Umfeld hat, ist die Entstehung der *enhanced Telecom Operations Map* (eTOM) im Bereich der Anbieter von Telekommunikationsdienstleistungen anzusiedeln. Beide Ansätze vereint das Prinzip der Prozessorientierung, welches bei eTOM allerdings in einer deutlich anderen Ausprägung zur Anwendung kommt. Im Gegensatz zu ITIL strukturiert eTOM die enthaltenen Prozesse nicht flach, sondern in einer Hierarchie aus insgesamt fünf Ebenen.

4.3.3.1 Überblick

Das gesetzte Ziel der eTOM besteht darin, alle Managementaufgabenbereiche eines modernen Informations- und Telekommunikationsdienstleisters durch Prozessempfehlungen zu unterstützen. Dabei ist eTOM nicht auf IT-spezifische Managementaufgaben beschränkt, sondern deckt auch klassisches *Enterprise Management* ab.

Einen Service-Level-Management-Prozess gibt es in eTOM in der Form wie in ITIL und ISO/IEC 20000 nicht. In einer gemeinsamen Veröffentlichung des *TeleManagement Forums* (TMF) und des *IT Service Management Forums* (itSMF) [57], die sich mit Möglichkeiten der Integration von eTOM und ITIL (Version 3) beschäftigt, wird vorgeschlagen, die eTOM um ein neues Level-2-Prozesselement mit dem Namen *ITIL Service Level Management* zu ergänzen, das der Level-1-Prozessgruppe *1.3.3 Enterprise Effectiveness Management* zugeordnet wird.

<i>Dokument</i>	<i>Kurztitel</i>	<i>Inhalte (Überblick)</i>
GB921-P [53]	eTOM Primer	Gesamtüberblick, Ziele, Adressaten
GB921 [20]	eTOM Main	Hauptdokument: Zielsetzung von eTOM, Prozesse der Level 0 (Prozessbereiche) und 1 (Prozessgruppen)
GB921-D [54]	Addendum D	Prozesse (Prozesselemente) der Level 2 und 3
GB921-F [55]	Addendum F	Beispiele für Prozessablaufmodelle (<i>process flows</i>)
GB921-U [56]	Addendum U	Anwenderleitfaden

Tabelle 4.9: Überblick über die zentralen Dokumente der eTOM

4.3.3.2 Bewertung

Im Ergebnis liefert eTOM allerdings weniger konkrete Beiträge für den Entwurf der Managementarchitektur für Service-Level-Management als beispielsweise ITIL. Der Grund dafür liegt darin, dass eTOM nochmals umfassender ist als ITIL und größeren Wert auf das Zeichnen einer strukturierten „Prozesslandkarte“ legt als auf die Darstellung spezifischer Prozesselemente in ihren Einzelheiten. Tabelle 4.10 zeigt, wo eTOM zur Erfüllung der funktionalen Anforderungen im Kontext der Prozess-Sicht der Architektur Hilfestellung bietet, und in welcher Ausprägung.

4.4 Modellbasierte Ansätze

Diese dritte Kategorie umfasst Ansätze, die sich mit der Modellierung bestimmter Aspekte des Dienst- oder Service-Level-Managements befassen. Informationsmodellierung stellt in diesem Zusammenhang einen Schwerpunkt dar.

4.4.1 Shared Information/Data Model (TMF)

Das *Shared Information/Data Model* (SID) ist ein objektorientiertes Managementinformationsmodell für IT- und Telekommunikationsdienstleister. Es

Anforderung	Erfüllungsgrad	
	Prozess	System
F1 (Dienstinf.)	■□□	□□□
F2 (Kunden)	■□□	□□□
F3 (FA/Lief.)	■□□	□□□
F4 (Dienstkat.)	□□□	□□□
F5 (Supp.-Level)	■□□	□□□
F6 (SLAs)	■□□	□□□
F7 (OLAs)	□□□	□□□
F8 (UCs)	□□□	□□□
F9 (Konflikte)	□□□	□□□
F10 (Messung)	■□□	□□□
F11 (Berichte)	■□□	□□□
F12 (Entstörung)	■□□	□□□
F13 (Verrechnung)	■□□	□□□
F14 (Transition)	■□□	□□□
F15 (Sicherheit)	■□□	□□□

Tabelle 4.10: Bewertung des Lösungspotenzials von eTOM im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

ist – wie auch eTOM – Teil des *New Generation Operations Systems and Software*-Programms (NGOSS) des *TeleManagement Forums* (TMF). SID ergänzt das Prozess-Rahmenwerk der eTOM, indem es sich mit ausgewählten Entitäten beschäftigt, die im Rahmen der eTOM-Prozesse eine Rolle spielen. Die Autoren versprechen, dass SID einen Großteil der Informationsanforderungen von Anbietern von IT- und Telekommunikationsdiensten abdeckt.

4.4.1.1 Überblick

Die übergeordnete Motivation für SID war gegeben durch die Notwendigkeit eines Informationsmodells als Grundlage für alle Managementaktivitäten eines modernen Telekommunikationsdiensteanbieters, wie sie durch eTOM in prozessorientierter Weise beschrieben sind. SID wird in einer Reihe von Dokumenten umfassend beschrieben. Einen übergeordneten Überblick gibt der *SID Primer* [58]. Darin wird der Begriff des *Informationsmodells* wie folgt definiert:

Informationsmodell Ein Informationsmodell ist eine Repräsentation von Geschäftskonzepten (*business concepts*), ihren Charakteristika und Bezie-

hungen, beschrieben in einer implementierungsunabhängigen Art und Weise.

Die Vorteile, die sich aus der Verfügbarkeit eines solchen Informationsmodells ergeben, werden in diesem Dokument so beschrieben:

- Durch die Vorgabe einer einheitlichen Terminologie wird das Informationsmanagement erleichtert.
- Ein Informationsmodell ermöglicht die Vereinheitlichung von Informationen innerhalb eines Unternehmens sowie zwischen verschiedenen Unternehmen.
- Es schlägt eine Brücke zwischen geschäftlichen Aktivitäten (*the business*) und den IT-Abteilungen, indem es Definitionen liefert, die für die Akteure in den Geschäftsprozessen verständlich und zugleich verbindlich genug für den Einsatz in der Softwareentwicklung sind.

<i>Dokument</i>	<i>Haupt-Konzept</i>	<i>Zugehörige Business Entities (Auszug)</i>
922-1A [59]	Agreement	AgreementApproval, AgreementAuthorization, AgreementTermOrCondition, AgreementItem, ServiceLevelAgreement, ServiceLevelAgreementItem
922-1BI [60]	Business Interaction	BusinessInteractionSpec, -Item, -Role, -Location, -Relationship, Notification, Request, Response, Command, InquiryRequest
922-1BT [61]	Business Entity Base Types	TimePeriod, Duration, Quantity, Money, Rate, CostRate, TimeUnit
922-1C [62]	Business Contract	BusinessContractSpec, BusinessBenefits, -Obligations, -Capabilities, -Context, -Activity, -Goal, -ContractLimitations

922-1J [63]	Project	ProjectElement, ProjectSpecification, SimpleActivity, ActivityStatus, ActivitySpecification ProjectElementRelationship, -Succession, -Substitution, -Division, -Fusion, -Dependency, -Specification
922-1L [64]	Location	GeographicPlace, -Site, -SiteRole, -Location, -LocationType, -LocationName, -LocationRelationship, -Address, -SubAddress, Country, State
922-1P [65]	Party	Individual, Organization, Name, Role, RoleType, PartyRoleAssociation, Language, Identification
922-1POL [66]	Policy	PolicyRule, -Event, -Set, -Condition, -Action, -Domain, -Conflict, -Decision, -DecisionPoint, -EnforcementPoint, -ExecutionPoint, -Repository, -Server, -Subject, -Target
922-1R [67]	Root Business Entities	Entity, Specification, Identification, Role
922-1T [68]	Time Related Entities	Calendar, CalendarEntry, -EntryParticipation, -ActionEntry, -JournalEntry, -Event, -ToDo, -Alarm, TemporalExpression

922-2 [69]	Customer	CustomerOrder, -OrderItem, -ServiceLevelAgreement, -Interaction, -Billing, -Account, -AccountContact, -AccountRelationship, LocationServiceRequest, AccessServiceRequest, DirectoryServiceRequest
922-3 [70]	Product	ProductSpecification, ProductOffering, ProductPrice, ProductBundle, ProductComponent
922-4SO [71]	Service Overview	CustomerFacingService, ResourceFacingService, CustomerFacingServiceSpec, ResourceFacingService, ServicePackageSpec, ServiceLevelSpecification, ServiceLevelObjective, ServiceLevelSpecParameter
922-4S-QoS [72]	Quality of Service	QoSService, ToSService, 802Service, DiffServService, AFSservice, EFSservice, NetworkForwardingService, TrafficIdentificationService, ClassifierService, MarkerService, SchedulingService
922-5LR [73]	Logical Resource	Software, OperatingSystem, DeviceInterface, CompoundResource, ResourceCollection, ResourceElement

922-5PR [74]	Physical Resource	PhysicalResourceSpecification, PhysicalResourceRole, PhysicalDevice, Hardware, ManagedHardware, PhysicalPort, PhysicalContainer, Card, MemoryCard, NetworkCard, Slot, HolderComposite, Rack, Chassis
922-6 [75]	Market/Sales	MarketSegment, MarketSegmentCharacteristic, MarketSegmentCharacteristicValue, PartyDemographic, PartyDemographicValue, PartyProfile, PartyType, Competitor, CompetitorMarketSegment, CompetitorMarketSegmentSWOT, CompetitorSWOT, CompetitorTier, SWOT
922-7RA [76]	Enterprise Domain Revenue Assurance Business Entities	RevenueAssuranceParameter, -Control, -KPI, -ParamValue, -Objective, -Consequence, -ViolationSpec, -Violation, -TroubleTicketSpec, -TroubleTicket, -ActionSpec, -Action, -ResponseSpec, -Response, -Assessment, -AssessmentItem, -AssessmentResult, -Recommendation

Tabelle 4.11: Überblick über die Domänen-Addenda des SID

4.4.1.2 Bewertung

Tabelle 4.12 zeigt, dass SID auf beiden Ebenen der Architekturentwicklung eine Rolle spielt, wenn auch nur in begrenztem Maße und beschränkt auf den

Anforderung	Erfüllungsgrad	
	Prozess	System
F1 (Dienstinf.)	□□□	■□□
F2 (Kunden)	□□□	■□□
F3 (FA/Lief.)	□□□	■□□
F4 (Dienstkat.)	□□□	□□□
F5 (Supp.-Level)	□□□	□□□
F6 (SLAs)	□□□	■□□
F7 (OLAs)	□□□	■□□
F8 (UCs)	□□□	■□□
F9 (Konflikte)	□□□	□□□
F10 (Messung)	□□□	■□□
F11 (Berichte)	□□□	■□□
F12 (Entstörung)	□□□	□□□
F13 (Verrechnung)	□□□	□□□
F14 (Transition)	□□□	□□□
F15 (Sicherheit)	□□□	□□□

Tabelle 4.12: Bewertung des Lösungspotenzials von SID im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

Aspekt der Informationsmodellierung.

4.4.2 Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement (Brenner)

Brenner beschäftigt sich in seiner Dissertationsschrift *Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement* [9] zunächst allgemein mit Fragen der Unterstützung von Dienstmanagementprozessen durch geeignete Management-Tools und verfolgt anschließend einen modellbasierten Ansatz für eine spezifische Auswahl von ITIL-Prozessen.

4.4.2.1 Überblick

Die Arbeit von Brenner besteht aus einem umfangreichen analytischen Teil und einem modellbasierten produktiven Teil. Im ersten Teil analysiert Brenner die Möglichkeiten und Grenzen der Werkzeugunterstützung von Dienstmanagementprozessen. Der Klassifizierungsansatz ist generisch und auf prinzipiell

jedes Prozessrahmenwerk anwendbar. Es werden vier prinzipielle Klassen prozessorientierter Dienstmanagement-Werkzeuge definiert: Workflow-orientierte Managementwerkzeuge, Workgroup-Computing-orientierte Managementwerkzeuge, Administrationswerkzeuge sowie eigenständige und technische Werkzeuge. In der Durchführung der Analyse konzentriert sich Brenner auf die Prozesse aus ITIL Version 2. Ein wesentliches Ergebnis ist die Zuordnung der ITIL-Prozesse zu den genannten Klassen von Dienstmanagement-Werkzeugen.

Für die Prozesse *Incident Management*, *Problem Management* und *Change Management* ergibt sich eine Zuordnung in den Quadranten der Workflow-orientierten Managementwerkzeuge. Hier setzt der produktive, modellbasierte Teil der Arbeit an. Er liefert Prozessmodelle, die in einen Ordnungsrahmen aus drei horizontalen Schichten (*Owner View*, *Designer View* und *Builder View*) und vier vertikalen Schichten (*Data*, *Function*, *Network* und *People*) eingegliedert werden.

4.4.2.2 Bewertung

Da sich Brenner in seinem modellbasierten Lösungsansatz nicht mit Aspekten des Service-Level-Managements beschäftigt, liefert seine Arbeit für die Entwicklung einer Managementarchitektur für SLM auch keine inhaltlich-konzeptionellen Beiträge. Entsprechend entfällt die Anwendung des Analyse- und Bewertungsschemas. Methodisch und strukturell allerdings bietet Brenner mit seinem Ordnungsrahmen ein Fundament, auf dem diese Arbeit in Teilen aufbaut. Auch die von Brenner entwickelte Taxonomie zur Klassifizierung von Dienstmanagementprozessen unter *Tool Support*-Aspekten wurde bereits in Kapitel 1 aufgegriffen.

4.4.3 Modellbasiertes SLM (Debusmann)

In seiner Dissertation aus dem Jahr 2005 [77], vorgelegt an der Universität Kassel, beschäftigt sich Markus Debusmann mit Aspekten der Spezifikation von Dienstvereinbarungen (SLAs) und ihrer automatischen Überwachung. Zwei wesentliche Fragestellungen werden von Debusmann adressiert:

1. Wie können Dienst(güte)vereinbarungen einheitlich spezifiziert werden?
2. Wie können spezifizierte Dienst(güte)vereinbarungen möglichst automatisiert etabliert und überwacht werden?

4.4.3.1 Überblick

Eine der zentralen Ideen der Arbeit besteht in der Definition und Verwendung von *Mustern* zur Verwendung bei der Modellierung von Dienstvereinbarungen und der Anwendung des Ansatzes *Model Driven Architecture* (MDA) auf die beiden genannten Problemstellungen. Hierzu muss das MDA-Konzept aber zunächst auf die Problemdomäne des Service-Level-Managements übertragen werden. Das Ergebnis ist ein SLA-Muster, aus dem eine SLA-Instanz erzeugt werden kann, die in einem weiteren Schritt so in das verteilte System eingebunden wird, dass eine automatisierte Überwachung durch das eingesetzte Netzmanagementsystem ermöglicht wird.

4.4.3.2 Bewertung

Die Arbeit von Debusmann behandelt eine Fragestellung von großer Relevanz für ein effektives Service-Level-Management. Die Betrachtung der automatisierten Überwachung instanziierten Dienstvereinbarungen durch ein angebundenes Infrastruktur-Überwachungswerkzeug liefert allerdings keine Lösungsansätze für die funktionalen Anforderungen, die dieser Arbeit zugrunde liegen. Entsprechend kann die Anwendung des Analyse- und Bewertungsschemas entfallen.

4.5 Sonstige Ansätze

In dieser letzten Sektion werden das *NGOSS SLA-Management-Handbuch* des TMF sowie die Dissertation von Schmidt mit dem Titel *Entwurf von Service Level Agreements auf der Basis von Dienstprozessen* vorgestellt und bewertet, die sich jeweils nicht eindeutig einer der drei Kategorien *Grundlegende Arbeiten*, *Prozessrahmenwerke* und *Modellbasierte Ansätze* zuordnen lassen, für diese Arbeit aber trotzdem relevant im Sinne der Klassifizierung in Tabelle 4.1 sind.

4.5.1 NGOSS SLA-Management-Handbuch (TMF)

Mit der *SLA Management Handbook Solution Suite* verfolgt das TMF das Ziel, ein umfassendes Rahmenwerk für die Entwicklung von SLAs bereitzustellen, das über die vorhandenen Prozesse und *Best Practices* von eTOM und anderen

Ansätzen hinausgeht bzw. diese ergänzt. Die *SLA Management Handbook Solution Suite* ist Teil des TMF-Programms *New Generation Operations Systems and Software* (NGOSS) [78] und besteht aus vier Bänden:

1. Executive Overview [79]
2. Concepts and Principles [80]
3. Service and Technology Examples [81]
4. Enterprise Perspective [82] – angekündigt

4.5.1.1 Überblick

Da die *SLA Management Handbook Solution Suite* viele spezifische Aspekte aus der Disziplin Service-Level-Management adressiert und insbesondere im zweiten Band (*Concepts and Principles*) konkrete Managementverfahren und -konzepte darstellt, bildet sie eine wertvolle Ideen-gebende Quelle für die Konzeption der Architekturteilmodelle (vgl. Einordnung in Tabelle 4.1) zur Erfüllung der funktionalen Anforderungen und Realisierung der Primäranwendungsfälle. Überblicksartig werden die wesentlichen Inhalte der Bände 1 bis 3 der Handbuch-Sammlung im Folgenden zusammengefasst.

Band 1: Executive Overview Band 1 der *SLA Management Handbook Solution Suite* gibt eine Übersicht über SLA-Management im Allgemeinen und richtet sich in erster Linie an Entscheider auf Unternehmens- und IT-Ebene. Die tatsächlich erreichte Dienstgüte (*Quality of Service*, QoS) kann stets nur so gut sein wie die des zugrunde liegenden Dienst- oder Netzdesigns. Diese Beschränkung kann dazu führen, dass die Leistung des Dienstes die Anforderungen des Unternehmens (Kundens) nicht erfüllt und dadurch den Erfolg eines Projektes oder sogar des Unternehmens gefährdet. Aus diesem Grund müssen die Dienstgüteeanforderungen eines Dienstes so genau wie möglich auf die Dienstgüteparameter und Metriken in den internen operativen Vereinbarungen und den externen Lieferantenverträgen abgebildet werden. Darüber hinaus zeigt Band 1 die Bedeutung und Vorteile für Kunden, Dienstanbieter sowie Zulieferer von Hard- und Software auf, die mit der Verhandlung von SLAs einhergehen.

Band 2: Concepts and Principles Band 2 der *SLA Management Handbook Solution Suite* definiert Ziel und Zweck von SLAs, die Voraussetzungen, unter denen sie angewendet werden, sowie ihre Beziehung zu anderen Dienst- und Geschäftsstrukturen. Anschließend werden drei Konzepte bzw. Verfahren genannt, mit deren Hilfe sich aus Sicht des TMF SLAs erstellen und managen lassen: der Dienstlebenszyklus, die Methodik zur *Key Quality Indicator-Identifikation* (KQI) und das *SLA Parameter Framework*.

SLA-Management orientiert am Dienstlebenszyklus Der zugrunde gelegte Dienstlebenszyklus entspricht in seinen wesentlichen Grundzügen dem in Kapitel 3.2.2 eingeführten Lebenszyklus nach Dreo. Lediglich eine Anpassungsphase ist im TMF-Lebenszyklus nicht explizit vorgesehen. Dafür gibt es eine Bewertungsphase, die der Betriebsphase nachgelagert ist und gerade aus SLM-Sicht durchaus sinnvoll ist. Zu jeder Phase dieses Lebenszyklus gehören nun bestimmte Betriebsverfahren aus der *Enhanced Telecom Operations Map* (eTOM) (vgl. Abschnitt 4.3.3), die für das Management von SLAs instrumentiert werden können.

Methode zur KQI-Identifikation Mit diesem Verfahren werden die Anforderungen des Kunden bestimmt, also seine spezifischen Erwartungen an die Dienstqualität.

Das SLA Parameter Framework Diese Methode kann verwendet werden, um Leistungsparameter sinnvoll einzuordnen. Das Framework teilt SLA-Parameter in sechs Kategorien ein, je nach Dienst und verwendeter Netztechnologie, Messungen aus Stichproben und durchschnittlicher Performanz. Hat man die Leistungsparameter ermittelt, müssen ihnen auch noch Werte bzw. Wertebereiche zugeordnet werden. Dies ist Teil einer genau festgelegten Vertragsverhandlung, die allerdings nicht mehr Bestandteil von Band 2 ist.

Band 3: Service and Technology Examples Band 3 der *SLA Management Handbook Solution Suite* vertieft noch einmal die wesentlichen Bestandteile der Prinzipien und Richtlinien aus Band 2. Darauf aufbauend wird eine Checkliste vorgestellt, die zahlreiche Punkte enthält, die in ein SLA aufgenommen werden können. Zu den behandelten Themen gehören Dienstbeschreibungen, Angaben zur (vereinbarten) Dienstgüte, Dienstkontrolle und Berichterstattung sowie Preis- und Abrechnungsinformationen. Darüberhinaus werden sieben mögliche Anwendungsszenarien vorgestellt, in denen gezeigt wird, wie die Prinzipien und Richtlinien aus Band 2 angewendet werden können.

Empfehlungen zum SLA-Entwurf Im zweiten Band der SLA Handbook Solution Suite werden 23 Empfehlungen zum SLA-Entwurf beschrieben. Die Intention dahinter ist ähnlich wie die von ISO/IEC 20000, Teil 2 (*Code of Practice*) [31]. Allerdings sind die Empfehlungen im Handbuch des TMF deutlich ausführlicher beschrieben, da das gesamte Dokument ausschließlich auf Aspekte des SLA-Managements fokussiert ist. Insofern macht es Sinn, diese Empfehlungen in Ergänzung des Teils 2 von ISO/IEC 20000 zu betrachten. Die Empfehlungen sind zu vier Kategorien zusammengefasst: Erfüllung (*Fulfillment*), Erbringung und Gewährleistung (*Assurance*), Management der Schnittstelle zum Kunden (*Customer Interface Management*) und allgemeine Anforderungen.

I. Erfüllung (*Fulfillment*)

1. Empfehlung: Ein SLA sollte klare und eindeutige Definitionen der folgenden Punkte enthalten: Messbare Leistungsparameter, Messtechnik und Messdauer, Verantwortlichkeiten von Kunde und Provider, Verfahren für den Fall, dass SLAs nicht eingehalten werden, Rahmenbedingungen der Verfügbarkeits- und Support-Zusagen, Berichtsarten, Dienstdefinitionen, anwendbares Recht und Schlichtungsverfahren.
2. Empfehlung: Für jeden angebotenen Dienst sollte der Kunde zugesagte Parameter und Wertebereiche dieser Parameter auswählen können.
3. Empfehlung: Bei der Definition der Leistungsparameter eines Dienstes in einem SLA sollten folgende Punkte beachtet werden: Die Parameter sollten die Leistung für Kunde und Provider verständlich beschreiben, Messungen und zugehörige Messumgebungen sind genau festzulegen, Messungen können vom Kunden, vom Provider oder von einer unabhängigen dritten Partei durchgeführt werden können, und Messergebnisse sollten zur Fehlersuche und -eingrenzung geeignet sein.
4. Empfehlung: Alle definierten Ziele sollten periodisch überprüft und gegebenenfalls angepasst werden.
5. Empfehlung: Der Provider sollte gewährleisten, dass Schwellenwerte definiert werden, deren Über- bzw. Unterschreitung vorbeugende Maßnahmen auslöst, um SLA-Verstöße rechtzeitig zu verhindern. Für die Soll-Werte der Leistungsparameter sollten verschiedene Optionen zur Auswahl stehen, die es ermöglichen, beispielsweise an bestimmten Wochentagen oder zu bestimmten Tageszeiten spezielle

Soll-Werte zuzuweisen. Ein Monitoring-System sollte bereitstehen, um Messinformationen abrufen zu können.

II. Erbringung und Gewährleistung (*Assurance*)

6. Empfehlung: Der Provider sollte die tatsächlich erbrachte Leistung im Zusammenhang seiner IT-Dienste mit den Zielen aus SLAs vergleichen.
7. Empfehlung: Genaue und kundenorientiert aufbereitete Informationen über SLA-Parameter sollten den Kunden in Echtzeit oder in regelmäßigen Abständen zur Verfügung gestellt werden.
8. Empfehlung: Strenge Zugriffskontrollen und Authentifizierung werden vorausgesetzt, damit der Kunde in dem Umfang auf seine Daten zugreifen kann, der in den SLAs festgelegt wurde.
9. Empfehlung: Der Provider sollte jede Verschlechterung der Leistung eines Dienstes ermitteln. Stellt eine Verschlechterung eine für einen Kunden maßgebliche Beeinträchtigung dar, so ist der Kunde darüber zu informieren, und es müssen Gegenmaßnahmen eingeleitet werden.
10. Empfehlung: Der Provider sollte über ein Überwachungssystem verfügen, um Fehler erkennen, zurückverfolgen und beheben zu können. Dies muss innerhalb der in den SLAs festgelegten Zeiten geschehen. Können Fehler nicht wie vorgesehen behoben werden, müssen betroffene Kunden darüber informiert werden.
11. Empfehlung: Kunden sollten über mögliche Abweichungen von den in den SLAs formulierten Zielen informiert werden. Dazu gehören auch Dienstausfälle in Folge geplanter oder außerplanmäßiger Wartungsarbeiten. In solchen Fällen sollte der Kunde über den voraussichtlichen Zeitpunkt und den Fortschritt der Wiederherstellung informiert werden.
12. Empfehlung: Für jeden Leistungsparameter sollte der Provider „weiche Schwellwerte“ festlegen, um eventuelle Schwierigkeiten frühzeitig zu erkennen. Der Kunde sollte im in den betreffenden SLAs festgelegten Umfang über jeden Leistungseinbruch informiert werden, der zu einem Verstoß gegen die in den SLAs festgelegten Ziele führen könnte.
13. Empfehlung: Bei einem Dienst, an dem mehrere Provider beteiligt sind, ist der Provider, der dem Kunden unmittelbar gegenübersteht, dafür verantwortlich, Einbrüche in der Leistung dieses Dienstes an

den Kunden zu melden. Dies gilt auch, wenn ein anderer Provider oder Netzbetreiber für diese Einbrüche verantwortlich ist.

III. Management der Schnittstelle zum Kunden (*Customer Interface Management*)

14. Empfehlung: Kunden sollten die Möglichkeit haben, Schwierigkeiten, Fehler oder Änderungswünsche an SLAs zu melden.
15. Empfehlung: Kundenanfragen bezüglich der Dienstqualität sollten umgehend beantwortet werden.
16. Empfehlung: Der Provider sollte Informationen über jeden Dienst vorrätig haben, den ein Kunde beauftragen könnte.

IV. Allgemeine Anforderungen

17. Empfehlung: SLAs sollen eindeutig identifizierbar sein und mit Seitenzahl und Datum versehen werden.
18. Empfehlung: Wenn möglich, sollten SLAs modular aufgebaut sein, um Zusammenhang und Abgrenzung allgemeiner und dienstbezogener Abschnitte zu verdeutlichen.
19. Empfehlung: Jedes Dienst-Element¹ soll separat definiert und eindeutig bezeichnet sein, damit im Leistungsbericht die Bezeichnungen aus den SLAs verwendet werden können.
20. Empfehlung: Die Verantwortlichkeiten des Providers gegenüber einem Kunden sollten genau festgelegt sein. Das schließt auch dienst- oder leistungsspezifische Ausschlüsse, wie beispielsweise angekündigte Wartungsintervalle, durch den Kunden verursachte Dienstausfälle und Ausfälle und Folgeschäden durch höhere Gewalt mit ein.
21. Empfehlung: Die Verantwortlichkeiten jedes Kunden gegenüber dem Provider sollten genau festgelegt sein. Dazu gehört beispielsweise die Festlegung der bevorzugten Art und Weise, Fehler an den Provider zu melden.
22. Empfehlung: Vertragsstrafen, Rechtsmittel und Wiederherstellungszeiträume sollten festgelegt werden.

¹Der Begriff des Dienst-Elements (*service element*) entspringt der *Enhanced Telecom Operations Map* (eTOM) und hat seinen Ursprung in der Betrachtung reiner Telekommunikationsdienste.

23. Empfehlung: Begriffe, die in den Dienst- und Leistungsspezifikationen verwendet werden, sollten in den SLAs definiert werden. Zu diesen Begriffen gehören beispielsweise Dienstzugangspunkt, Dienstverfügbarkeit oder Dienstunterbrechung.

4.5.1.2 Bewertung

So umfassend die *SLA Management Handbook Solution Suite* auch ist, so bleibt sie in vielen Punkten doch recht unkonkret. Automatisierungsaspekte spielen – wenn überhaupt – nur am Rande eine Rolle. Hilfreiche Eingaben für diese Arbeit liefert vor allem der Band 2 mit den beschriebenen Konzepten und Prinzipien. Besonders die vorgestellten Empfehlungen zum SLA-Management können zum Entwurf einzelner architektureller Modellelemente insbesondere im Kontext von Informations- und Funktionsmodell beitragen. Beispielsweise liefert Empfehlung 1 wichtige Aspekte für den Teil des Informationsmodells der Prozess-Sicht, der sich mit der Beschreibung und Modellierung der Attribute von Dienstvereinbarungen beschäftigt. Tabelle 4.13 zeigt, für welche Teile der Realisierung des der Managementarchitektur zugrunde liegenden Funktionsspektrums das SLA-Management-Handbuch lösungsbezogene Beiträge liefern kann.

In einem Forschungsartikel aus dem Jahr 2007 mit dem Titel *Frameworks for Business-driven Service Level Management – A Criteria-based Comparison of ITIL and NGOSS* [83] wurden das SLA-Management-Handbuch und die *IT Infrastructure Library* (in der Version 2) hinsichtlich ihrer Eignung als Rahmenwerk für ein geschäftsorientiertes Service-Level-Management untersucht. Ein wesentliches Ergebnis des Artikels ist eine Empfehlung zur Kopplung und Integration der beiden Ansätze.

4.5.2 SLA-Entwurf auf Basis von Dienstprozessen (Schmidt)

Das Ziel der Arbeit von Schmidt [14] besteht darin, den Entwurf von Dienstvereinbarungen an den Erfordernissen von Dienstmanagementprozessen auszurichten.

4.5.2.1 Überblick

Dahinter verbirgt sich eine einfache, aber realitätsnahe Annahme: Dienstvereinbarungen bestimmen und steuern in gewisser Weise die Ausführung von

<i>Anforderung</i>	<i>Erfüllungsgrad</i>	
	Prozess	System
F1 (Dienstinf.)	□□□	□□□
F2 (Kunden)	■□□	□□□
F3 (FA/Lief.)	■□□	□□□
F4 (Dienstkat.)	■ ■ □	□□□
F5 (Supp.-Level)	■□□	□□□
F6 (SLAs)	■ ■ □	□□□
F7 (OLAs)	■ ■ □	□□□
F8 (UCs)	■ ■ □	□□□
F9 (Konflikte)	□□□	□□□
F10 (Messung)	■ ■ □	□□□
F11 (Berichte)	■ ■ □	□□□
F12 (Entstörung)	■□□	□□□
F13 (Verrechnung)	□□□	□□□
F14 (Transition)	□□□	□□□
F15 (Sicherheit)	□□□	□□□

Tabelle 4.13: Bewertung des Lösungspotenzials des SLA-Management-Handbuchs im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

Dienstmanagementprozessen. So hängt beispielsweise die zeitliche Zielvorgabe bei der Behebung von Störungen an einem Dienst (wie lange darf die Wiederherstellung maximal dauern?) von den in den SLAs zu diesem Dienst zugesicherten Garantien hinsichtlich Verfügbarkeit und maximaler Wiederherstellungsdauer ab. Während die meisten Ansätze im Dienstmanagement davon ausgehen, dass Dienstvereinbarungen alle relevanten Parameter spezifizieren, die Einfluss auf die Ausführung von Managementprozessen haben, geht Schmidt vereinfacht gesagt den umgekehrten Weg: Er untersucht Managementprozesse auf sinnvolle Steuergrößen und leitet daraus determinierende SLA-Parameter ab.

4.5.2.2 Bewertung

Schmidt beschäftigt sich dediziert mit dem Entwurf von Dienstvereinbarungen und ihren Implikationen auf Dienstmanagementprozesse. Entsprechend liefert er mit seinen Ergebnissen Beiträge zur Realisierung der Anforderungen F6 sowie F12 bis F15.

Anforderung	Erfüllungsgrad	
	Prozess	System
F1 (Dienstinf.)	□□□	□□□
F2 (Kunden)	□□□	□□□
F3 (FA/Lief.)	□□□	□□□
F4 (Dienstkat.)	□□□	□□□
F5 (Supp.-Level)	□□□	□□□
F6 (SLAs)	■ ■ □	□□□
F7 (OLAs)	□□□	□□□
F8 (UCs)	□□□	□□□
F9 (Konflikte)	□□□	□□□
F10 (Messung)	□□□	□□□
F11 (Berichte)	□□□	□□□
F12 (Entstörung)	■ ■ □	□□□
F13 (Verrechnung)	■ ■ □	□□□
F14 (Transition)	■ ■ □	□□□
F15 (Sicherheit)	■ ■ □	□□□

Tabelle 4.14: Bewertung des Lösungspotenzials des Ansatzes von Schmidt im Hinblick auf die funktionalen Zielsetzungen dieser Arbeit

4.5.3 Weitere Veröffentlichungen im Kurzüberblick

Der Vollständigkeit halber seien noch einige weitere Veröffentlichungen zum Themengebiet Service-Level-Management genannt, die zwar im spezifischen Kontext dieser Arbeit nicht von Relevanz sind, für eventuelle weiterführende Arbeiten oder zur Komplettierung des Bildes der SLM-Landschaft aber erwähnenswert sind:

- Das Buch *Foundations of Service Level Management* [84] von Sturm et al. stellt eine Alternative zum Buch von Lewis dar, um einen ersten Überblick über Zielsetzungen und Aufgabenbereiche des Service-Level-Managements zu erhalten.
- Eine Auswahl von Fachbeiträgen, Analysen und Interviews zum Thema Service-Level-Management liefern Bernhard et al. mit ihrem Buch *Service-Level-Management in der IT – Wie man erfolgskritische Leistungen definiert und steuert* [85].
- Hiles liefert mit *The Complete Guide to IT Service Level Agreements* [86]

einen Leitfaden zum Thema Dienstvereinbarungen, der sehr beispielorientiert vorgeht.

- Tyurin untersucht in seiner Arbeit [87] den *Aufbau und Zusammenhang der drei Service Level Management Vertragstypen SLA, OLA, UC*.
- Eine Fallstudie zum Thema Service-Level-Management liefert Hassenpflug mit seiner Diplomarbeit *Analyse von Service Level Management-Prozessen in der BMW Group* [88].

4.6 Zusammenfassung und abschließende Bewertung

In diesem Kapitel, das den *State of the art* auf dem Gebiet des Service-Level-Managements reflektiert, wurde vor allem darauf geachtet, die vorgestellten Veröffentlichungen im Kontext der vorliegenden Arbeit und im Hinblick auf die im nächsten Kapitel beginnende Entwicklung einer Managementarchitektur gemäß der Methodik aus Kapitel 2 und den Anforderungen aus Kapitel 3 darzustellen. Es wurde dargestellt und begründet, welche Veröffentlichungen zur Begriffsbildung herangezogen werden, welche einen Einfluss auf das methodische Vorgehen ausüben, und welche zur Lösungsentwicklung beitragen.

Kapitel 5

Prozess-Sicht der Managementarchitektur

Inhalt dieses Kapitels

5.1	Organisationsmodell	170
5.1.1	Organisationsdomänen	170
5.1.2	Rollen	171
5.1.3	Verantwortungsmodelle	174
5.1.4	Zusammenfassung	176
5.2	Informationsmodell	177
5.2.1	Grundlagen der Prozess-Informationsmodellierung	177
5.2.2	Identifikation SLM-spezifischer Konzepte	181
5.2.3	Konzepterweiterungen	186
5.2.4	Ableitung und Klassifizierung von Informations- artefakten	192
5.2.5	Zusammenfassung	214
5.3	Funktionsmodell	216
5.3.1	Festlegung der Funktionsbereiche	216
5.3.2	Funktionsbereich I: Dienstinformationsmanagement	221
5.3.3	Funktionsbereich II: Stakeholder-Management . . .	237
5.3.4	Funktionsbereich III: Vereinbarungsmanagement .	242
5.3.5	Funktionsbereich IV: Messdaten- und Berichtsma- nagement	256
5.3.6	Zusammenfassung	264
5.4	Kommunikationsmodell	266
5.4.1	Intraprozess-Kommunikationsmodell	267
5.4.2	Interprozess-Kommunikationsmodell	282
5.4.3	Zusammenfassung	294
5.5	Zusammenfassung und nächste Schritte	296
5.5.1	Ergebnisse der Prozess-Sicht	296

Ziel dieses Kapitels ist es, Modelle der Prozess-Sicht zu entwickeln, die die Primäranwendungsfälle aus der Anforderungsanalyse vollständig abdecken. Entsprechend dem Ordnungs- und Entwicklungsrahmen orientiert sich das Kapitel im Aufbau an den vier Teilmodellen einer Managementarchitektur.

Abbildung 5.1 zeigt, dass dieses Kapitel den ersten der zwei Kernbestandteile der Managementarchitektur und damit Ausgangspunkt für den zweiten Schritt, den Entwurf der Modelle der System-Sicht, bildet.

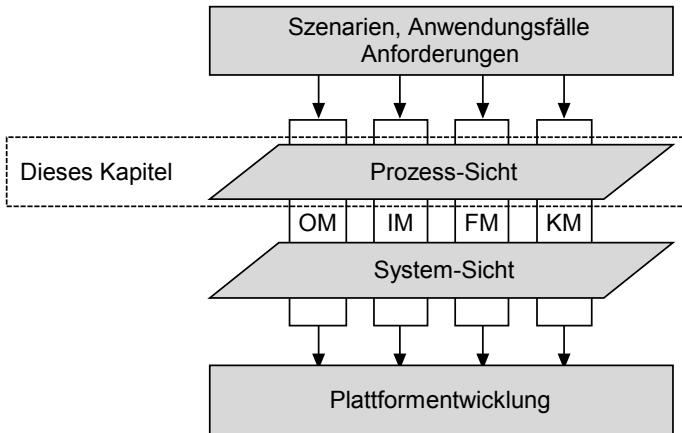


Abbildung 5.1: Einordnung dieses Kapitels in den Ordnungsrahmen

Es folgt also, gestützt auf den Anforderungen, die Darstellung des Organisations-, Informations-, Funktions- und Kommunikationsmodells auf der Prozessebene. Damit stellt dieses Kapitel eine Verbindung her zwischen den Anwendungsfällen aus der szenariobasierten Anforderungsanalyse und Entwurfsaspekten eines IT-gestützten Managementsystems – und zwar unter Berücksichtigung prozessorientierter Aspekte. Diese sind im Rahmen des Organisationsmodells im Wesentlichen Organisationsdomänen, Rollen und Verantwortlichkeiten, im Rahmen des Informationsmodells die benötigten Informationsartefakte und Anforderungen an ihre Modellierung, im Rahmen des Funktionsmodells Funktionsbereiche und zugehörige Kontroll- und Objektflüsse, gegeben durch konkrete Referenz-Teilprozessabläufe (Workflows) , sowie im

Rahmen des Kommunikationsmodells die Darstellung der Kommunikationsbeziehungen innerhalb des Service-Level-Managements und die Einordnung der SLM-spezifischen Aktivitäten in den Gesamtkomplex Dienstmanagement.

5.1 Organisationsmodell

Fragestellungen dieses Moduls: Die Fragen, die durch das Organisationsmodell der Prozessebene in erster Linie beantwortet werden müssen, lauten:

- Welche Organisationsdomänen spielen im Service-Level-Management eine Rolle?
- Welche SLM-spezifischen Rollen können innerhalb dieser Domänen identifiziert werden?
- Auf welche Arten können Rollen in Managementprozesse eingebunden sein? Welche verschiedenen Verantwortungsmodelle sind also denkbar?

Das Wissen hierüber wird sowohl für die Entwicklung der übrigen Architekturteilmodelle der Prozess-Sicht als auch für den Entwurf der Modelle der System-Sicht im Rahmen der Managementarchitektur benötigt. Speziell in die Referenz-Workflows des Funktionsmodells fließen Rollen- und Verantwortungsmodelle ein.

5.1.1 Organisationsdomänen

Aus den der Anforderungsanalyse zugrunde gelegten Szenarien und Anwendungsfällen lassen sich – in Übereinstimmung mit der gängigen *Best Practice*, wie im vorangegangenen Kapitel beschrieben – die folgenden Organisationsdomänen ableiten, in denen die verschiedenen Rollen „beheimatet“ sind.

IT-Provider Der Provider (Dienstanbieter) ist die Organisation oder Abteilung einschließlich aller ihr (intern) angeschlossenen Anlagen und Personen, die die Planung, Bereitstellung und operative Unterstützung von IT-Diensten für einen oder mehrere Kunden durchführt und verantwortet. Im Zusammenhang mit einem IT-gestützten Managementsystem für Service-Level-Management ist der Provider nicht nur der hauptsächliche Nutznießer eines solchen, sondern auch verantwortlich für dessen Installation, Etablierung und Wartung sowie damit verbundene Schulungsaktivitäten auch über die Grenzen der eigenen Organisationsdomäne hinweg.

Kunde Ein Kunde (Dienstnehmer, *Customer*) ist jede Organisation oder Unternehmensfachabteilung, der IT-Dienste vom Provider bereitgestellt werden. Die Orientierung des Providers an den Erfordernissen und Wünschen der Kunden hinsichtlich der IT-Dienstleistungen ist insbesondere im Hinblick auf die Ausrichtung der IT an den Geschäftsprozessen von entscheidender Wichtigkeit. Service-Level-Management ist diejenige Disziplin, die die damit verbundenen konkreten Ziele unter anderem durch entsprechende Dienstvereinbarungen formalisiert. Im Umkehrschluss ist daher die Berücksichtigung bestimmter Rollen aus der Kundendomäne für die IT-Unterstützung des Service-Level-Managements essenziell, wie auch die Anforderungsanalyse nachgewiesen hat.

Externer Zulieferer Ein externer Zulieferer (*External Supplier*) ist eine aus wirtschaftlicher (und juristischer) Sicht nicht zum Dienstanbieter gehörende Organisation, die den Provider mit (Sub-)Diensten und anderen Mitteln versorgt, die dieser zur Erbringung seiner Dienste benötigt. Aus Sicht des Zulieferers tritt ein beliefertes Provider als Kunde auf.

5.1.2 Rollen

Bevor Rollen innerhalb dieser Domänen beschrieben werden, folgen einige kurze Erläuterungen zum Rollenbegriff, der allgemein wie folgt beschrieben werden kann: *Eine Rolle definiert im Rahmen von Prozessen Aufgaben, Kompetenzen und Verantwortungen eines direkt Beteiligten.*

Im Hinblick auf Softwaresysteme ist folgender Rollenbegriff relevant: *Eine Benutzerrolle definiert Aufgaben, Eigenschaften und vor allem Rechte eines Benutzers in einer Software bzw. in einem (Betriebs-)System. Statt Benutzern Rechte direkt zuzuweisen, wird eine Benutzerrolle definiert, die dann vielen Benutzern zugeordnet werden kann. Dies erleichtert die Rechteverwaltung des Softwaresystems, da insbesondere bei Änderungen der Rechtestruktur nur die Rechte der Benutzerrolle angepasst werden müssen.*

Beide Definitionen beschreiben einen für diese Arbeit zutreffenden Rollenbegriff, und während in diesem Kapitel die erste, eher prozessbezogene Definition zutreffender ist, so wird die systembezogene Rolle Anwendung finden, sobald auf Basis der Modelle der Prozess-Sicht die Modelle der System-Sicht entwickelt werden.

Allgemein ist noch zu sagen, dass Rollen nicht gleichzusetzen sind mit Personalstellen. Das würde dem Rollenkonzept widersprechen. Vielmehr können

mehrere Rollen von einer einzigen Person oder eine einzige Rolle von mehreren Personen ausgefüllt werden.

5.1.2.1 Rollen der Dienstanbieter-Domäne

Das am stärksten diversifizierte Rollenmodell aus Sicht des Service-Level-Managements ist das der Provider-Domäne. Die Anforderungsanalyse liefert Anhaltspunkte für die folgenden Rollen:

Service-Level-Manager Der Service-Level-Manager ist operativ verantwortlich für den Gesamtprozess Service-Level-Management. Er vertritt einerseits den Provider gegenüber allen Kunden und andererseits die Kundeninteressen innerhalb der Provider-Organisation. Der Abschluss von Dienstvereinbarungen untersteht seiner Verantwortung.

Dienstportfolioverwalter Der Dienstportfolioverwalter (*Service Portfolio Supervisor*) ist verantwortlich für die Erstellung und Pflege des Dienstportfolios im Rahmen des Service-Level-Managements. Er berichtet an den Service-Level-Manager, sofern die beiden Rollen nicht in Personalunion ausgefüllt werden.

Kontraktverwalter Der Kontraktverwalter (*Contract Supervisor*) ist verantwortlich für die Erstellung und Pflege (allerdings nicht für die Verhandlung und den Abschluss) aller Dienstvereinbarungen (SLAs), operativen Vereinbarungen (OLAs) und Lieferantenverträge (UCs). Er berichtet an den Service-Level-Manager, sofern die beiden Rollen nicht in Personalunion ausgefüllt werden.

Andere Prozessmanager Für Prozesse außerhalb der Disziplin des Service-Level-Managements – also zum Beispiel *Change Management*, *Incident Management*, *Capacity Management* oder *Budgeting and Accounting* – die im Rahmen der Kommunikationsaspekte der Managementarchitektur von Relevanz sind, werden die Rollen der jeweiligen Prozessmanager betrachtet. In diesem Punkt orientiert sich die Arbeit an der im Prozessmanagement allgemeinen Empfehlung, dass jeder Prozess durch einen Manager operativ verantwortet wird und für andere Prozessmanager als Ansprechpartner fungiert. Die Frage, welche diese „anderen“ Prozesse genau sind, wird im Interprozess-Kommunikationsmodell der Prozess-Sicht untersucht und durch ein Referenz-Prozessmodell beantwortet.

Vertreter einer internen Dienstbringerpartei Neben den anderen Prozessen sind vor allem die an der Dienstbringung beteiligten internen Parteien, meistens gegeben durch die IT-Fachgruppen (zum Beispiel Abteilung Netzbetrieb, Abteilung Serverbetrieb, Abteilung Applikationsentwicklung, etc.) aus SLM-Sicht relevant. Der Vertreter einer solchen Dienstbringerpartei (*Internal Party Representative*) ist für den Service-Level-Manager der Partner bei der Aushandlung operativer Vereinbarungen.

Bemerkung: Die beiden Verwalter-Rollen (im Englischen: *Supervisor*) übernehmen Aufgaben im Zusammenhang mit der Administration und Dokumentation von Diensten im Rahmen des Dienstportfolios bzw. Verträgen und Vereinbarungen im Rahmen des Vereinbarungsmanagements. Eine treffende Bezeichnung dieser Rollen vor diesem Hintergrund wäre auch *Bibliothekar*.

5.1.2.2 Rollen der Dienstnehmer-Domäne

Kundenvertreter Aus Provider-Sicht verläuft sämtliche Kundenkommunikation nur über den oder die Kundenvertreter, womit diese Rolle die einzige (menschliche) Schnittstelle zum Kunden darstellt. Der Kundenvertreter (*Customer Representative*) ist autorisiert, Dienstvereinbarungen mit dem Provider abzuschließen und ist Quelle für kundenspezifische Anforderungen an einen gelieferten oder angefragten Dienst.

Dienstanwender Jede Person, die einen IT-Dienst nutzt, tritt als Dienstanwender (*User*) auf. Aus der Perspektive des Service-Level-Managements bestehen keine direkten Kontaktpunkte zu den Anwendern, da jegliche Kommunikation zwischen Provider- und Kundenseite (vgl. MNM-Dienstmodell) über den Kundenvertreter abläuft. Dennoch komplettiert der Dienstanwender als Rolle das SLM-Organisationsmodell und muss als solche auch berücksichtigt werden, da sie speziell im Informationsmodell Relevanz hat. Beispielsweise können Dienstvereinbarungen Informationen enthalten, die im Zusammenhang mit der Anwenderrolle stehen (etwa eine Eingrenzung der Anwender aus der betrachteten Kundendomäne, für die die Dienstvereinbarung gilt, oder die Anzahl der Anwender eines Dienstes als wichtiger Parameter der Dienstbringung).

5.1.2.3 Rolle der Zulieferer-Domäne

In der Zulieferer-Domäne ist nur eine Rolle – die des Lieferantenvertreters – von Bedeutung und Interesse. Wie ein Subdienst vom Zulieferer realisiert wird und welche Prozesse hierfür auf Lieferantenseite etabliert sind, ist für den Provider weder allgemein, noch spezifisch aus Sicht des Service-Level-Managements maßgeblich.

Lieferantenvertreter Aus Provider-Sicht verläuft sämtliche Lieferantenkommunikation nur über den oder die Lieferantenvertreter (*Supplier Representative*), womit diese Rolle die einzige (menschliche) Schnittstelle zum Lieferanten darstellt. Der Lieferantenvertreter steht in enger Analogie zur Rolle des Kundenvertreters in der Kundendomäne. Er ist autorisiert, Lieferantenverträge (UCs) mit dem Provider abzuschließen.

Mit diesem einfachen Organisationsmodell, das die drei bereits der Anforderungsanalyse zugrunde gelegten Domänen verfeinert, ist der Grundstein für die weiteren Teilmodelle gelegt. Abbildung 5.2 fasst die Ergebnisse bis hierhin überblicksartig zusammen.

5.1.3 Verantwortungsmodelle

Im Zusammenhang mit der Planung und Ausführung von Managementprozessen werden in der Regel nicht konkreten Prozessbeteiligten (Menschen), sondern Rollen bestimmte Verantwortungen im Rahmen der Prozessaktivitäten zugewiesen. Hierin liegt begründet, warum ein differenziertes Domänen- und Rollenmodell wie das zuvor vorgestellte für den Entwurf eines prozessorientierten Managementsystems so wichtig ist. Damit im Rahmen des Funktionsmodells der Managementarchitektur Rollen und ihre Verantwortlichkeiten dargestellt werden können, werden nun noch konkrete Verantwortungsmodelle definiert, die in diesem Kontext berücksichtigt werden.

Die Verantwortungsmodelle legen fest, welche grundsätzlichen Varianten für die Einbindung von Rollen im Rahmen von Prozessaktivitäten bestehen. Die Begrifflichkeiten und Definitionen stützen sich auf und erweitern das Konzept der *RACI-Diagramme* wie es unter anderem durch die *IT Infrastructure Library* [2] beschrieben wird. Folgende Verantwortungsmodelle sind demnach denkbar:

- Operativ verantwortlich (*responsible*): Ein Akteur ist in seiner Rolle zuständig für die konkrete Durchführung einer Aktivität gemäß der für diese Aktivität gegebenenfalls definierten Verfahrensbeschreibung.

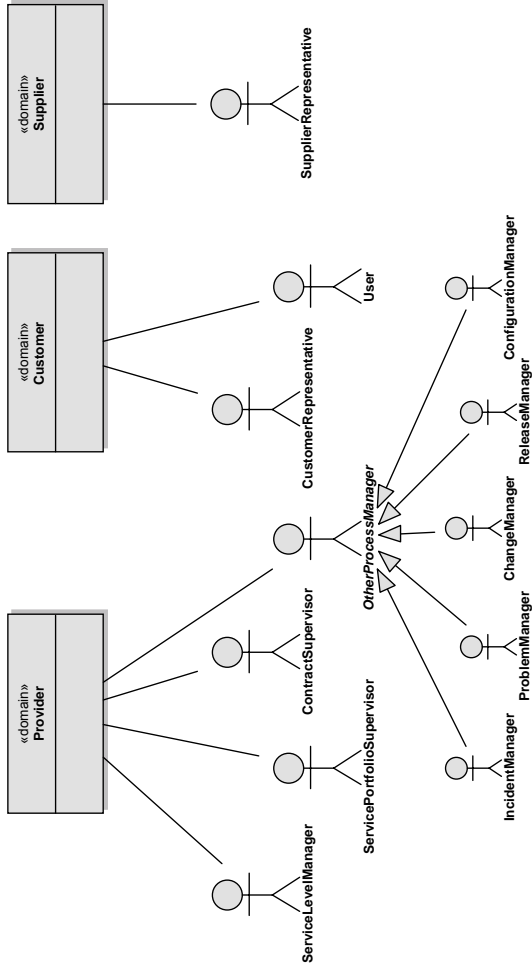


Abbildung 5.2: Domänen- und Rollenmodell im Überblick

- Rechenschaftspflichtig (*accountable*): Ein Akteur ist in seiner Rolle rechenschaftspflichtig für eine Aktivität und ihre Durchführung, obwohl nicht operativ für die Durchführung zuständig.
- Informiert (*informed*): Ein Akteur wird in seiner Rolle über die Durchführung, Zwischen- und/oder Endergebnisse einer Aktivität informiert, ohne selbst operativ verantwortlich zu sein.
- Unterstützend (*supportive*): Ein Akteur liefert in seiner Rolle erforderliche Eingaben (in der Regel in Form von Informationen), die zur Durchführung einer Aktivität benötigt werden, also obligatorisch sind.
- Beratend hinzugezogen (*consulted*): Ein Akteur wird während der Durchführung einer Aktivität, falls erforderlich, konsultiert. Im Unterschied zum Verantwortungsmodell *unterstützend* werden die Eingaben und Informationen eines beratend hinzugezogenen Akteurs nicht grundsätzlich bei jeder Instanz der betreffenden Aktivität benötigt, sondern vielfach nur in definierten Sonderfällen oder bei Schwierigkeiten bei der verfahrenskonformen Durchführung.
- Verifizierend (*verifying*): Ein Akteur verifiziert in seiner Rolle Zwischen- oder Endergebnisse einer Aktivität, bevor diese als abgeschlossen angesehen werden und der Prozessablauf mit der nachfolgenden Aktivität fortgesetzt wird.

5.1.4 Zusammenfassung

Das Organisationsmodell der Prozess-Sicht bildet den ersten Baustein des Prozessmodells der Managementarchitektur und definiert organisatorische Domänen, Rollen und Verantwortungsmodelle, die die Grundlage für ein prozessorientiertes Service-Level-Management darstellen. Damit wurde der organisationstheoretische Rahmen definiert, in den sich die weiteren Teilmodelle einordnen. Besonders in die Entwicklung des Funktionsmodells fließen Rollendefinitionen und Verantwortungsmodelle ein.

5.2 Informationsmodell

Fragestellungen dieses Moduls: Das Informationsmodell der Prozess-Sicht soll Antworten auf die folgenden Fragen geben:

- Welche Informationen müssen im Rahmen des Service-Level-Managements und seiner Aktivitäten verarbeitet und verwaltet werden?
- In welchen Informationsartefakten werden diese Informationen zweckmäßigerweise gebündelt?
- Welche konkreten Informationsanforderungen bestehen im Hinblick auf diese Artefakte? Welche Klassen von Informationsanforderungen lassen sich also artefaktspezifisch ermitteln?
- Welches sind die übergeordneten Zusammenhänge, Abhängigkeiten und Multiplizitäten zwischen diesen Informationsartefakten?

5.2.1 Grundlagen der Prozess-Informationsmodellierung

Informationen sind eine essenzielle Ressource in jedem Prozess. Insbesondere in Dienstmanagementprozessen liegen Eingaben (*Inputs*) und Ergebnisse (*Outputs*) nahezu immer in Form von Informationen vor. Daher ist es im Rahmen einer Managementarchitektur für Service-Level-Management unerlässlich, die konkreten Informationsanforderungen zu analysieren.

5.2.1.1 Terminologie

Einleitend dazu einige Begriffsdefinitionen, teils in Anlehnung an ISO/IEC 20000 [52] und Scherer [89]:

Dokument Ein Dokument (*Document*) bezeichnet allgemein eine Sammlung von Informationen und ihr unterstützendes Medium.

Informationsartefakt Ein Informationsartefakt (*Information Artifact*) (Synonym: Informationsobjekt, Prozessartefakt) ist ein reales (physisches) oder virtuelles (logisches) Dokument, welches Informationen in einem

übergeordneten Kontext in festgelegter und strukturierter Weise speichert und zur Intra- oder Interprozesskommunikation verwendet wird. Mittels Informationsartefakten werden Informationsflüsse im Kontext von Managementprozessen realisiert – und zwar sowohl zwischen Aktivitäten ein- und desselben Prozesses als auch über Prozessgrenzen hinweg. In dieser Eigenschaft grenzen sich Artefakte von Dokumenten ab: Jedes Informationsartefakt ist auch ein Dokument, jedoch nicht jedes Dokument notwendigerweise ein Informationsartefakt.

Beispiel: Eine Verfahrensbeschreibung einer Prozessaktivität stellt zwar ein Dokument, jedoch kein Informationsartefakt im Dienstmanagement dar. Die Verfahrensbeschreibung bestimmt zwar, wie eine Aktivität durchgeführt wird, ist nach deren Abschluss jedoch für weitere Aktivitäten oder andere Prozesse nicht von Bedeutung und somit nicht Teil eines Informationsflusses im Rahmen der betrachteten *Dienstmanagement*prozesse. Auf der Ebene des *Prozessmanagements* wäre eine Verfahrensbeschreibung hingegen ein Informationsartefakt, da sie in diesem Kontext klar zu den gemanagten Objekten (*Managed Objects*) zählt. Abbildung 5.3 illustriert diesen Zusammenhang.

Der Fokus des Informationsmodells der in dieser Arbeit entwickelten Architektur liegt auf den Informationsartefakten aus Dienstmanagement-Sicht. Vor dieser Zielsetzung beschäftigt sich dieses Teilkapitel mit Informationsartefakten, die speziell im Service-Level-Management relevant sind, indem es diese Artefakte auf Basis der Anforderungen und Primäranwendungsfälle identifiziert, beschreibt und ihre Zusammenhänge darstellt. In Anlehnung an Scherer [89] können Informationsartefakte unterschiedliche Ausprägungen annehmen, von denen nachfolgend einige für diese Arbeit relevante eingeführt und erläutert werden:

Datensatz Ein Datensatz (*Record*) ist ein Informationsartefakt, in dem geplante und durchgeführte Aktivitäten oder Ergebnisse dokumentiert werden und das somit als Nachweis für erfolgte Aktivitäten (zum Beispiel im Zusammenhang von Prozessabläufen) dienen kann. Beispiele für Datensatzartefakte im Dienstmanagement (außerhalb des Service-Level-Managements) sind *Trouble Tickets*, *Change Records* oder Meeting-Protokolle.

Dokumentation Im Unterschied zum Datensatz werden in einer Dokumentation (*Documentation*) Zielvereinbarungen, generelle Richtlinien oder

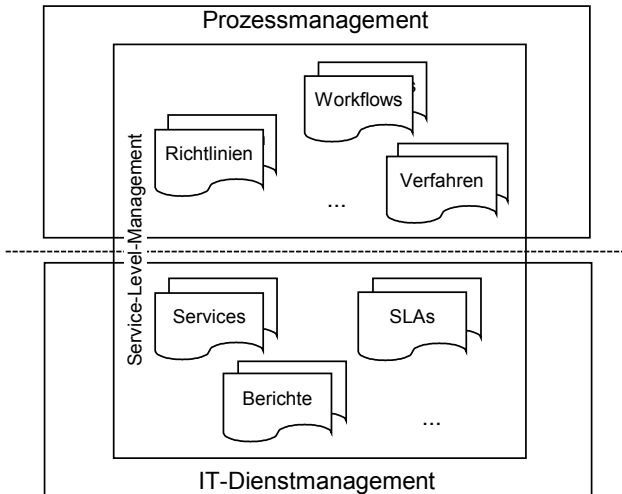


Abbildung 5.3: Abgrenzung zwischen Dienstmanagement und Prozessmanagement im Hinblick auf relevante Informationsartefakte

Verfahrensbeschreibungen festgehalten. Auch Verträge und Vereinbarungen, wie sie im Service-Level-Management eine bedeutende Rolle spielen, können im Kontext dieser Klassifikation als Dokumentationen aufgefasst werden; es werden Ziele und Zusicherungen zur Dienstbringung und -qualität dokumentiert.

Bericht Ein Bericht (*Report*) ist ein Informationsartefakt, in dem messbare Kenngrößen und gegebenenfalls ergänzende Erläuterungen adressatspezifisch aufbereitet und dokumentiert werden. In der Regel basiert ein Bericht auf einem oder mehreren Messdatensätzen. Berichtsanforderungen im Dienstmanagement spielen vor allem im Rahmen der Konformität mit ISO/IEC 20000 und der Umsetzung von *IT Governance* eine bedeutende Rolle.

Datenbank Eine Datenbank (*Database*) speichert mehrere Informationsartefakte gleichen oder unterschiedlichen Typs und ermöglicht effizienten Zugriff (Selektion), Sichtenbildung und klassische Datenbankoperationen auf diesen Artefakten sowie die Abbildung physischer oder logischer Beziehungen (Relationen) zwischen ihnen. Beispiele für typische Datenbankartefakte im Dienstmanagement sind die Konfigurationsma-

agementdatenbank (*Configuration Management Database*) und diverse Wissensdatenbanken (*Knowledge Databases*).

Datenbank-Sicht Eine Datenbank-Sicht (*Database View*) realisiert eine virtuelle Datenbank (oder Tabelle), die die Informationen aus einer oder mehreren Datenbanken oder anderer Datenbank-Sichten in festgelegter Weise darstellt.

Diese Liste dient der Verständnisbildung und der groben Einordnung. Sie erhebt keinen Anspruch auf Vollständigkeit, ist aber für die Zwecke dieser Arbeit hinreichend umfassend, um die Identifikation und Klassifizierung der Informationsartefakte im Service-Level-Management, die nachfolgend durchgeführt wird, zu unterstützen.

5.2.1.2 Vorgehen

Die Identifikation und Klassifizierung der im Service-Level-Management relevanten und notwendigen Informationsartefakte stützt sich – dem streng deduktiven Vorgehen der gesamten Arbeit folgend – auf den Ergebnissen der Anforderungsanalyse. Sie folgt zwei einfachen Schritten:

1. Schritt: Aus dem Funktionsspektrum und den Primäranwendungsfällen werden in erster Instanz alle *Konzepte identifiziert*, die im Service-Level-Management von zentraler Bedeutung sind. Ihr *Zweck* wird – wo möglich und sinnvoll in Anlehnung an die Kapitel 2 (Methodik und Begriffsbildung) und 3 (Anforderungsanalyse) – beschrieben, und ihre Zusammenhänge werden mit Hilfe eines UML-Klassendiagramms dargestellt. Als Konzepte werden in diesem Zusammenhang alle Domänen, Objekte und *Outputs* des Service-Level-Managements betrachtet, die sich direkt aus den funktionalen Anforderungen ergeben.
2. Schritt: Auf dieser Grundlage werden *Informationsartefakte gemäß der im vorherigen Abschnitt eingeführten Typen definiert*, die im Rahmen eines SLM-unterstützenden Managementsystems eine sinnvolle und zweckmäßige Unterstützung der SLM-Konzepte ermöglichen. Die Informationsartefakte verfeinern also das Konzeptmodell um konkrete Objekte, die zur Realisierung der Informationsflüsse im Kontext des Service-Level-Managements eingesetzt werden. Sie stellen vor allem aus Sicht des Funktionsmodells der Prozess-Sicht die eigentlichen gemanagten Objekte (*Managed Objects*) dar.

5.2.2 Identifikation SLM-spezifischer Konzepte

Einen ersten Zugang zu einem umfassenden Informationsmodell für Service-Level-Management bieten die Grundkonzepte, die im Mittelpunkt des Funktionsspektrums stehen, das die Anforderungsanalyse ergeben hat. Die nachfolgende Liste beschreibt auf Basis der funktionalen Anforderungen F1 bis F15 die zentralen SLM-spezifischen Konzepte, die dem Informationsmodell zugrunde liegen:

1. Dienst (*Service*): Funktionalität, die von einem Objekt an einer Schnittstelle angeboten wird (vgl. Kapitel 2.3.1, Dienstbegriff – systemtechnische Definition); Hilfsmittel für einen Kunden (vgl. Kapitel 2.3.1, Dienstbegriff – Wert-orientierte Definition); Abstraktion über Komponenten und Subdiensten (vgl. Kapitel 2.3.1, Dienstbegriff – Dienstmodell; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F1)
2. Kunde (*Customer*): Organisation oder Fachabteilung, der IT-Dienste vom Provider bereitgestellt werden (vgl. Abschnitt 5.1, Organisationsmodell der Prozess-Sicht; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F2)
3. Dienstbringerpartei (*Service Delivery Party*)
 - a) Interne Dienstbringerpartei (*Internal Delivery Party*): An der Dienstbringung/-realisierung beteiligte Provider-interne Partei (vgl. Kapitel 5.1, Organisationsmodell der Prozess-Sicht; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F3)
 - b) Externer Zulieferer (*External Supplier*): Nicht zum Dienstanbieter gehörende Organisation, die diesen mit (Sub-)Diensten und anderen Mitteln versorgt (vgl. Kapitel 5.1, Organisationsmodell der Prozess-Sicht; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F3)
4. Dienstkatalog (*Service Catalog*): Verzeichnis aller durch einen Kunden beauftragbaren IT-Dienste, realisiert als kundenspezifische Sicht auf das Dienstportfolio (vgl. Kapitel 2.3.4, Begriffe auf der Schnittstelle zwischen Kunde und Provider; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F4)

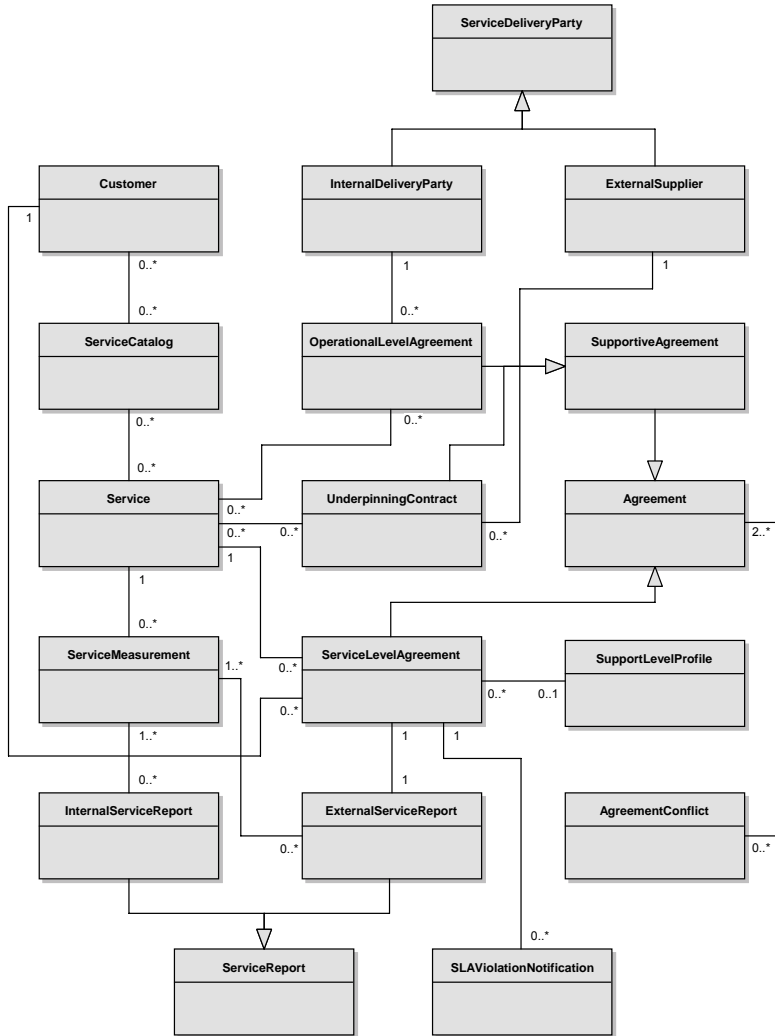


Abbildung 5.4: UML-Konzeptdiagramm für die Disziplin Service-Level-Management, Teil 1

5. Support-Level-Profil (*Support Level Profile*): Beschreibung eines im Rahmen des Abschlusses einer Dienstvereinbarung wählbaren Support-Levels, welche alle maßgeblichen Aspekte der Dienstunterstützung insbesondere im Bereich Erreichbarkeit und Support-Qualität umfasst (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F5)
6. Vereinbarung (*Agreement*)
 - a) Dienstvereinbarung (*Service Level Agreement*): Schriftliche Vereinbarung zwischen dem Dienstanbieter und einem Kunden, in der Dienstleistungen und Service-Levels spezifiziert und dokumentiert sind (vgl. Kapitel 2.3.4, Begriffe auf der Schnittstelle zwischen Kunde und Provider; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F6)
 - b) Unterstützende Vereinbarung (*Supportive Agreement*)
 - i. Operative Vereinbarung (*Operational Level Agreement*) Vereinbarung zwischen dem Service-Level-Manager und einer internen Dienstbringerpartei zur Untermauerung der Erbringung und Qualität von (Sub-)Dienstleistungen, die für die Einhaltung von Dienstvereinbarungen mit IT-Kunden erforderlich sind (vgl. Kapitel 2.3.2, Begriffe der Providerdomäne; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F7)
 - ii. Lieferantenvertrag (*Underpinning Contract*): Vertrag/Dienstvereinbarung zwischen dem Provider und einem externen Zulieferer (vgl. Kapitel 2.3.2, Begriffe der Providerdomäne; Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F8)
7. Vereinbarungskonflikt (*Agreement Conflict*): Feststellung (und Dokumentation) widersprüchlicher Zielvereinbarungen in unterschiedlichen, voneinander abhängigen Vereinbarungen (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F9)
8. Dienstmessung (*Service Measurement*): Ermittlung der Ausprägung festgelegter Dienstgüteparameter mit Hilfe definierter Messverfahren (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F10)
9. (Dienst-)Bericht (*Service Report*)

- a) Interner Bericht (*Internal Service Report*): Bericht über die Leistung sowie Zielabweichungen eines IT-Dienstes unter Berücksichtigung aller für diesen IT-Dienst gültigen Dienstvereinbarungen, gerichtet an alle in diesen Dienst involvierten Dienstleisterparteien (entsprechend operativer Vereinbarungen) und relevante Entscheidungsträger (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F11)
 - b) Externer Bericht (*External Service Report*): Bericht über die Leistung sowie Zielabweichungen eines IT-Dienstes im Hinblick auf eine konkrete Dienstvereinbarung, gerichtet an den Kunden, mit dem diese Dienstvereinbarung geschlossen wurde (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F11)
10. Benachrichtigung über eine SLA-Verletzung (*SLA Violation Notification*): Benachrichtigung eines Kunden über eine akut vorliegende oder drohende Verletzung eines oder mehrerer Ziele der Dienstleistung oder Service-Level-Ziele (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F11)
 11. (Dienst-)Anfrage (*Service Request*): Dokumentierte Nutzeranfrage zu einem Dienst, die das Ziel verfolgt, Unterstützung (*Support*) durch den Dienstleister zu erhalten (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F12)
 12. (Dienst-)Störung (*Service Incident*): Dokumentiertes Ereignis, das nicht zum vereinbarungsgemäßen Betrieb eines Dienstes gehört und tatsächlich oder potenziell eine Dienstdegradierung verursacht (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F12)
 13. (Dienst-)Rechnung (*Service Invoice*): Dokumentierte Leistungsverrechnung für die Inanspruchnahme eines Dienstes durch einen Kunden in einem bestimmten Abrechnungszeitraum (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F13)
 14. (Dienst-)Transitionsplan (*Service Transition Plan*): Dokumentierte Planung zur Transition eines Dienstes in die operative Infrastruktur (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F14)

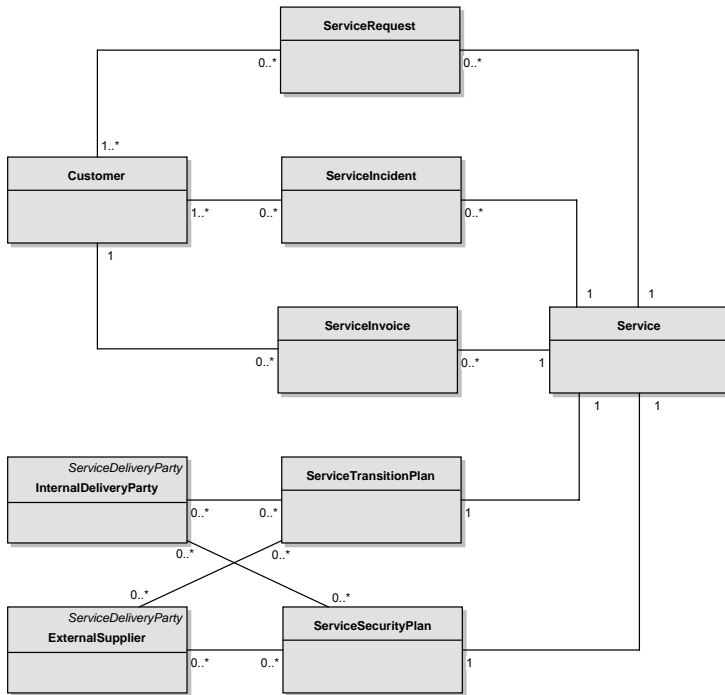


Abbildung 5.5: UML-Konzeptdiagramm für die Disziplin Service-Level-Management, Teil 2

15. (Dienst-)Informationssicherheitsplan (*Service Security Plan*): Dokumentierte Anforderungen an die und Planung der Informationssicherheit eines Dienstes (vgl. Kapitel 3.2, Funktionale Anforderungen an Service-Level-Management – Anforderung F15)

Abbildungen 5.4 und 5.5 zeigen unter Darstellung der Multiplizitäten, wie diese Konzepte zusammenhängen, wenn man das Verständnis zugrunde legt, welches durch die Primäranwendungsfälle geschaffen wurde. Hierzu einige konkretisierende Erläuterungen:

- Jeder Dienst kann in Dienstkatalogen enthalten sein, die für Kunden bereitgestellt werden, wobei ein Dienstkatalog für mehrere Kunden gültig

sein kann und die Anzahl der gültigen Dienstkataloge für einen Kunden nicht limitiert ist.

- Jeder von einem Kunden beauftragte Dienst unterliegt einer Dienstvereinbarung mit diesem Kunden.
- Dienstvereinbarungen stützen sich ab auf interne operative Vereinbarungen und/oder Verträge mit externen Lieferanten.
- Support-Level-Profile stellen eine Basis für die Verhandlung von Dienstvereinbarungen dar, indem sie eine Auswahl allgemeingültiger, dienstunabhängiger Support-Levels bereitstellen.
- Konflikte können sich ergeben, wenn in gültigen Dienstvereinbarungen formulierte Ziele einen höheren Leistungswert des Dienstes zusagen als durch die unterstützenden operativen Vereinbarungen und Lieferantenverträge abgedeckt.
- Für jeden Dienst werden Messungen durchgeführt, deren Ergebnisse die Basis für interne und externe Berichte darstellen.
- Jeder Bericht (intern wie extern) bezieht sich auf einen konkreten (End-)Dienst.
- Interne Berichte adressieren die an der Diensterbringung beteiligten internen Parteien mit dem Ziel, Zielabweichungen zu beseitigen und sowohl den Nutz- als auch den Leistungswert des Dienstes gemäß der für diesen Dienst geschlossenen und gültigen Dienstvereinbarungen zu erreichen.
- Externe Berichte adressieren die Kunden und bilden die Grundlage für die Beurteilung des Erfüllungs- bzw. Nicht-Erfüllungsgrades bestehender Dienstvereinbarungen.

5.2.3 Konzepterweiterungen

Selbstverständlich soll dieses Konzeptmodell nicht als unabänderbare, normative Vorgabe angesehen werden, sondern lediglich einen möglichen, mit den Ergebnissen der Anforderungsanalyse kompatiblen Ausgangspunkt für den zweiten Schritt der Informationsmodellierung aus Prozess-Sicht darstellen. Dass Erweiterungen nicht nur sinnvoll sein können, sondern auch möglich sind, zeigt dieser Abschnitt, in dem beispielhaft drei konzeptionelle Erweiterungen vorgestellt werden, die allerdings im Weiteren nicht als zum Grundmodell gehörend erachtet werden. Damit wird speziell der nicht-funktionalen Anforderung NF4 (Erweiterbarkeit) an die Architektur Rechnung getragen.

5.2.3.1 Multi-Level-SLAs

Bislang wurden Dienstvereinbarungen unter der Annahme betrachtet, dass jede Beauftragung eines Dienstes durch einen Kunden durch genau eine Dienstvereinbarung untermauert wird und dass eine Dienstvereinbarung nicht mehrere Dienste oder mehrere Kunden umfasst. In diesem Kontext stellt sich natürlich schnell die Frage, ob identische Inhalte oder Abschnitte unterschiedlicher Dienstvereinbarungen aus Gründen der Effizienz und zur Vermeidung redundanter Datenhaltung zu einer übergeordneten Dienstvereinbarung gebündelt werden sollten. Dies würde zu einem hierarchischen Ansatz in der Gestaltung und im Management von Dienstvereinbarungen führen. Die *IT Infrastructure Library* prägt in diesem Zusammenhang den Begriff der *Multi-Level-SLAs*.

Um diese Idee im Rahmen der Architektur dieser Arbeit aufzugreifen, ist eine Erweiterung des Konzeptmodells an dieser Stelle erforderlich. Grundsätzlich sind folgende Fälle von Interesse:

1. Eine Dienstvereinbarung bezieht sich auf genau einen Dienst und gilt für genau einen Kunden. → Dies ist der klassische Fall, der im Grundkonzeptmodell Anwendung findet. Das bisherige Konzept `ServiceLevelAgreement` wird zur genaueren Abgrenzung in `CustomerServiceSpecificSLA` umbenannt.
2. Eine übergeordnete Dienstvereinbarung gilt für genau einen Kunden, bezieht sich jedoch auf mehrere (gegebenenfalls alle) Dienste. → Das Konzept `CustomerSpecificServiceCorporateSLA` wird eingeführt.
3. Eine übergeordnete Dienstvereinbarung bezieht sich auf genau einen Dienst, gilt aber für mehrere (gegebenenfalls alle) Kunden, die diesen Dienst beauftragt haben. → Das Konzept `CustomerCorporateServiceSpecificSLA` wird eingeführt.
4. Eine übergeordnete Dienstvereinbarung bezieht sich auf alle Dienste und gilt für alle Kunden. → Das Konzept `CustomerServiceCorporateSLA` wird eingeführt.

Bemerkung: ITIL selbst differenziert an dieser Stelle weniger genau und unterscheidet lediglich zwischen *Corporate level SLAs*, *Customer level SLAs* und *Service specific level SLAs*.

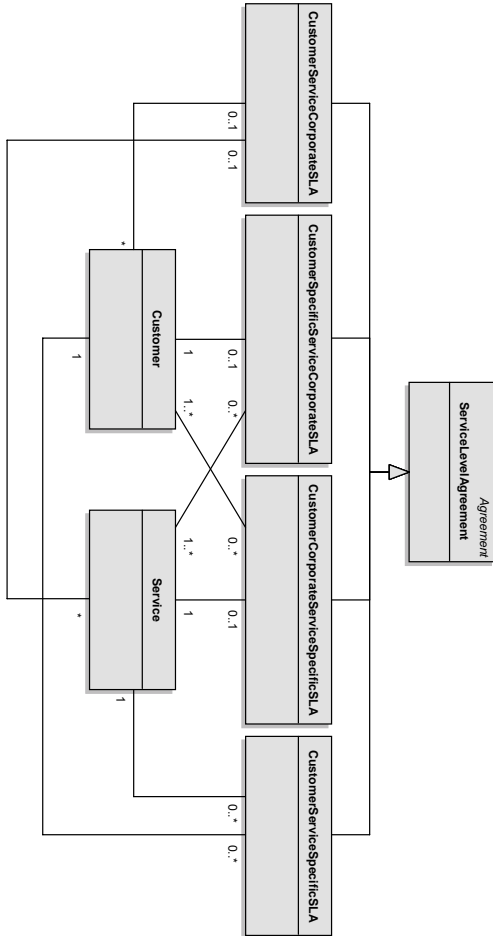


Abbildung 5.6: Konzeptionelle Erweiterung um Multi-Level-SLAs

Abbildung 5.6 illustriert die entsprechend resultierende Erweiterung des Konzeptmodells. Diese liefert natürlich zusätzliches Konfliktpotenzial, da nun weitere vielfältige Beziehungen und Abhängigkeiten zwischen verschiedenen Dienstvereinbarungen entstehen können. Eine in diesem Fall einfache und pragmatische Lösung, mit diesen Konflikten umzugehen, besteht in der Definition eines Überschattungsmechanismus, der im Falle eines Konfliktes bewirkt, dass der in der Hierarchie am weitesten unten befindliche (und damit spezifischste) Inhalt zur Geltung kommt und alle übergeordneten SLAs überschattet werden.

Die Erweiterung der Managementarchitektur um einen solchen Multi-Level-SLA-Ansatz erhöht potenziell die Einsatzfähigkeit und Beherrschbarkeit eines auf ihr aufbauenden IT-gestützten Managementsystems für Service-Level-Management. Allerdings realisiert sie in dieser Eigenschaft keine Vergrößerung des Funktionsumfangs, da hierarchisch aufeinander aufbauende Dienstvereinbarungen letzten Endes unter Inkaufnahme erhöhter Redundanz immer auf einen flachen Ansatz (*Single level SLAs*) abgebildet werden können. Dies rechtfertigt auch, warum Multi-Level-SLAs im Rahmen dieser Arbeit explizit als Erweiterung und nicht als Teil des Grundkonzeptmodells angesehen werden.

5.2.3.2 Diensthierarchien

Das Grundkonzeptmodell sieht Dienstkataloge als Teilsichten auf das Dienstportfolio an, wobei bislang keine explizite (d.h. im Konzeptmodell abgebildete) Differenzierung zwischen *Enddiensten* und *Subdiensten* vorgenommen wurde. Als Enddienste gelten solche Dienste, die von Kunden beauftragt werden und somit Gegenstand von Dienstvereinbarungen sein können. Subdienste hingegen bilden die Bausteine, aus denen Enddienste komponiert werden. Allerdings sind diese beiden Konzepte nicht immer disjunkt: So können bestimmte Dienste sowohl als Enddienste an Kunden bereitgestellt als auch als Subdienste zur Realisierung anderer Enddienste verwendet werden.

Das in Kapitel 4 vorgestellte *NGOSS SLA Management Handbook* legt einen starken Fokus auf den Aspekt der Diensthierarchien im Kontext von Service-Level-Management, der durch die Forderung nach einer *nahtlosen Dienstleistung* (*seamless service delivery*) unter Berücksichtigung der *Herausforderung einer Ende-zu-Ende-Dienstleistung* (*end-to-end service challenge*) zum Ausdruck gebracht wird. Dies legt eine Erweiterung des Konzeptmodells nahe, die die oben erwähnte Differenzierung zwischen End- und Subdiensten explizit macht und operative Vereinbarungen und Lieferantenverträge entsprechend verfeinert.

Eine solche Erweiterung zeigt Abbildung 5.7. Sie verlangt vom Provider, dass

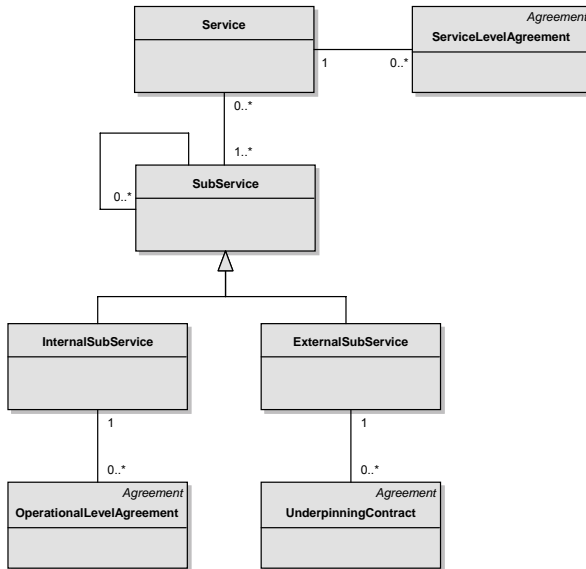


Abbildung 5.7: Konzeptionelle Erweiterung um explizite Diensthierarchien

nicht nur ein differenziertes Portfolio aller den Kunden angebotenen IT-Dienste erstellt wird, sondern dass in einem vorgelagerten Schritt alle Sub- oder Basisdienste als Bausteine für die Enddienste beschrieben werden. Operative Vereinbarungen und Lieferantenverträge werden in Analogie zu Dienstvereinbarungen im Grundmodell subdienstspezifisch mit internen Dienstbringerparteien und externen Lieferanten geschlossen.

Die Tatsache, dass explizite Diensthierarchien als Erweiterung und nicht als Teil des Grundkonzeptmodells betrachtet werden, soll nicht zu dem falschen Schluss verleiten, dass Diensthierarchien und Subdienste im Grundmodell gar keine Rolle spielen. Im Gegenteil sind Diensthierarchien in nahezu allen IT-Provider-Szenarien relevant; das Grundkonzeptmodell abstrahiert lediglich von diesem Umstand.

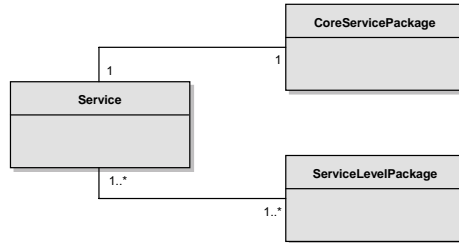


Abbildung 5.8: Konzeptionelle Erweiterung um Dienstmodule

5.2.3.3 Dienstmodularisierung

Bei der Erweiterung des Grundmodells um die Abbildung von Diensthierarchien wurde die Differenzierung zwischen Enddiensten und Subdiensten eingeführt. Dienstmodularisierung hingegen verfolgt das Ziel, im Kontext der Enddienste einen modularen Ansatz zu realisieren, der für jeden (End-)Dienst zwischen einem Basis-Dienstpaket (*Core Service Package*) und mehreren Service-Level-Paketen (*Service Level Package*) unterscheidet. Diese Idee hat ihren Ursprung in der Version 3 der *IT Infrastructure Library*. Was sich dahinter im Detail verbirgt, wird im Folgenden kurz erläutert. Anschließend wird wieder aufgezeigt, wie sich das Konzeptmodell entsprechend erweitern und verfeinern lässt.

Wie schon in Kapitel 2.3.1 beschrieben, setzt sich der Wert eines Dienstes durch seinen Nutzwert und seinen Leistungswert zusammen. Jeder Dienst besitzt also eine Nutzfunktionalität, die auf einem bestimmten Leistungsniveau erbracht wird. Für jede dieser beiden Dimensionen gilt, dass sie für einen konkreten Dienst unterschiedliche Ausprägungen annehmen kann. Dieser Umstand kann es sinnvoll machen, explizit zwischen dem *Basisdienst* als Realisierung der grundlegenden Nutzfunktionalität eines Dienstes auf einem minimal erforderlichen Leistungsniveau und darauf aufbauenden *funktionalen Dienstweiterungen* und höheren realisierbaren Leistungsniveaus zu differenzieren. ITIL empfiehlt zu diesem Zweck die Spezifikation des Basisdienstes im Rahmen eines Basis-Dienstpakets (analog zur Grundausstattung eines Fahrzeugs) und die Spezifikation möglicher Erweiterungen dieses Basisdienstes im Rahmen dedizierter Service-Level-Pakete (vergleichbar mit Sonderausstattungs Paketen).

Wie sich diese Empfehlung leicht in eine Erweiterung des Grundkonzeptmodells überführen lässt, zeigt Abbildung 5.8. Natürlich lässt sich die Dienstmodularisierung auch mit der expliziten Abbildung von Diensthierarchien durch

Modellierung interner und externer Subdienste kombinieren, indem die Differenzierung zwischen Basisdienst und Diensterverweiterungen auch auf diese Subdienste angewendet wird. Man erkennt leicht, dass das Konzeptmodell durch diese kombinierten Erweiterungen zunehmend an Umfang und Komplexität gewinnt. Aus diesem Grund wird bei der nachfolgenden Ableitung und Klassifizierung der Informationsartefakte ausschließlich auf das Grundkonzeptmodell ohne Erweiterungen Bezug genommen.

5.2.4 Ableitung und Klassifizierung von Informationsartefakten

Ausgehend vom grundlegenden Konzeptmodell werden nun konkrete Informationsartefakte definiert, wobei die Klassen Datensatz (*Record*), Dokumentation, Bericht (*Report*), Datenbank (*Database*) und Datenbank-Sicht (*Database view*) als *Stereotypen* zur Anwendung kommen. Für die als Datensatz, Dokumentation und Bericht typisierten Artefakte werden jeweils Klassen von Informationsanforderungen entwickelt und beschrieben, die einen nahtlosen Übergang in das Informationsmodell der System-Sicht ermöglichen. Die Klassen der Informationsanforderungen werden in den UML-Modellen mittels Komposition zu den Informationsartefakten in Beziehung gesetzt.

Die genaue und formale Spezifikation einschließlich Attributierung und Datenmodellierung erfolgt im Rahmen der architekturellen System-Sicht im nachfolgenden Kapitel. Der Fokus dieses Teilkapitels liegt auf der Identifikation der Artefakte und ihrer Informationsanforderungen. Das Ergebnis ist ein erstes Informationsmodell der Prozess-Sicht. Jeder der nachfolgenden Teilabschnitte analysiert und erweitert einen konkreten Teil des übergeordneten Konzeptmodells.

Bemerkung: Stereotypisierung einer UML-Klasse wird typischerweise eingesetzt, um den Verwendungszusammenhang dieser Klasse zu verdeutlichen. Dies ist auch hier der Fall: Durch den Einsatz der Artefakttypen als Stereotypen wird dargestellt, welche übergeordnete Funktion jedes Informationsartefakt erfüllt.

5.2.4.1 Dienst und Dienstkatalog

Der Verfeinerungsschritt beginnt mit den Konzepten *Dienst* und *Dienstkatalog*, die als gemanagte Objekte (*Managed Objects*) im Zentrum der Primäranwendungsfälle 1 bis 3 und 7 bis 12 stehen. Abbildung 5.9 zeigt überblicksartig,

welche Informationsartefakte unmittelbar mit diesen Konzepten in Verbindung stehen. Nachfolgend wird auf diese Artefakte näher eingegangen.

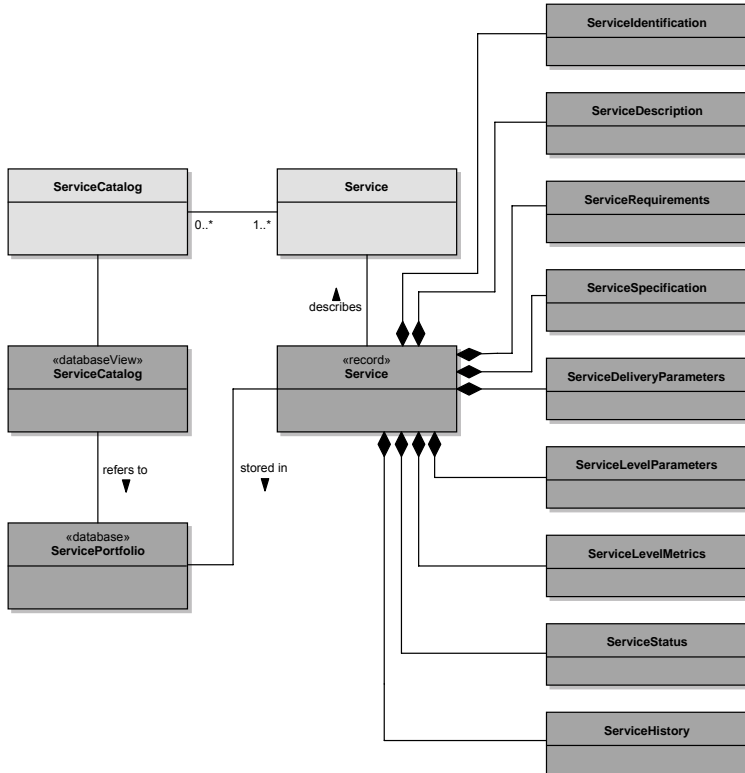


Abbildung 5.9: Partielles Informationsmodell für die Konzepte *Dienst* und *Dienstkatalog*

Service Ein wichtiges und zentrales Informationsartefakt ist das Artefakt **Service**, das als Abstraktion eines generischen IT-Dienstes alle aus Sicht des Service-Level-Managements relevanten Aspekte dieses Dienstes umfasst. Aus Prozess-Sicht sind neun Klassen von Informationsanforderungen zu einem Dienst für ein effektives Service-Level-Management von Bedeutung und müssen durch das **Service**-Artefakt abgedeckt werden. In Ergänzung zur grafischen

ServiceIdentification	Eindeutige Identifikation und Benennung des Dienstes
ServiceDescription	Beschreibung der Kern-Dienstfunktionalität einschließlich aller für potenzielle Kunden relevanten Diensteigenschaften
ServiceRequirements	Bekannte Kundenanforderungen aller Art (technisch, organisatorisch, Informationssicherheits-, usw.) an den Dienst
ServiceSpecification	Spezifikation der technischen Ausgestaltung des Dienstes
ServiceDelivery Parameters	Parameter, deren konkrete Ausprägung sich auf die funktionale Erbringung des Dienstes (Nutzwert) auswirken und die somit verschiedene Varianten des prinzipiell gleichen Dienstes ermöglichen
ServiceLevelParameters	Parameter, deren konkrete Ausprägung sich auf die Dienstperformanz (Leistungswert) auswirken
ServiceLevelMetrics	Menge möglicher (idealerweise eindeutig messbarer) Dienst(güte)parameter
ServiceStatus	Status des Dienstes unter Berücksichtigung des Dienstlebenszyklus
ServiceHistory	Historie zur Nachverfolgung von Änderungen am Service-Artefakt

Tabelle 5.1: Informationsanforderungen an das Artefakt Service

Darstellung in Abbildung 5.9 enthält Tabelle 5.1 zusätzliche Erläuterungen zu jeder der neun Informationsklassen.

Eine eindeutige Identifikation und Benennung eines Dienstes zur Abgrenzung von anderen Diensten ist essenziell für die Verwaltung eines differenzierten Dienstportfolios. Zusätzlich muss der Service-Datensatz eine Beschreibung der Dienstfunktionalität einschließlich aller für tatsächliche und potenzielle Kunden relevanten Diensteigenschaften in adäquater Weise beinhalten. Eine adäquate Beschreibung liegt vor, wenn ihre Verständlichkeit für alle potenziellen Adressaten dieser Informationen, also auch und insbesondere für die IT-Kunden, sichergestellt ist.

Kundenanforderungen an einen Dienst, seine Freigabe und seine Erbringung werden ebenfalls als Teil des Service-Datensatzes gespeichert. Informationen

zur technischen Ausgestaltung eines Dienstes, die für dessen Bereitstellung und Unterstützung (*Support*) benötigt werden, bilden die technische Spezifikation des Dienstes. Hierunter fallen unter anderem auch Abhängigkeiten von bestimmten Ressourcen und Subdiensten.

Unter Umständen, abhängig vom Dienst und seiner technischen Realisierung, können (oder müssen) bestimmte Eigenschaften der Diensterbringung (Erbringungsparameter) variabel festlegbar sein. Dies geschieht während der Verhandlung einer Dienstvereinbarung mit einem Kunden. Ein Beispiel für einen Erbringungsparameter eines E-Mail-Dienstes ist die Anzahl der Konten oder Adressen, die dem Kunden (höchstens) zur Verfügung gestellt werden. Im Service-Artefakt werden nur die Parameter und ihre Wertebereiche definiert. Die spezifische Ausprägung einzelner Parameter wird im Rahmen des SLA-Abschlusses betrachtet.

Analog zu den Erbringungsparametern sind die Service-Level-Parameter die konkreten „Stellschrauben“, über die der Leistungswert eines Dienstes gesteuert oder beeinflusst werden kann. Beispielsweise wäre die Anzahl der physischen Maschinen, auf die die Last eines Dienstes verteilt wird, ein Service-Level-Parameter, dessen Ausprägung natürlich Einfluss auf die im Betrieb erreichte durchschnittliche Antwortzeit (als eine der Service-Level-Metriken) hat.

Mögliche Dienst(güte)parameter werden als Grundlage für die Definition von Service-Level-Zielen im Rahmen von Dienstvereinbarungen benötigt. Nicht alle im Service-Datensatz enthaltenen Metriken müssen später zwangsläufig zur Definition und Messung der Service-Level-Ziele herangezogen werden. Die hier berücksichtigten Metriken stellen aber einen *Pool* möglicher Kennzahlen als Ausgangspunkt für die Definition von Dienstvereinbarungen und zugehörigen Messspezifikationen dar.

Schließlich wird der aktuelle Status des Dienstes im Hinblick auf die Phasen des Dienstlebenszyklus ebenso festgehalten wie Änderungen am Dienst bzw. Service-Datensatz. Letztere bilden die Diensthistorie. Im Sinne einer Versionierung soll sich auf Basis dieser Informationen jede frühere Version des Informationsartefakts Service wiederherstellen lassen.

Bemerkung: Diese Informationsanforderungen entspringen einer reinen SLM-Sicht. Anders gesagt besteht nicht der Anspruch, durch das hier beschriebene Dienstartefakt *alle denkbaren* managementrelevanten Aspekte eines IT-Dienstes abzudecken, sondern nur diejenigen, die aus der Perspektive eines integrierten Service-Level-Managements im Sinne der Anforderungsanalyse unmittelbar erfor-

derlich sind. Analoges gilt für alle weiteren Informationsartefakte, die in diesem Kapitel beschrieben werden.

ServicePortfolio Das Dienstportfolio, realisiert durch das Artefakt **ServicePortfolio**, umfasst die Gesamtheit aller angebotenen IT-Dienste. Ein Provider pflegt genau ein Dienstportfolio, unabhängig von der Zahl seiner Dienste und IT-Kunden. Es bietet die Grundlage für die Definition von Dienstkatalogen. In dieser Eigenschaft kann das Dienstportfolio als ein klassisches Datenbankartefakt angesehen werden. Entsprechend beschränken sich die Informationsanforderungen an dieses Artefakt auf die Speicherung bzw. Referenzierung der enthaltenen Objekte, also der **Service-Datensätze**.

Bemerkung: Grundsätzlich gilt für jedes Datenbankartefakt der Prozess-Sicht, dass dieses nicht zwangsläufig das Vorhandensein einer physischen Datenbank als Teilkomponente der Systemarchitektur mit sich bringen muss. Entscheidend ist am Ende nur, dass alle Informationsanforderungen der Prozess-Sicht erfüllt werden und sich das tatsächliche Systemdesign auf das Prozessmodell abbilden lässt – und umgekehrt.

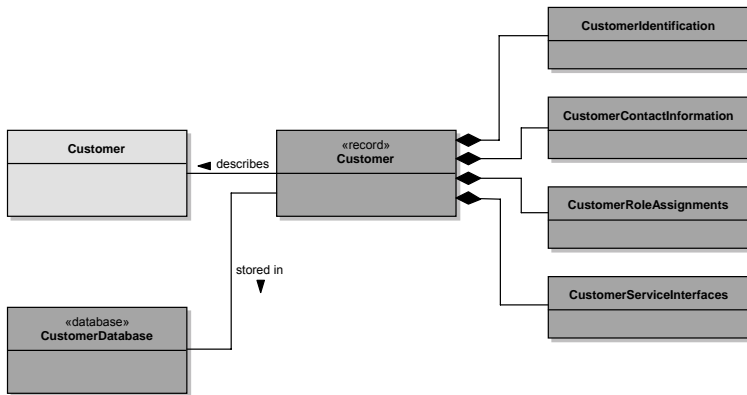
ServiceCatalog Ein Dienstkatalog realisiert eine kundenspezifische Sicht auf das Dienstportfolio. So enthält ein Dienstkatalog alle für einen oder mehrere bestimmte Kunden gültigen und somit beziehbaren Dienste. Realisiert wird ein Dienstkatalog durch das Informationsartefakt **ServiceCatalog**, welches vom Typ Datenbank-Sicht ist, da in jedem Fall gilt: Die durch einen Dienstkatalog repräsentierten Informationen sind eine echte Teilmenge der Informationen des gesamten Dienstportfolios, ergänzt um Meta-Informationen, die den Dienstkatalog selbst als Objekt beschreiben.

Tabelle 5.2 listet die Informationsanforderungen an das Informationsartefakt **ServiceCatalog** auf. Sie sind aus Gründen der Übersichtlichkeit nicht zusätzlich in Abbildung 5.9 abgebildet.

Bemerkung: Während Datenbankartefakte beim Systementwurf genau einmal instanziiert werden, können beliebig viele Sichten auf eine Datenbank existieren. Das erklärt, warum sich für Informationsartefakte des Typs Datenbank keine zusätzlichen Informationsanforderungen wie zum Beispiel eine Gültigkeitsdauer ergeben, während diese für Artefakte des Typs Datenbank-Sicht erforderlich sein können.

ServiceCatalog Identification	Benennung und eindeutige Identifizierung des Dienstkatalogs
ServiceCatalogValidity	Gültigkeitsdauer des Dienstkataloges einschließlich der Festlegung von Rezensionsintervallen/-zeitpunkten
ServiceCatalogHistory	Aufzeichnung aller Änderungen am Dienstkatalogartefakt

Tabelle 5.2: Informationsanforderungen an das Artefakt ServiceCatalog

Abbildung 5.10: Partielles Informationsmodell für das Konzept *Kunde*

5.2.4.2 IT-Kunde

Für ein effektives Service-Level-Management ist es nicht nur wichtig, dass ein Provider sein Dienstportfolio kennt und verwaltet, sondern dass er auch Informationen über sein Kundenumfeld besitzt. Der Kunde (*Customer*) entspricht einer der Domänen des Organisationsmodells der Prozess-Sicht und bildet eine der Klassen im Konzeptmodell. Auch SID kennt die Klasse *Customer* als Teil des *Party*-Konzepts [65].

Abbildung 5.10 illustriert die Informationsartefakte *Customer* vom Typ Datensatz sowie *CustomerDatabase* vom Typ Datenbank und setzt sie in Beziehung zur Konzeptklasse *Customer*. Darüber hinaus zeigt die Abbildung die Komposition des *Customer*-Datensatzes aus verschiedenen Informationsklassen, deren Anforderungen in Tabelle 5.3 näher spezifiziert sind.

CustomerIdentification	Eindeutige Identifikation des IT-Kunden
CustomerContact Information	Informationen zu Ansprechpartnern, Erreichbarkeit und Möglichkeiten der Kontaktaufnahme mit dem IT-Kunden
CustomerRole Assignments	Informationen zu relevanten Rollenzuweisungen, Dienst- und Vertragsverantwortlichkeiten innerhalb der Kundendomäne
CustomerService Interfaces	Informationen über tatsächliche (aktive) oder potenzielle Schnittstellen (wie physische Standorte, technische Anbindung), die als Dienstübergabepunkte zwischen Provider und Kunde verwendet werden (können)

Tabelle 5.3: Informationsanforderungen an das Artefakt Customer

Besonders hervorzuheben sind hier die Rollenzuweisungen und die Informationen über kundenspezifische Dienstübergabepunkte. Erstere sind aus Sicht des Service-Level-Managements für den Provider wichtig, um situationsbezogen effektiv den richtigen Ansprechpartner ermitteln zu können. Die Rolle des Kundenvertreters aus dem Organisationsmodell der Prozess-Sicht kann gegebenenfalls von unterschiedlichen Personen innerhalb der Kundendomäne mit unterschiedlichen (Dienst-)Zuständigkeiten ausgefüllt werden. Informationen über bereits bestehende oder potenzielle physische oder virtuelle Dienstübergabepunkte können die Dienstbereitstellung und den Abschluss einer Dienstvereinbarung, der die Festlegung der Schnittstelle zwischen Provider und Kunde für einen spezifischen Dienst beinhaltet, deutlich erleichtern und beschleunigen. Gemäß MNM-Dienstmodell liegt der Dienstübergabepunkt zwischen dem Dienst-Client und dem Dienstzugangspunkt. In manchen Fällen können Dienstzugangspunkt und Dienstübergabepunkt auch zusammen fallen.

5.2.4.3 Diensterbringerpartei

Nicht nur über die Dienstnehmerseite, sondern auch über die an der Dienstbringung beteiligten Parteien müssen dem Service-Level-Management Informationen vorliegen, die den auf diesen Bereich bezogenen Informationsbedarf der jeweiligen Anwendungsfälle abdecken. Abbildung 5.11 zeigt die Informationsartefakte, die im Zusammenhang mit dem Konzept *Diensterbringerpartei* stehen.

Da die übergeordneten Informationsanforderungen an die Artefakte zur Re-

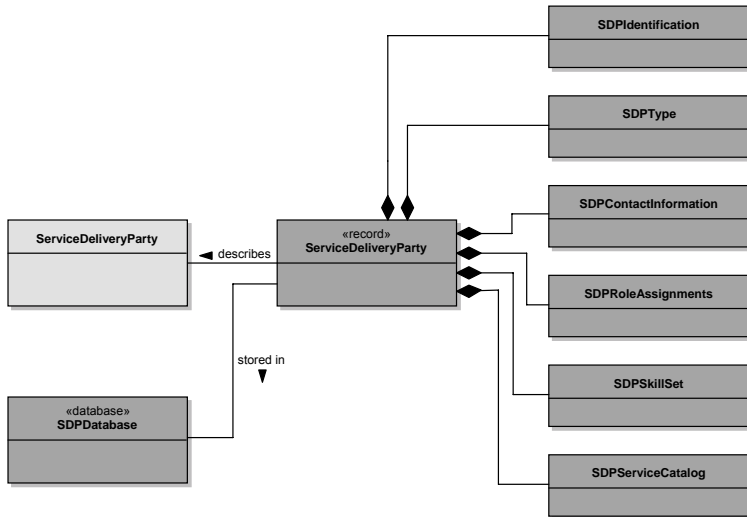


Abbildung 5.11: Partielles Informationsmodell für das Konzept *ServiceDeliveryParty*

präsentation von *internen Dienstbringerparteien* und *externen Zulieferern* – also die Vorgaben zur Bildung konkreter Datenmodelle – identisch sind, erfolgt die Differenzierung zwischen beiden Konzepten über die Informationsklasse *SDPType*. Diese und weitere Klassen von Informationsanforderungen fasst die Tabelle 5.4 zusammen.

Besondere Erwähnung verdienen die Informationsanforderungen *SDPSkillSet* und *SDPServiceCatalog*. Das *SDPSkillSet* einer konkreten Dienstbringerpartei soll in dienstunabhängiger Weise beschreiben, in welchen dienstrelevanten technischen Bereichen und Teildisziplinen die betreffende Partei welche Kompetenzen und Fähigkeiten aufweist. Welche Kategorien in welcher Granularität hierfür sinnvollerweise gewählt werden, hängt wesentlich vom betrachteten Anwendungsszenario ab. Ein Beispiel für eine sehr grobe Kategorisierung ist eine Differenzierung nach den Hauptkompetenzfeldern Software, Hardware und Support/Beratung. Schließlich könnte im Bereich Software weiter nach Desktop-Software, Betriebssoftware, Server-Software und Spezialsoftware unterschieden werden; analog für die anderen Hauptkompetenzfelder.

Im Unterschied zum *SDPSkillSet* soll der *SDPServiceCatalog* konkrete von der betrachteten Dienstbringerpartei erbrachte oder angebotene IT-

SDPIdentification	Eindeutige Benennung und Identifikation der an der Dienstleistung beteiligten Partei
SDPType	Legt fest, ob es sich um eine interne Partei (Konzeptklasse <i>InternalDeliveryParty</i>) oder um einen externen Zulieferer (Konzeptklasse <i>ExternalSupplier</i>) von IT-Dienstleistungen handelt
SDPContactInformation	Informationen zu Ansprechpartnern, Erreichbarkeit und Möglichkeiten der Kontaktaufnahme mit der Dienstleistungspartei
SDPRoleAssignments	Informationen zu relevanten Rollenzuweisungen, Dienst- und Vertragsverantwortlichkeiten innerhalb der Provider- bzw. Lieferantendomäne
SDPSkillSet	Informationen zu Kompetenzen, Fähigkeiten (<i>Skills</i>) sowie generellen Zuständigkeiten im Hinblick auf die Dienstleistung
SDPServiceCatalog	Informationen zu den IT-Dienstleistungen, die von der Dienstleistungspartei angeboten werden (Dienstkatalog der Dienstleistungspartei)

Tabelle 5.4: Informationsanforderungen an das Artefakt *ServiceDeliveryParty*

Dienstleistungen enthalten – ganz in Analogie zu den Dienstkatalogen, die der IT-Provider seinen Kunden bereitstellt.

Mit der Realisierung der Informationsartefakte *ServiceDeliveryParty* vom Typ Datensatz und *SDPDatabase* vom Typ Datenbank unter Berücksichtigung der beschriebenen Klassen von Informationsanforderungen ist der Weg dafür geebnet, dass der Provider zu jeder Zeit über alle relevanten Informationen zu verfügbaren internen Dienstleistern und externen Lieferanten, Kontaktmöglichkeiten und Ansprechpartnern, Kompetenzen und bereitgestellten Diensten verfügt, die für ein effektives Service-Level-Management wichtig sind.

5.2.4.4 Support-Level-Profil

Das Support-Level-Profil soll die für die Unterstützung eines Dienstes relevanten Randbedingungen kapseln, sodass diese Informationen von den eigentlichen Dienstvereinbarungen entkoppelt sind. Diese modulare Art und Weise der Datenhaltung und Informationsverwaltung ist sinnvoll, um Redundanzen zu vermeiden und die Administration der Informationen zu erleichtern.

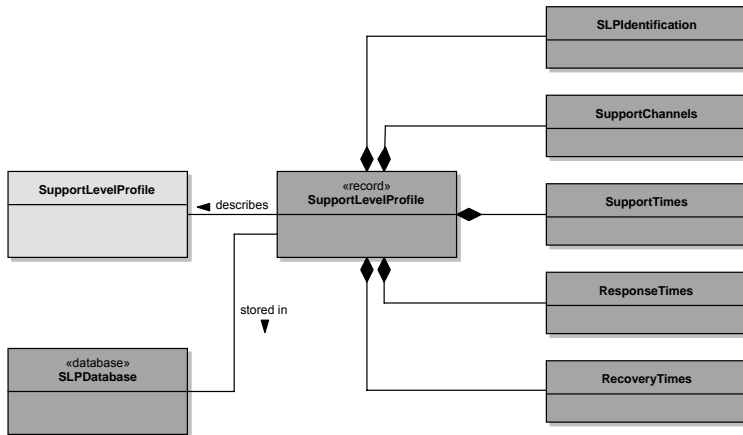


Abbildung 5.12: Partielles Informationsmodell für das Konzept *Support-Level-Profil*

Beispielsweise könnte ein Provider zwischen den generellen Support-Leveln „Gold“, „Silber“ und „Bronze“ unterscheiden, wobei „Gold“ den „besten“ (im Sinne von am besten erreichbaren, reaktionsschnellsten), aber auch teuersten Support gewährleistet. Eine Änderung an diesem Support-Level müsste dank der Existenz eines Support-Level-Profiles nur an einer einzigen Stelle vorgenommen werden, da die Dienstvereinbarungen, die dieses Support-Level enthalten, das entsprechende Profilartefakt referenzieren.

Abbildung 5.12 zeigt das partielle Informationsmodell für die Konzeptklasse `SupportLevelProfile`, das ein gleichnamiges Datensatzartefakt sowie das Informationsartefakt `SupportLevelDatabase` vom Typ Datenbank definiert, ganz analog zu den zuvor betrachteten Ausschnitten des Informationsmodells. Tabelle 5.5 gibt wieder einen Überblick über die im Diagramm enthaltenen Klassen von Informationsanforderungen.

Neben der eindeutigen Identifikation des Support-Level-Profiles wird vor allem gefordert, dass die Kanäle festgelegt werden, über die Support-Anfragen von Anwendern an den Provider abgesetzt werden können, dass die Zeiträume festgelegt werden, in denen diese Kanäle jeweils aktiv sind, und dass maximale Reaktionszeiten auf Anfragen und Wiederherstellungszeiten für Störungen definiert werden, die bei Bedarf durch abweichende Festlegungen in Dienstvereinbarungen überschattet werden können.

SLPIdentification	Eindeutige Identifikation und Benennung des Support-Level-Profiles
SupportChannels	Support-Kanäle: Über welchen Weg/welche Wege ist die Support-Gruppe – meistens gegeben durch den <i>Service Desk</i> – für Kunden und Anwender im Falle eines Support-Anliegens erreichbar?
SupportTimes	Festlegung der Zeiträume, in denen die Erreichbarkeit über die definierten Support-Kanäle garantiert wird
ResponseTimes	Maximale Reaktionszeiten: Wie viel Zeit darf maximal vergehen, bis eine Reaktion (Antwort) seitens der Support-Gruppe auf eine Anfrage durch einen Anwender erfolgt?
RecoveryTimes	Maximale Wiederherstellungszeiten: Wie viel Zeit darf maximal vergehen, bis eine gemeldete bzw. festgestellte Dienstdegradierung behoben wird?

Tabelle 5.5: Informationsanforderungen an das Artefakt `SupportLevelProfile`

5.2.4.5 Dienstvereinbarung (SLA)

Eine Dienstvereinbarung (*Service Level Agreement*, SLA) ist ein vertragsähnliches Dokument, das eine Provider-Kunden-Beziehung zielbezogen dokumentiert, einen Dienst und ein Support-Level-Profil referenziert und für die Erbringung des betreffenden Dienstes Ziele definiert, die sich sowohl auf den Nutz- als auch auf den Leistungswert des Dienstes sowie wichtige Randbedingungen seiner Erbringung beziehen. Die Anforderungen an den Teil des Informationsmodells der Architektur, der sich mit der Datenmodellierung zur Realisierung von Dienstvereinbarungen beschäftigt, sind in Abbildung 5.13 wieder als Klassen dargestellt, deren Komposition das Informationsartefakt `ServiceLevelAgreement` vom Typ `Dokumentation` ergibt. Tabelle 5.6 spezifiziert die einzelnen Informationsanforderungen genauer.

Die Klassen `ServiceDeliveryTargets` und `ServiceLevelTargets` haben eine Entsprechung im referenzierten Dienstartefakt durch die dort enthaltenen Informationsklassen `ServiceDeliveryParameters` und `ServiceLevelMetrics`. Während durch eine Instanz des Informationsartefakts `Service` ein konkreter Dienst aus dem Portfolio des Providers beschrieben wird und die Erbringungsparameter dabei die „Stellschrauben“ definieren, die es ermöglichen, die funktionale Er-

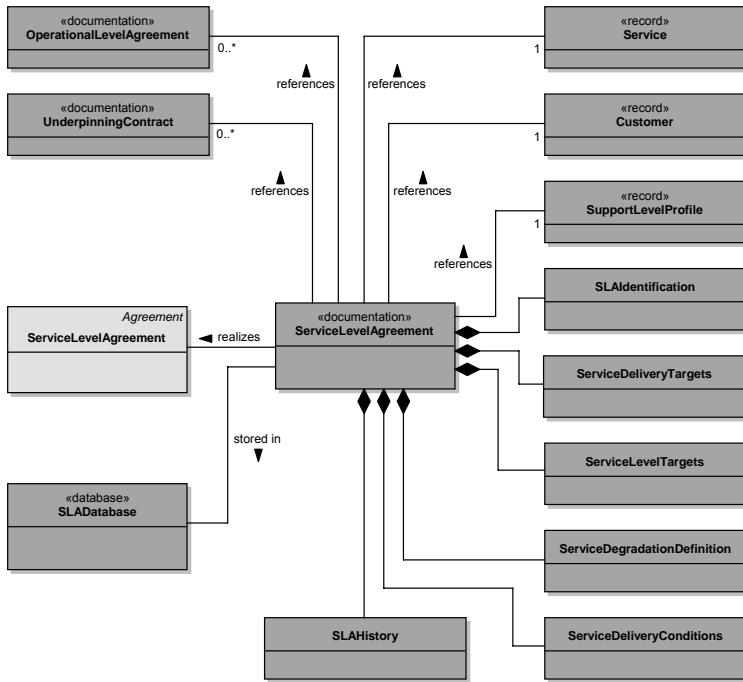


Abbildung 5.13: Partielles Informationsmodell für das Konzept *ServiceLevelAgreement*

bringung des Dienstes an die spezifischen Kundenanforderungen anzupassen, wird durch eine Instanz des Informationsartefakts *ServiceLevelAgreement* die konkrete und kundenspezifische Dienstleistung auf Basis dieser Instanz des Dienstartefakts festgelegt. Hierzu werden Vereinbarungen über die Ausprägung der Erbringungsparameter getroffen, die als Ziele der Dienstleistung (ServiceDeliveryTargets) spezifiziert werden. Analoges gilt für die Service-Level-Metriken, wobei in beiden Fällen nicht alle in der assoziierten Instanz des Dienstartefakts definierten Parameter bzw. Metriken im Rahmen der Dienstvereinbarung ausgeprägt werden müssen. Zum besseren Verständnis sei auf das konkrete Beispiel in Kapitel 7.2 verwiesen.

Service-Level-Management beschäftigt sich aber nicht nur mit den gewünschten und planmäßigen Zuständen von IT-Diensten und ihrer Erbringung, son-

SLAIdentification	Eindeutige Identifikation der Dienstvereinbarung
ServiceDeliveryTargets	Vereinbarte Ziele für die Ausprägung der Erbringungsparameter aus dem assoziierten Dienstartefakt zur Sicherstellung eines Nutzwertes, der den Kundenanforderungen entspricht
ServiceLevelTargets	Vereinbarte Ziele für die Ausprägung der Service-Level-Metriken aus dem assoziierten Dienstartefakt zur Sicherstellung eines Leistungswertes, der den Kundenanforderungen entspricht
ServiceDegradation Definition	Definition der Kriterien für die Feststellung einer Dienstdegradierung: Ausgehend von definierten Degradierungsdimensionen wird festgelegt, wie die Degradierung des betreffenden Dienstes festgestellt und quantifiziert wird.
ServiceDeliveryConditions	Alle relevanten (Rand-)Bedingungen und Verantwortlichkeiten in der Dienstleistung
SLAHistory	Dokumentation zurückliegender Änderungen an der Dienstvereinbarung

Tabelle 5.6: Informationsanforderungen an das Artefakt `ServiceLevelAgreement`

dem auch mit unerwünschten Dienstzuständen, die den Nutz- und/oder Leistungswert des Dienstes negativ beeinträchtigen. Wann genau ein Dienst in diesem Zusammenhang als *degradiert* gilt, muss in jeder Dienstvereinbarung eindeutig festgelegt werden. Grundlage für die Festlegung dieser Kriterien sind mögliche Dimensionen der Degradierung eines Dienstes. Schmitz [12] liefert in seinem Rahmenwerk zur Unterstützung der Dienst- und Geschäftsauswirkungsanalyse im Fehlerfall eine sehr hilfreiche Definition des Degradierungsbegriffs, die zwischen folgenden Dimensionen differenziert:

- Degradierungssubjekt
- Degradierungsabhängigkeiten
- Degradierungsweise
- Degradierungsgrad

- Degradierungszeit

Die Informationsklasse **ServiceDegradationDefinition** verlangt, dass anhand dieser Dimensionen genau definiert wird, unter welchen Bedingungen der Dienst, auf den sich die Dienstvereinbarung bezieht, als wie stark degradiert angesehen wird.

Die Informationsanforderung **ServiceDeliveryConditions** beschäftigt sich mit den wichtigen Aspekten der Definition aller Randbedingungen der Dienstleistung sowie der Festlegung von Verantwortlichkeiten. So kann etwa festgelegt werden, dass die Nutzung des Dienstes durch einen Anwender nur unter der Bedingung ermöglicht wird, dass eine bestimmte Version eines Dienst-Clients (vgl. Kapitel 2.3.1) verwendet wird. Im Kontext mancher Dienste kann es sinnvoll sein, bestimmte alltägliche und einfache Wartungsaktivitäten der Verantwortung des Kunden zu übertragen.

5.2.4.6 Unterstützende Vereinbarung

Während eine Dienstvereinbarung gemäß MNM-Dienstmodell eine Provider-Kunden-Beziehung der Ausprägung *Dienstleistung* untermauert, wird eine unterstützende Vereinbarung geschlossen, um wiederum bestimmte Teile einer oder mehrerer Dienstvereinbarungen auf eine belastbare Grundlage zu stellen. Jedes **SupportiveAgreement** wird dazu mit einer an der Dienstleistung beteiligten Partei (**ServiceDeliveryParty**) über die Erbringung eines oder mehrerer Subdienste geschlossen und ist für den Kunden transparent. Abbildung 5.14 zeigt das grundlegende Modell für das Informationsartefakt **SupportiveAgreement**, Tabelle 5.7 enthält einige Erläuterungen zu den assoziierten Klassen von Informationsanforderungen.

Abhängig davon, ob es sich bei der Dienstleistungspartei um eine interne Partei oder einen externen Zulieferer handelt, wird zwischen den im Folgenden beschriebenen Formen unterstützender Vereinbarungen differenziert.

OperationalLevelAgreement (OLA) Eine operative Vereinbarung dokumentiert Zielvereinbarungen, die im Rahmen des Service-Level-Managements mit einer an der Dienstleistung beteiligten *internen* Partei getroffen werden. Operative Vereinbarungen dienen der Untermauerung von in SLAs vereinbarten bzw. zu vereinbarenden Zielen. Dabei handelt es sich allerdings nicht um eine 1-zu-1-Beziehung. Vielmehr kann eine operative Vereinbarung mehrere Dienstvereinbarungen unterstützen, und eine Dienstvereinbarung kann durch mehrere operative Vereinbarungen unterlegt sein.

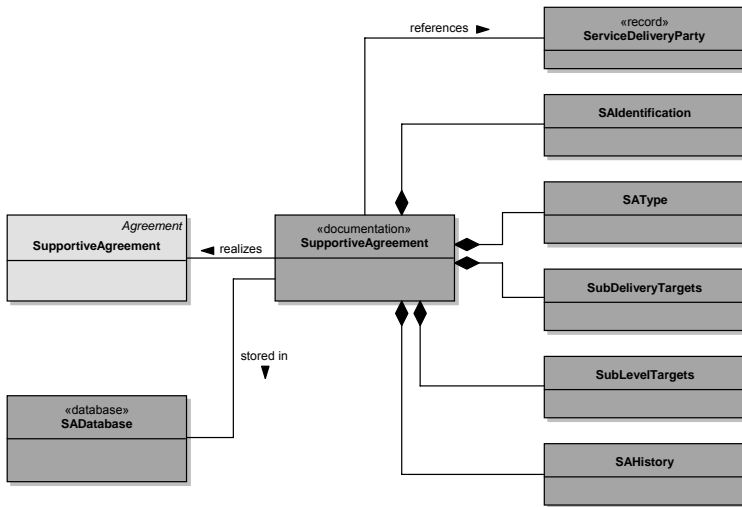


Abbildung 5.14: Partielles Informationsmodell für das Konzept *Unterstützende Vereinbarung*

SAIdentification	Eindeutige Identifikation der unterstützenden Vereinbarung
SAType	Legt fest, ob es sich um eine operative Vereinbarung (Konzeptklasse <i>OperationalLevelAgreement</i>) oder um einen Vertrag mit einem externen Zulieferer (Konzeptklasse <i>UnderpinningContract</i>) handelt
SubDeliveryTargets	Beschreibung der Dienstleistungen (technische Systeme, Subdienste, etc.), die Gegenstand der unterstützenden Vereinbarung sind und von der Dienstleisterpartei bereitzustellen sind
SubLevelTargets	Vereinbarte leistungsbezogene Zielsetzungen für die Dienstleistungen, die Gegenstand der unterstützenden Vereinbarung sind
SAHistory	Dokumentation zurückliegender Änderungen an dieser unterstützenden Vereinbarung

Tabelle 5.7: Informationsanforderungen an das Artefakt *SupportiveAgreement*

UnderpinningContract (UC) Ein Lieferantenvertrag dokumentiert Zielvereinbarungen, die zwischen dem Provider und einem seiner *externen* Lieferanten bezüglich der Zulieferung von technischen System, Subdiensten und sonstigen Leistungen, die vom Provider zur Realisierung einer oder mehrerer seiner Dienste erforderlich sind, getroffen werden. Die elementaren Informationsanforderungen sind mit denen der operativen Vereinbarungen (OLAs) vergleichbar – jeweils auf einen externen Lieferanten statt auf eine interne Partei bezogen.

5.2.4.7 Dienstmessung

Das Grundkonzept der Dienstmessung (*Service Measurement*) wird an dieser Stelle verfeinert, indem zwischen der Spezifikation einer Messung auf der einen Seite und ihrer eigentlichen Durchführung auf der anderen Seite differenziert wird. Nur die Durchführung einer Messung gemäß Messspezifikation führt zu einer Reihe von Messergebnissen, die in einem Messdatensatz festgehalten werden. Abbildung 5.15 stellt die entsprechenden Informationsartefakte `ServiceMeasurementSpecification`, `ServiceMeasurementResults` (jeweils vom Typ Datensatz) und `MeasurementDatabase` (vom Typ Datenbank), ihre Zusammenhänge sowie für die beiden Datensatzartefakte die aus Sicht des Service-Level-Managements jeweils relevanten Informationsklassen dar.

ServiceMeasurementSpecification Die zu einem Dienst aus dem Portfolio gehörende Messspezifikation muss genaue Vorgaben darüber machen, was und wie während der Betriebsphase des Dienstes gemessen werden soll. Tabelle 5.8 beschreibt die verschiedenen Klassen von Informationsanforderungen an das Artefakt `ServiceMeasurementSpecification`. Einen Ansatz, um aus formalen Dienstmessspezifikationen automatisch Messsysteme zu generieren, liefert Garschhammer mit seiner bereits in Kapitel 1 erwähnten Arbeit zur Dienstgütebehandlung [13].

ServiceMeasurementResults Leistungsparameter von Diensten, die gemäß Messspezifikation erhoben wurden, müssen aufgezeichnet und in Datensätzen gespeichert werden. Diese wiederum werden vor allem zur Generierung von Berichten benötigt. Ziel des Informationsartefakts `ServiceMeasurementResults` vom Typ Datensatz ist also die Speicherung der Ergebnisse von in einem bestimmten Zeitraum durchgeführten Messungen. Tabelle 5.9 konkretisiert die Klassen von Informationsanforderungen für dieses Artefakt.

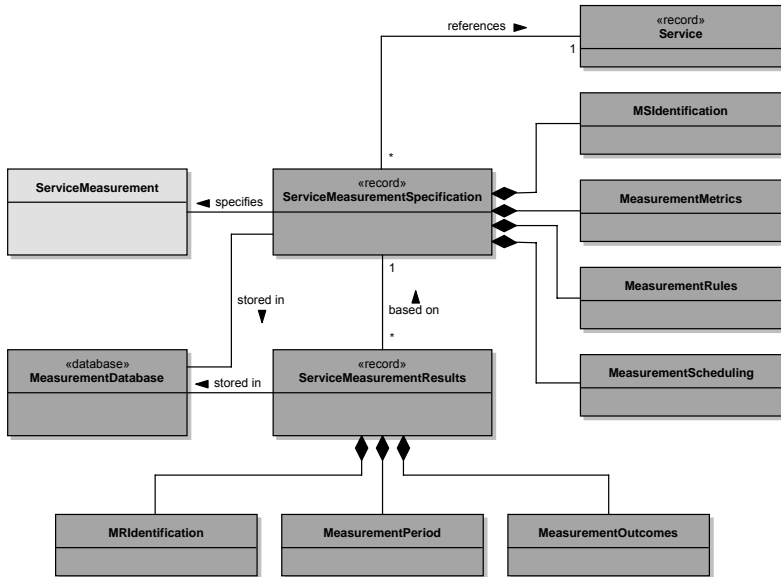


Abbildung 5.15: Partielles Informationsmodell für das Konzept *Dienstmessung*

MSIdentification	Eindeutige Identifikation der Messspezifikation
MeasurementMetrics	Festlegung der zu messenden Leistungsparameter auf Basis des assoziierten Service-Datensatzes und über den assoziierten Dienst abgeschlossener Dienstvereinbarungen
MeasurementRules	Messvorschriften, die vorgeben, auf welche Weise die definierten Metriken gemessen und ermittelt werden
MeasurementScheduling	Angaben über die Frequenz und Terminierung von Messungen gemäß dieser Spezifikation

Tabelle 5.8: Informationsanforderungen an das Artefakt `ServiceMeasurementSpecification`

MRIdentification	Eindeutige Identifikation des Messergebnisdatensatzes
MeasurementPeriod	Genaue Angabe des Zeitraums, in dem die Messergebnisse erhoben wurden.
MeasurementOutcomes	Dokumentation der eigentlichen Messergebnisse in aussagekräftiger Art und Weise und in Konformität zur zugrunde liegenden Messspezifikation

Tabelle 5.9: Informationsanforderungen an das Artefakt ServiceMeasurementResults

5.2.4.8 Dienstbericht

Im Rahmen des Service-Level-Managements müssen regelmäßig Berichte erstellt werden. Diese sind nicht etwa nur ein „Nebenprodukt“ des Dienstmanagements, sondern Kernbestandteil des Managements der Dienstqualität, da nur auf der Grundlage von regelmäßigen und aussagekräftigen Berichten ein kontinuierlicher Verbesserungsprozess – bezogen auf die Dienstqualität – institutionalisiert werden kann. So fordert auch der Qualitätsmanagementstandard ISO 9000 [21] ein „faktisches Vorgehen zur Entscheidungsfindung“ (*factual approach for decision making*), wonach Managemententscheidungen und Verbesserungsaktivitäten stets auf der Basis belastbarer Fakten getroffen bzw. unternommen werden sollen.

Damit Berichte in dieser Weise sinnvoll instrumentiert werden können, ist es wichtig, dass sie in standardisierter Weise erstellt werden und dass Berichte über unterschiedliche Zeiträume miteinander vergleichbar sind. Berichtsspezifikationen werden eingesetzt, um dies zu gewährleisten. Zudem wird zwischen internen und externen Berichten differenziert, die sich in Inhalten und Adressaten unterscheiden.

Zunächst stehen die beiden Artefakte ServiceReport vom Typ Bericht und ReportSpecification vom Typ Datensatz im Mittelpunkt der Abbildung 5.16. Die Informationsklassen, die für das ServiceReport-Artefakt dargestellt sind, gelten gleichermaßen für interne wie für externe Berichte. Entsprechend erben die Artefaktklassen InternalReport und ExternalReport von ServiceReport. Analoges gilt für das ReportSpecification-Artefakt im Zusammenhang mit den Subklassen InternalReportSpecification und ExternalReportSpecification.

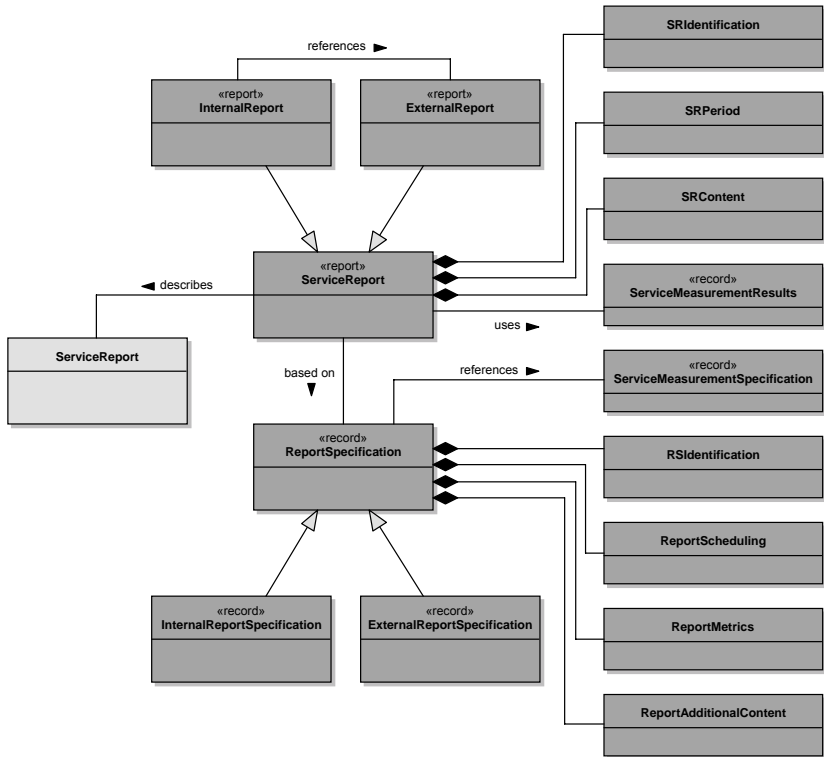


Abbildung 5.16: Partielles Informationsmodell für das Konzept *Dienstbericht*, Teil 1

ReportSpecification Die Klassen von Informationsanforderungen an das Artefakt ReportSpecification werden in Tabelle 5.10 beschrieben. Hier besteht eine enge Analogie zu den Informationsklassen des Artefakts ServiceMeasurementSpecification – begründet durch den Umstand, dass eine oder mehrere Messspezifikationen die Basis für jede Berichtsspezifikation bilden und daher auch entsprechend aus dieser referenziert werden.

ServiceReport Ein Bericht über einen Dienst wird auf der Basis einer hierfür festgelegten Berichtsspezifikation und der erhobenen Messdaten im Idealfall automatisiert generiert und gegebenenfalls um weitere berichtsrelevante Infor-

RSIdentification	Eindeutige Identifikation und Benennung der Berichtsspezifikation
ReportScheduling	Festlegung der Frequenz und Terminierung des spezifizierten Berichts
ReportMetrics	Festlegung der Leistungsparameter aus der oder den assoziierten Messspezifikationen, die in den Bericht übernommen werden sollen
ReportAdditional Content	Spezifikation weiterer erforderlicher Inhalte, die der Bericht enthalten muss

Tabelle 5.10: Informationsanforderungen an das Artefakt ReportSpecification

SRIdentification	Eindeutige Identifikation des Dienstberichts
SRPeriod	Angabe des Zeitraums, auf den sich der Bericht bezieht
SRContent	Inhalte des Berichts, insbesondere Ausprägung der berichtsrelevanten Kennzahlen basierend auf der assoziierten Berichtsspezifikation

Tabelle 5.11: Informationsanforderungen an das Artefakt ServiceReport

mationen manuell ergänzt. Ein Bericht basiert stets auf genau einer Berichtsspezifikation. Umgekehrt können auf Basis einer Berichtsspezifikation prinzipiell beliebig viele Berichte generiert werden. Tabelle 5.11 beschreibt die Klassen von Informationsanforderungen, die für das Artefakt ServiceReport identifiziert werden können.

InternalReport vs. ExternalReport Abbildung 5.17 illustriert abschließend, in welchem Hauptmerkmal sich interne von externen Berichten unterscheiden. Obwohl die Klassen von Informationsanforderungen an die Spezifikationen beider Arten von Berichten identisch sind, besteht der wesentliche Unterschied darin, dass ein interner Bericht mit der Artefaktklasse Service assoziiert ist, während sich ein externer Bericht auf ein ServiceLevelAgreement-Artefakt bezieht.

5.2.4.9 Benachrichtigung über eine SLA-Verletzung

Wie bereits aus der Anforderungsanalyse deutlich wurde, reichen Dienstberichte in der gerade beschriebenen Form nicht in jedem Fall aus, um die funktiona-

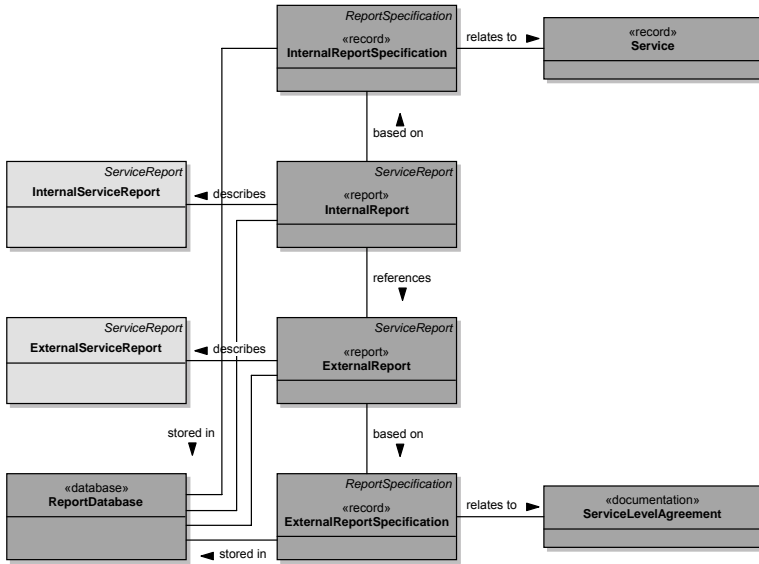


Abbildung 5.17: Partielles Informationsmodell für das Konzept *Dienstbericht*, Teil 2

len Anforderungen an ein effektives und kundenorientiertes Berichtswesen des Service-Level-Managements zu erfüllen. In bestimmten, im Rahmen des Funktionsmodells näher zu spezifizierenden Fällen wird ein zusätzliches Instrument benötigt, welches IT-Kunden zeitnah über drohende oder bereits eingetretene Verletzungen von in Dienstvereinbarungen definierten Zielen informiert. Abbildung 5.18 stellt das in diesem Zusammenhang relevante Informationsartefakt SLAViolationNotification und seine Klassen an Informationsanforderungen vor, die in Tabelle 5.12 näher beschrieben werden.

Dieses Artefakt referenziert stets eine Dienstvereinbarung (*ServiceLevelAgreement*), um die Frage zu beantworten, welches SLA verletzt wurde oder verletzt zu werden droht, sowie einen Messdatensatz (*ServiceMeasurementResults*), auf dessen Grundlage die (potenzielle) Verletzung identifiziert wurde.

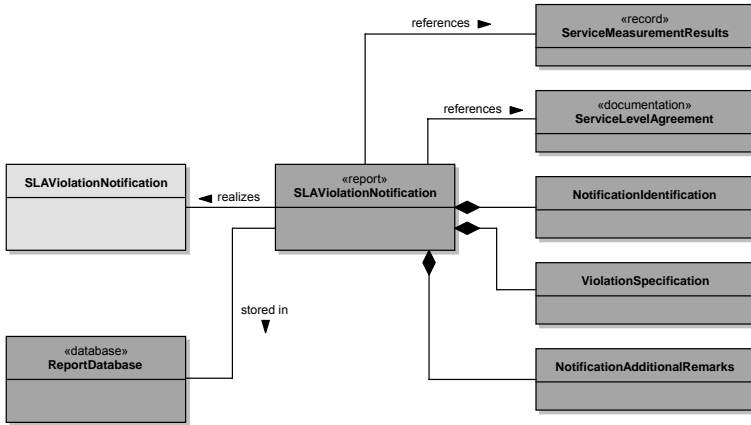


Abbildung 5.18: Partielles Informationsmodell für das Konzept *Benachrichtigung über eine SLA-Verletzung*

NotificationIdentification	Eindeutige Identifikation der Benachrichtigung
ViolationSpecification	Spezifikation der SLA-Verletzung unter Angabe der betroffenen <i>Service Delivery</i> - und <i>Service-Level-Ziele</i> und dem Grad ihrer tatsächlichen und/oder prognostizierten Nicht-Einhaltung
NotificationAdditionalRemarks	Ergänzende Zusatzinformationen zur Übermittlung an den IT-Kunden

Tabelle 5.12: Informationsanforderungen an das Artefakt *SLAViolationNotification*

5.2.4.10 Vereinbarungskonflikt

Das letzte zu verfeinernde Konzept aus dem Grundkonzeptmodell ist der *Vereinbarungskonflikt*. Wie in Abbildung 5.19 zu erkennen ist, bezieht sich ein Konflikt stets auf zwei konkrete *Agreement*-Artefakte. Tabelle 5.13 beschreibt die Klassen der Informationsanforderungen an das Artefakt *AgreementConflict*.

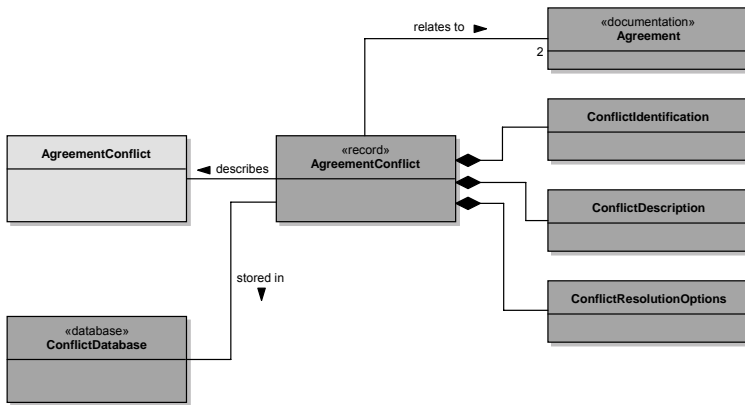


Abbildung 5.19: Partielles Informationsmodell für das Konzept *Vereinbarungskonflikt*

ConflictIdentification	Eindeutige Identifikation des Vereinbarungskonflikts
ConflictDescription	Spezifikation des Konflikts unter Angabe der konfliktären Ziele
ConflictResolution Options	Mögliche Optionen zur (gegebenenfalls teilautomatisierten) Auflösung des Vereinbarungskonflikts

Tabelle 5.13: Informationsanforderungen an das Artefakt AgreementConflict

5.2.5 Zusammenfassung

In diesem Teilkapitel wurde ein Informationsmodell für ein prozessorientiertes Service-Level-Management entwickelt. Dabei wurden in einem ersten Schritt zunächst – gestützt auf funktionalen Anforderungen und Primäranwendungsfällen – die wesentlichen Basiskonzepte für ein effektives Service-Level-Management identifiziert und zu einem übergeordneten objektorientierten Konzeptmodell zusammengefügt.

Dieses Grundkonzeptmodell bildete die Grundlage für den zweiten Schritt, die Ableitung, Typisierung und Verfeinerung konkreter Informationsartefakte. Im Mittelpunkt standen hierbei die Darstellung der Beziehungen zwischen verschiedenen Artefakten einschließlich der Festlegung sinnvoller Multiplizitäten.

ten sowie die Definition von Klassen von Informationsanforderungen für jedes Artefakt. Letztere ebnet den Weg hin zu strukturierten Datenmodellen und somit zum Informationsmodell der System-Sicht.

5.3 Funktionsmodell

Fragestellungen dieses Moduls: Die Fragen, die durch das Funktionsmodell der Prozess-Sicht beantwortet werden müssen, sind:

- Welche Funktionsbereiche lassen sich auf Basis der Ergebnisse aus der Anforderungsanalyse für die Disziplin Service-Level-Management ermitteln?
- Wie ordnen sich die vorliegenden Anwendungsfälle in diese Funktionsbereiche ein?
- Wie sehen mögliche Abläufe (Workflows) dieser Anwendungsfälle in den einzelnen Funktionsbereichen unter Berücksichtigung aller zeitlichen und kausalen Abhängigkeiten zwischen Anwendungsfällen aus?
- Welche Verantwortungsverteilung ergibt sich innerhalb dieser Workflows für die Rollen des Organisationsmodells gemäß der Verantwortungsmodelle?
- Welche Rolle spielen die Artefakte des Informationsmodells im Rahmen der Workflows? Welche Objektflüsse ergeben sich daraus?

5.3.1 Festlegung der Funktionsbereiche

In Kapitel 3 wurde ein allgemeines Funktionsspektrum eines Managementsystems für Service-Level-Management beschrieben, das aus 15 Funktionsanforderungen besteht. Dieses Funktionsspektrum bietet einen sinnvollen Rahmen für die funktionale Einordnung der Anwendungsfälle, die diesem Kapitel (Prozess-Sicht der Architektur) gemäß Entwicklungsrahmen zugrunde liegen. Eine solche Zuordnung von Anwendungsfällen zu Funktionsanforderungen wurde in Tabelle 3.2 vorgenommen.

Eine gröbere Aufteilung des Funktionsspektrums als die in die 15 einzelnen Funktionsanforderungen ist durch die drei Ringe aus Abbildung 3.10 gegeben. Abbildung 5.20 greift diese Einordnung auf, setzt aber statt der Funktionsanforderungen F1 bis F15 die Primäranwendungsfälle in die entsprechenden Segmente. Schwerpunktmäßig werden in diesem Teilkapitel die beiden inneren Segmente betrachtet, die die fundamentalen Kern-Funktionen, präzisiert durch die Anwendungsfälle 1 bis 18, sowie den klassischen Betrachtungsbereich des Service-Level-Managements, repräsentiert durch die Anwendungs-

fälle 19 bis 32, umfassen. Der äußere Ring, definiert als der erweiterte Wirkungsbereich von Service-Level-Management, nimmt in diesem Kontext eine Sonderrolle ein. Die enthaltenen Primäranwendungsfälle 33 bis 38 beziehen sich auf Anforderungen, die sehr spezifische Kommunikationsbeziehungen zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen betreffen. Aspekte des erweiterten Betrachtungsbereichs und damit der verbleibenden Anwendungsfälle 33 bis 38 werden im Kontext des Interprozess-Kommunikationsmodells der Prozess-Sicht adressiert.

Betrachtet man also die beiden inneren Segmente, in Abbildung 5.20 dunkelgrau und hellgrau eingefärbt, so lässt sich jedes davon in zwei weitere Untersegmente einteilen, angedeutet durch die gestrichelten Kreislinien innerhalb der Segmente. Jedes Untersegment enthält nur Anwendungsfälle, die eine thematisch-inhaltliche Nähe zueinander aufweisen, wie in Kürze deutlich wird. Die resultierenden vier Untersegmente bilden somit die vier primären Funktionsbereiche im Service-Level-Management und stellen damit eine erste grobe Klassifizierung innerhalb des Funktionsmodells der Prozess-Sicht dar.

- Der Funktionsbereich I beschäftigt sich mit dem Thema *Dienstinformationsmanagement*, welches für ein effektives Service-Level-Management ein essenzielles Fundament darstellt. Er umfasst die Anwendungsfälle 1 bis 3 und 7 bis 15.
- Der Funktionsbereich II umfasst das Thema *Stakeholder-Management* und beinhaltet die Anwendungsfälle 4 bis 6 und 16 bis 18.
- Der Funktionsbereich III beschäftigt sich mit allen Aspekten des *Vereinbarungsmanagements*. Ihm sind somit die Anwendungsfälle 19 bis 28 zuzuordnen.
- Der Funktionsbereich IV beschäftigt sich schließlich mit dem *Messdaten- und Berichtsmanagement* und bezieht sich auf die Anwendungsfälle 29 bis 32.

Bemerkung: Die vier Funktionsbereiche können auch als *Subprozesse* des Service-Level-Managements angesehen werden.

Innerhalb dieser vier Haupt-Funktionsbereiche wird eine weitere Zerlegung in Funktionsteilbereiche vorgenommen, die sich auf die Funktionsanforderungen F1 bis F11 abbilden lassen. Abbildung 5.21 illustriert diese Hierarchie ausschnittartig und enthält zudem eine schematische Darstellung des weiteren

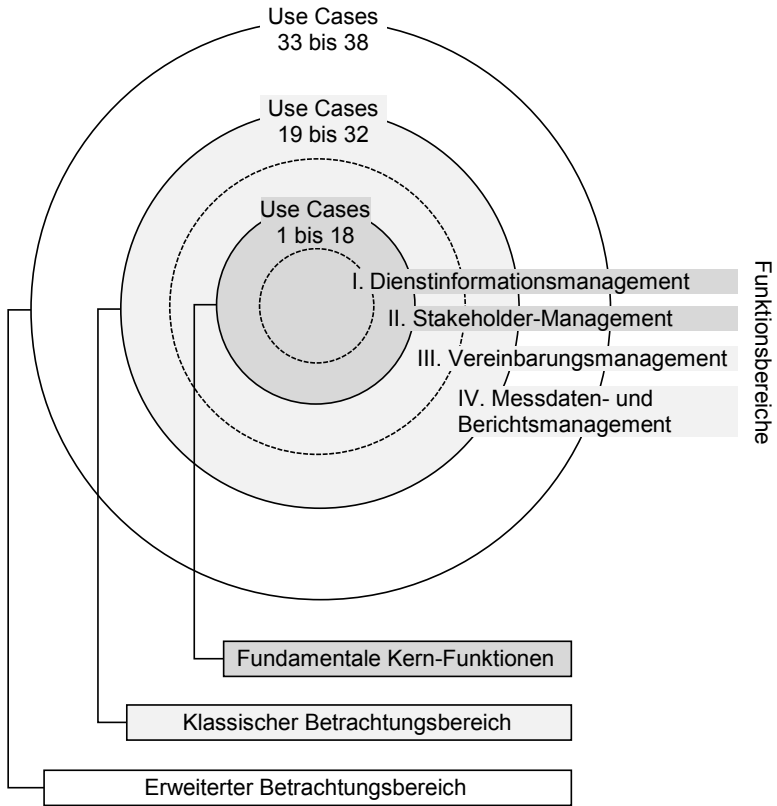


Abbildung 5.20: Grundlegende Funktionsbereiche des Service-Level-Managements und ihre Einordnung in das Funktionsspektrum

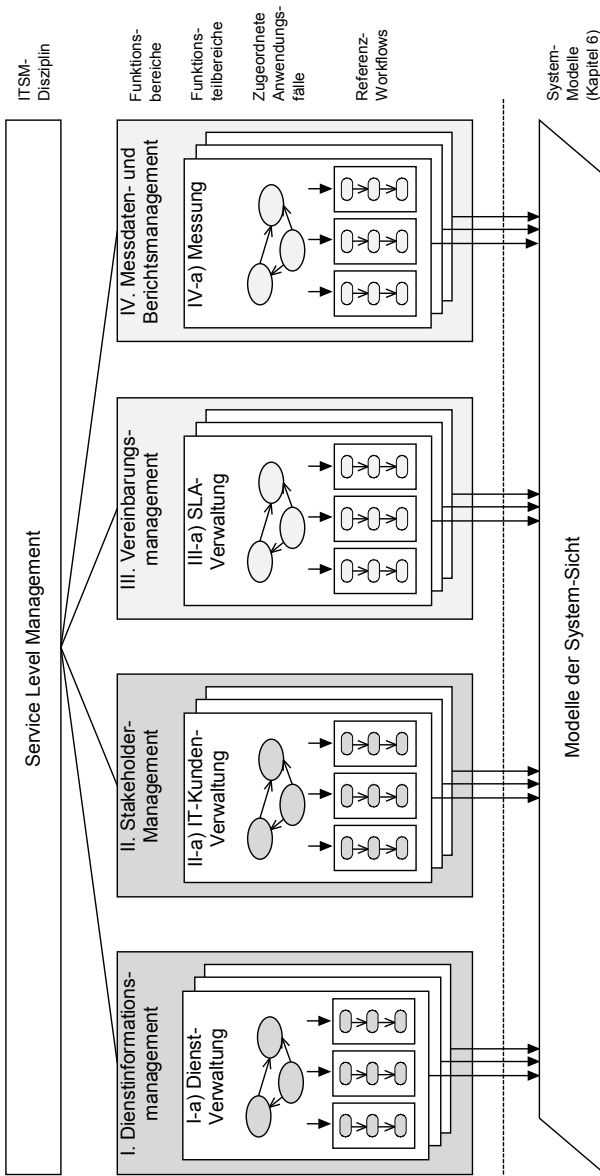


Abbildung 5.21: Darstellung der Funktionsbereiche und ihrer Teilbereiche sowie schematische Darstellung des weiteren Top-Down-Vorgehens im Rahmen des Funktionsmodells

Vorgehens im Rahmen des Funktionsmodells der Prozess-Sicht: Zu jedem Primäranwendungsfall wird ein Referenz-Workflow (Nutzfallablauf) angegeben, der eine mögliche Aktivitätenfolge im Kontext dieses Anwendungsfalls reflektiert. Die Granularität der einzelnen Workflow-Aktivitäten wird dabei so gewählt, dass Rollenwechsel im Hinblick auf das Verantwortungsmodell *operativ verantwortlich (responsible)* nicht innerhalb einer, sondern nur zwischen verschiedenen Aktivitäten vorkommen. Diese Maßgabe ist zweckmäßig, um zu gewährleisten, dass der Detailgrad der Workflows ausreichend groß ist, um zwischen Verantwortlichkeiten unterschiedlicher Akteure sinnvoll zu differenzieren und gleichzeitig zu verhindern, dass die Modellierung ein akzeptables Maß an Komplexität überschreitet.

Die Referenz-Workflows stellen sowohl Kontrollflüsse als auch Informationsflüsse dar, wobei sich letztere auf die im Rahmen des Informationsmodells identifizierten Artefakte beziehen. Darüber hinaus finden Rollen- und Verantwortungsmodelle aus dem Organisationsmodell Anwendung.

Auf der Ebene der System-Sicht der Architektur werden im Rahmen der Auseinandersetzung mit konkreten Managementfunktionen und Automatisierungsaspekten nicht mehr vorrangig die Primäranwendungsfälle, sondern die Aktivitäten aus diesen nun zu definierenden Referenz-Workflows von Interesse sein, da ja jede Workflow-Aktivität in geeigneter Weise durch ein IT-gestütztes Managementsystem unterstützt werden muss.

Bemerkung 1: Üblicherweise werden bei der Ablaufdarstellung eines Anwendungsfalles neben einem Primärszenario auch weitere denkbare Sekundärszenarien betrachtet. Dieses Kapitel beschränkt sich aus Gründen des Umfangs auf die Darstellung eines typischen Primärszenarios für jeden Anwendungsfall, geht aber, wo sich dies als sinnvoll erweist, zum Abschluss jeweils auch auf denkbare Varianten ein, ohne diese im Detail zu behandeln oder zu modellieren.

Bemerkung 2: Alle im Folgenden dargestellten Referenz-Workflows berücksichtigen die Abhängigkeiten zwischen den Primäranwendungsfällen, wie sie am Ende von Kapitel 3 tabellarisch beschrieben wurden. Das heißt konkret: Wird der Workflow eines bestimmten Anwendungsfalles eingehalten, so ist aufgrund des Workflow-Entwurfs sichergestellt, dass alle Bedingungen, die für diesen Anwendungsfall und Workflow erfüllt sein müssen, auch tatsächlich erfüllt sind. Hierzu ein kurzes Beispiel: Anwendungsfall 19 (*Close SLA*) ist abhängig von den Anwendungsfällen 22 (*Close OLA*) und 25 (*Close UC*), was bedeutet, dass eine Dienst-

vereinbarung stets vor ihrem endgültigen Abschluss und Inkrafttreten durch unterstützende Vereinbarungen abgesichert sein muss. Das wird im Referenz-Workflow zu Anwendungsfall 19 durch eine Entscheidungsraute abgebildet, in die die provisorische Dienstvereinbarung eingeht – mit dem Ziel, die Substanziierung ihrer Ziele der Dienstleistung und Service-Level-Ziele zu beurteilen und gegebenenfalls den Abschluss neuer unterstützender Vereinbarungen durch Auslösen der Workflows zu den Anwendungsfällen 22 und 25 anzustoßen.

5.3.2 Funktionsbereich I: Dienstinformationsmanagement

Ein IT-gestütztes Managementsystem für die Dienstmanagementdisziplin Service-Level-Management muss Funktionalitäten zur Verfügung stellen, die die Administration von Dienstinformationen im Zusammenhang mit dem Dienstportfolio und den Dienstkatalogen unterstützen. Folgende Sekundärfunktionsbereiche ergeben sich:

- *Dienstverwaltung*
- *Dienstkatalogverwaltung*
- *Support-Level-Verwaltung*

5.3.2.1 Dienstverwaltung

Die Verwaltung von Managementinformationen zu allen betriebenen und in Planung befindlichen Diensten bildet eine wichtige Grundlage für alle weiteren Aktivitäten im Service-Level-Management. Direkt zum Funktionsteilbereich *Dienstverwaltung* gehören die Anwendungsfälle 1 (*Create new service*), 2 (*Change service*) und 3 (*Delete service*). Im Fokus steht dabei jeweils das Informationsartefakt *Service*.

Anwendungsfall 1 – Primärszenario Abbildung 5.22 stellt den Referenz-Workflow für den ersten Anwendungsfall dar. Er beschreibt einen idealtypischen Ablauf und damit das Primärszenario.

Bemerkung: Nicht alle Aktivitäten in diesem Workflow fallen direkt in die operative Verantwortung des Service-Level-Managements. Folgende Farbsemantik kommt daher zur Anwendung:

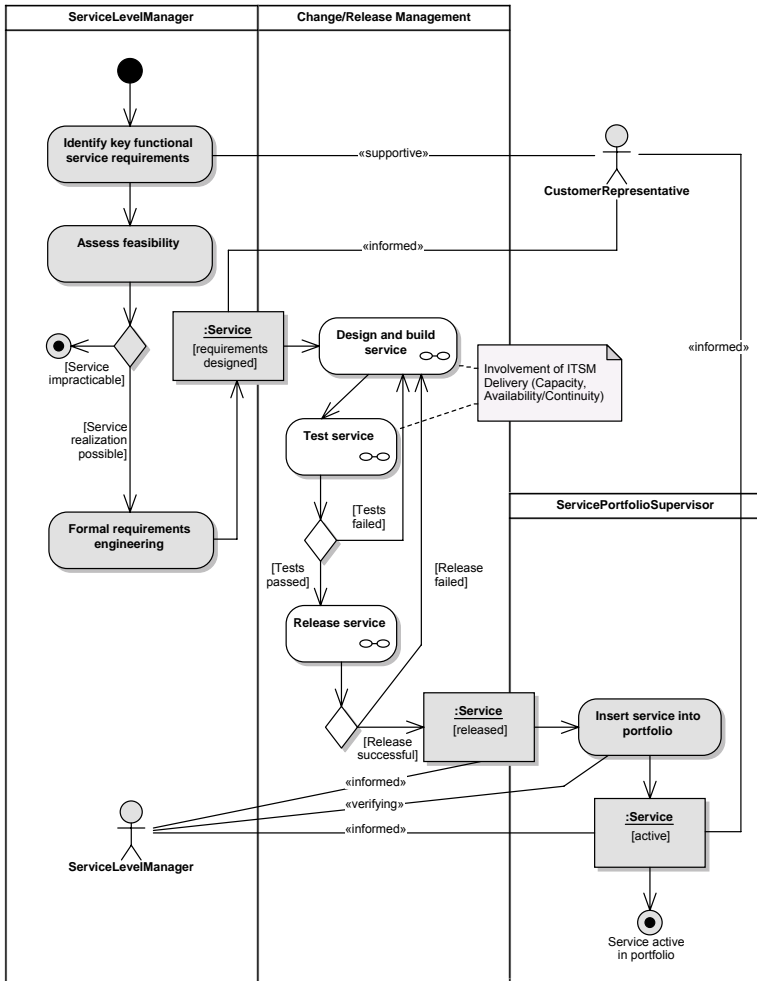


Abbildung 5.22: Referenz-Workflow 1 (Create new service)

- Hellgrau ausgefüllte Aktivität: Die Aktivität ist operativer Bestandteil des Service-Level-Managements.
- Weiß ausgefüllte Aktivität: Die Aktivität wird operativ im Rahmen anderer Dienstmanagementprozesse ausgeführt. → Wichtige Grundlage für das Interprozess-Kommunikationsmodell der Prozess- und System-Sicht

Die Aktivitäten gemäß Referenz-Workflow 1 werden nachfolgend erläutert:

- Identifikation funktionaler Dienstanforderungen (*Identify key functional service requirements*): Zunächst werden die allgemeinen funktionalen Anforderungen an den Dienst identifiziert und dokumentiert (vgl. *service functionality* im MNM-Dienstmodell), die im Wesentlichen widerspiegeln sollen, wie und zu welchem Zweck der Dienst eingesetzt werden soll und zur Unterstützung der Geschäftsprozesse einer oder mehrerer Kundendomänen beiträgt. Die Spezifikation der Informationsklasse **ServiceRequirements** des Service-Artefakts (vgl. Informationsmodell der Prozess-Sicht) beschreibt den gewünschten und erwarteten Output aus dieser Aktivität, die von Seiten eines oder mehrerer (potenzieller) Kunden des zu entwerfenden Dienstes unterstützt werden kann. Konkrete kundenspezifische Dienstleistungsanforderungen, also Kundenbedürfnisse hinsichtlich der Erbringung einer bestimmten Variante/Ausprägung sowie der Performanz des Dienstes, werden allerdings erst im Rahmen der Dienstbeauftragung durch den Kunden und dem damit verbundenen Abschluss einer Dienstvereinbarung (vgl. Anwendungsfall und Referenz-Workflow 19) ermittelt und festgelegt.
- Einschätzung der Umsetzbarkeit (*Assess feasibility*): Bevor der Dienst entwickelt werden kann, wird die technische Realisierbarkeit der identifizierten Kernanforderungen beurteilt, was im schlimmsten Fall dazu führen kann, dass der Dienst als nicht realisierbar bewertet wird (Endzustand: *Service impracticable*). Dieser Fall sollte allerdings die Ausnahme darstellen und nur dann eintreten, wenn eine vorherige Überarbeitung der Anforderungen nicht zu einem Satz realisierbarer Anforderungen geführt hat.
- Formale Anforderungsdefinition (*Formal requirements engineering*): Nachdem die funktionalen Kernanforderungen bestimmt wurden und der Dienst als grundsätzlich praktikabel bewertet wurde, müssen die konkreten Anforderungen an den Dienstentwurf, seine Implementierung

und Freigabe einschließlich Akzeptanz- und Abnahmekriterien durch das Service-Level-Management formal definiert werden. Bezogen auf das Service-Artefakt umfasst diese Aktivität sowohl eine Vervollständigung und Konkretisierung der ServiceRequirements als auch den Grobentwurf einer ersten technischen Dienstspezifikation, repräsentiert durch die Informationsklasse ServiceSpecification.

- Entwurf und Implementierung des Dienstes (*Design and build service*): Diese Aktivität umfasst alle Aufgaben von der technischen Planung bis zur Vorbereitung eines Dienstes zur Ausrollung bzw. Bereitstellung. Die *IT Infrastructure Library* widmet dem Thema *Service Design* in seiner aktuellen Version ein komplettes Buch [25], und es ist nicht das Anliegen dieser Arbeit, Aspekte des technischen Dienstentwurfs im Detail zu adressieren. Aus Sicht des Service-Level-Managements ist es wichtig, dass ein unterstützendes Managementsystem die für nachfolgende SLM-Aktivitäten relevanten Aspekte des Dienstentwurfs dokumentiert und verfügbar macht. Dies ist gewährleistet, wenn die Informationsanforderungen an das Service-Artefakt erfüllt werden.
- Testen des Dienstes (*Test service*): Vor der Aktivierung eines Dienstes im Portfolio und somit der potenziellen Bereitstellung an den oder die ersten Kunden muss der Dienst getestet werden. Dies geschieht in der Regel in einer kontrollierten Testumgebung. Erfolgreiche Tests können im Rahmen von Freigabe-Richtlinien zur Voraussetzung für die Ausrollung gemacht werden.
- Überarbeiten des Dienstentwurfs (*Redesign service*): Im Falle fehlgeschlagener Tests muss der Dienstentwurf angepasst werden, um ein reibungsloses Funktionieren des Dienstes im späteren Betrieb zu gewährleisten.
- Ausrollen und Freigabe (*Release service*): Nach erfolgreichen Tests kann der Dienst in die Live-Umgebung ausgerollt werden. Das ist nicht gleichbedeutend mit der Bereitstellung und Lieferung des Dienstes an einen Kunden. Vielmehr wird der Dienst technisch auf die Bereitstellung an einen Kunden vorbereitet.
- Aktivieren des Dienstes (*Activate service*): Die Aktivierung des Dienstes bewirkt, dass der Dienst nun im Dienstportfolio als aktiver Dienst zur Verfügung steht und damit zu einem oder mehreren Dienstkatalogen hinzugefügt werden kann. Beides ist Voraussetzung dafür, dass ein Kunde eine Dienstvereinbarung über diesen Dienst abschließen und selbigen nutzen kann.

Anwendungsfall 1 – Sekundärszenarien Neben dem in Abbildung 5.22 dargestellten Referenz-Workflow sind abweichende Abläufe möglich, die sich zwar grundsätzlich am Referenz-Workflow orientieren, jedoch untypische Vorgänge bzw. Ausnahmen oder Fehler innerhalb des Anwendungsfalls abdecken bzw. behandeln. Es wird darauf verzichtet, diese erschöpfend darzustellen. Gemäß der in der Softwareentwicklung gängigen Praxis werden nur einige beispielhafte Sekundärszenarien überblicksartig angerissen:

1. Während der Realisierung oder des Testens des neuen Dienstes durch den Provider ändern sich die funktionalen Kernanforderungen an den Dienst grundlegend. Die aktualisierten Anforderungen durchlaufen den Workflow erneut bis sie an das *Change Management* übergeben werden, wo das bisherige Dienst-Design angepasst wird.
2. Die Unterstützung der Identifikation funktionaler Dienstanforderungen durch die Rolle des Kundenvertreters kann entfallen, wenn ein Dienst ohne aktuellen Bedarf proaktiv geplant wird oder wenn es sich um einen Dienst handelt, dessen Kernanforderungen wohlbekannt sind.

Anwendungsfall 2 – Primärszenario Abbildung 5.23 stellt den Referenz-Workflow für Anwendungsfall 2 (*Change service*) dar, der dem vorigen Workflow ähnelt. Ein wichtiger Punkt aus Sicht des Service-Level-Managements ist vor allem die Beurteilung einer potenziellen Implikation einer Änderung an einem Dienst auf bestehende Dienstvereinbarungen. Nachfolgend einige Erläuterungen zu diesem zweiten Referenz-Workflow:

- Definition der Änderungsanfrage (*Define change request*): Basierend auf der Instanz des Informationsartefakts *Service*, die den zu ändernden Dienst beschreibt, werden die erforderlichen Anpassungen beschrieben und zusammen mit dem *Service*-Artefakt an das *Change Management* übergeben.
- Bewertung und Entwurf der Änderung (*Assess and design change*): Die Änderungsanfrage wird hinsichtlich verschiedener Kriterien (wie Priorität, Kategorie, Risiko) bewertet und autorisiert. Daran schließt sich unmittelbar der Entwurf der technischen Umsetzung an. Das Artefakt *ServiceTransitionPlan* (vgl. Abbildung 5.5) spielt hierbei eine wichtige Rolle.
- An dieser Stelle teilt sich der Kontrollfluss in zwei parallele Pfade, die nach Abschluss der jeweils enthaltenen Aktivitätssequenzen wieder zusammenlaufen (*Join*):

- Pfad 1 (**ServiceLevelManager**): SLA überarbeiten (*Revise SLA*): Nachdem die Änderung autorisiert und geplant wurde, muss überprüft werden, ob eine Notwendigkeit besteht, existierende Dienstvereinbarungen anzupassen. In diesem Fall wird – gegebenenfalls mehrfach – Anwendungsfall 20 (*Revise SLA*) ausgelöst.
- Pfad 2 (**Change/Release Manager**): Implementierung der Änderung (*Build change*) und Testen der Änderung (*Test change*): Diese Aktivitäten stehen in enger Analogie zu den (Teil-)Aktivitäten *Build service* und *Test service* aus dem Referenz-Workflow zu Anwendungsfall 1.
- Freigabe und Ausrollen der Änderung (*Release change*): Mit diesem Schritt wird die erfolgreich getestete Änderung in die operative Umgebung ausgerollt. Erst wenn alle tangierten Dienstvereinbarungen angepasst wurden und der Dienst erfolgreich getestet wurde, wird in diese Aktivität eingetreten. Im Falle einer erfolgreichen Ausrollung erfolgt nicht nur eine Information an den Service-Level-Manager, sondern auch die Übertragung der operativen Verantwortung für den letzten Workflow-Schritt an die Rolle des Dienstportfolioverwalters.
- Aktualisierung des Dienstportfolios (*Update service portfolio*): Es liegt abschließend in der Verantwortung des Dienstportfolioverwalters sicherzustellen, dass die Instanz des Service-Artefakts eine zutreffende Spezifikation des geänderten Dienstes enthält. Zudem wird der geänderte Dienst im Portfolio aktiviert und somit die Übernahme in Dienstkataloge ermöglicht.

Anwendungsfall 2 – Sekundärszenarien

1. Die erforderliche oder gewünschte Änderung des Dienstes steht in direktem Zusammenhang mit einem konkreten Änderungswunsch eines Kunden. Die Definition der Änderungsanfrage wird somit durch die Rolle des Kundenvertreters unterstützt.
2. Die angefragte Änderung des Dienstes wird durch das formale *Change Management* nicht autorisiert und zurückgewiesen. In diesem Fall wird die Workflow-Ausführung vorzeitig abgebrochen.

Anwendungsfall 3 – Primärszenario Der dritte und letzte Workflow des Funktionsteilbereichs *Dienstverwaltung* ist in Abbildung 5.24 dargestellt. Er bezieht

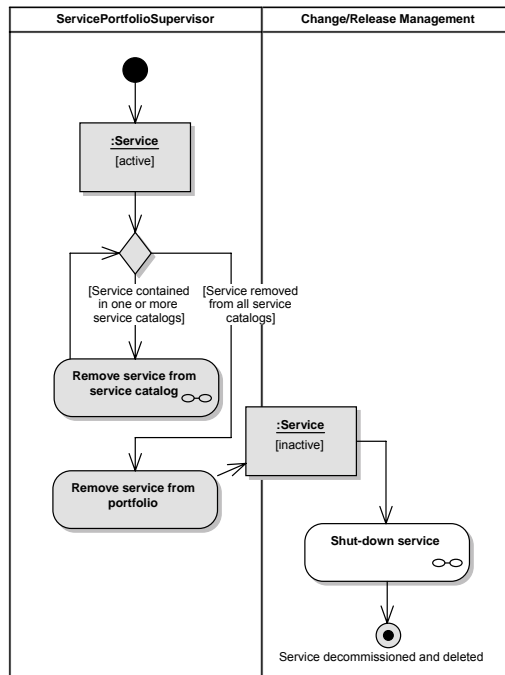


Abbildung 5.24: Referenz-Workflow 3 (*Delete service*)

sich auf Anwendungsfall 3 (*Delete service*). In diesem Ablauf wird vor allem der Zusammenhang zwischen der Außerbetriebnahme eines Dienstes und bestehenden Dienstkatalogen und Dienstvereinbarungen, die diesen Dienst beinhalten bzw. die sich auf diesen Dienst beziehen, berücksichtigt.

- Dienst aus Dienstkatalog entfernen (*Remove service from service catalog*): Wenn ein Dienst außer Betrieb genommen und aus dem Portfolio entfernt werden soll, darf er nicht mehr in Dienstkatalogen enthalten sein. Da vor dem Entfernen eines Dienstes aus einem Dienstkatalog (vgl. Anwendungsfall 12 und zugehöriger Workflow) wiederum sichergestellt wird, dass keine aktiven Dienstvereinbarungen über diesen Dienst bestehen, ist diese Bedingung hier implizit mit abgedeckt. Denn ein SLA über einen Dienst kann gemäß der definierten Abhängigkeiten zwischen den Anwendungsfällen nur so lange bestehen wie der Dienst in einem

gültigen Dienstkatalog enthalten ist.

- Dienst aus Dienstportfolio entfernen (*Remove service from portfolio*): Wenn keine Dienstvereinbarungen mehr über den zu terminierenden Dienst bestehen und dieser auch keinem Kunden mehr über einen Dienstkatalog angeboten wird, kann er aus dem Portfolio entfernt werden. Dabei wird die betreffende Instanz des Service-Artefakts nicht gelöscht, sondern auf einen inaktiven Status gesetzt.
- Dienst außer Betrieb nehmen (*Shut-down service*): Die eigentliche technische Außerbetriebnahme des Dienstes erfolgt schließlich, nachdem der Dienst im Dienstportfolio deaktiviert wurde. Diese Aktivität umfasst beispielsweise den Abbau von Komponenten, die Deinstallation von Software, Rückgabe von Lizenzen, Information der Support-Gruppen und vieles andere mehr – abhängig von der Art des Dienstes – und liegt nicht mehr in der operativen Verantwortung des Service-Level-Managements und einer seiner Rollen.

5.3.2.2 Dienstkatalogverwaltung

Der Funktionsteilbereich der *Dienstkatalogverwaltung* setzt direkt auf der *Dienstverwaltung* auf. Während zuvor nur die Administration von IT-Diensten adressiert wurde, wird nun betrachtet, wie auf dieser Grundlage Dienstkataloge definiert und verwaltet werden können. Hierzu gehören die Anwendungsfälle 7 (*Initialize service catalog*), 8 (*Assign catalog to customer*), 9 (*Dissociate catalog from customer*) und 10 (*Delete service catalog*) sowie 11 (*Add service to catalog*) und 12 (*Remove service from catalog*). Das zentrale Informationsartefakt in diesem Teilbereich ist der **ServiceCatalog** vom Typ Datenbank-Sicht, der jedoch stets im Zusammenhang mit den Artefaktklassen **Service** und **Customer** zu sehen ist, wie auch die nachfolgend dargestellten Referenz-Workflows zeigen.

Bemerkung: Die Anwendungsfälle 4 bis 6 fallen in den Funktionsbereich II – Teilbereich *Kundenverwaltung* und werden später behandelt.

Anwendungsfall 7 – Primärszenario Die Erstellung und Initialisierung eines neuen Dienstkatalogs umfasst in erster Linie das Anlegen einer neuen Instanz des Informationsartefakts **ServiceCatalog**. Zu diesem Zeitpunkt beinhaltet der

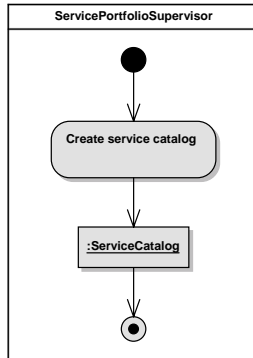
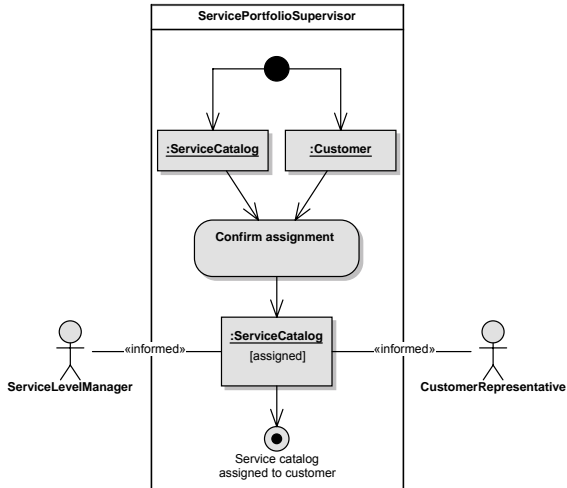


Abbildung 5.25: Referenz-Workflow 7 (*Initialize service catalog*)

Dienstkatalog noch keine Dienste und ist auch noch nicht mit einem Kunden assoziiert. Der sehr einfache Referenz-Workflow zu diesem Anwendungsfall wird in Abbildung 5.25 dargestellt.

Anwendungsfall 8 – Primärszenario Ein bestehender Dienstkatalog kann einem oder mehreren Kunden zugewiesen werden. Dies ist Gegenstand von Anwendungsfall 8, wobei der Referenz-Workflow hierzu die Information des Service-Level-Managers als auch des Kundenvertreters über den neu assoziierten Dienstkatalog vorsieht. Die eigentliche Zuweisung erfolgt durch die Herstellung einer logischen Verbindung zwischen je einer Instanz des **ServiceCatalog**-Artefakts und des **Customer**-Artefakts.

Anwendungsfall 9 – Primärszenario Während vor der Zuweisung eines Dienstkatalogs an einen Kunden keine spezifischen Vorbedingungen erfüllt sein müssen, muss vor der Aufhebung der Gültigkeit eines Dienstkatalogs für einen Kunden sichergestellt sein, dass keine Dienstvereinbarungen mit diesem Kunden auf den Dienstkatalog und seine enthaltenen Dienste referenzieren. Um diese Bedingung zu erfüllen, müssen bestehende Dienstvereinbarungen gegebenenfalls aufgehoben werden. Abbildung 5.27 zeigt, wie aus dem Workflow zur Dissoziation eines Dienstkatalogs von einem Kunden unter Umständen der Anwendungsfall 21 (*Resign SLA*) ausgelöst wird.

Abbildung 5.26: Referenz-Workflow 8 (*Assign catalog to customer*)

Anwendungsfall 10 – Primärszenario Das Löschen eines Dienstkatalogs setzt voraus, dass dieser keine Gültigkeit für einen Kunden hat. Der in Abbildung 5.28 dargestellte Workflow berücksichtigt diese Voraussetzung, bevor die Aktivität *Delete service catalog* zur Ausführung kommt.

Anwendungsfall 11 – Primärszenario Das Hinzufügen eines Dienstes zu einem Dienstkatalog wird durch Verknüpfung des jeweiligen Service-Artefakts und des ServiceCatalog-Artefakts, die im Rahmen der Aktivität *Link service with catalog* vorgenommen wird. Über die erfolgreiche Umsetzung werden sowohl der Service-Level-Manager als auch der betreffende Kundenvertreter informiert. Insbesondere letzteres ist wichtig, um den Kunden über den neu für ihn verfügbaren Dienst zu benachrichtigen. Abbildung 5.29 stellt den Referenz-Workflow dar.

Anwendungsfall 12 – Primärszenario Um einen Dienst aus einem Dienstkatalog zu entfernen, muss die Assoziation zwischen den beiden betreffenden Informationsartefakten *Service* und *ServiceCatalog* aufgelöst werden. Voraussetzung hierfür ist, dass keine Dienstvereinbarung über den Dienst mit einem der Kunden besteht, für die der Dienstkatalog gültig ist. Der in Abbildung

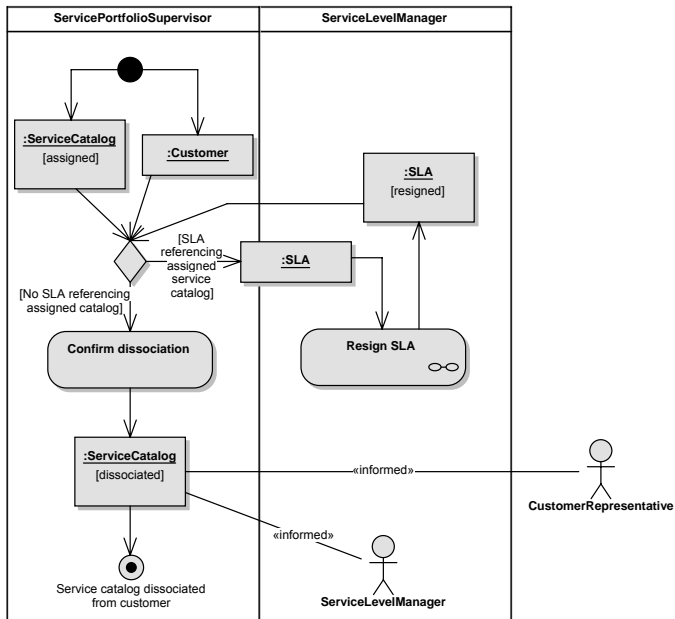


Abbildung 5.27: Referenz-Workflow 9 (*Dissociate catalog from customer*)

5.30 illustrierte Referenz-Workflow berücksichtigt diesen essenziellen Zusammenhang zwischen Anwendungsfall 12 und Anwendungsfall 21 (*Resign SLA*). Nachdem ein Dienst aus einem Dienstkatalog entfernt wurde, werden die Kunden, für die der Dienstkatalog gültig ist, über diese Änderung des für sie verfügbaren Dienstangebots informiert.

5.3.2.3 Support-Level-Verwaltung

Die Primäranwendungsfälle 13 (*Create support level profile*), 14 (*Change support level profile*) und 15 (*Delete support level profile*) bilden den dritten und letzten Teilbereich des *Dienstinformationsmanagements*, die *Support-Level-Verwaltung*. Hinter dem Konzept eines Support-Level-Profils steht die Idee, das Niveau der Dienstunterstützung von den eigentlichen Dienstvereinbarungen und Diensten zu entkoppeln und entsprechende Profile, die konkrete Ausprägungen verschiedener Support-Parameter zusammenfassen, unabhängig und

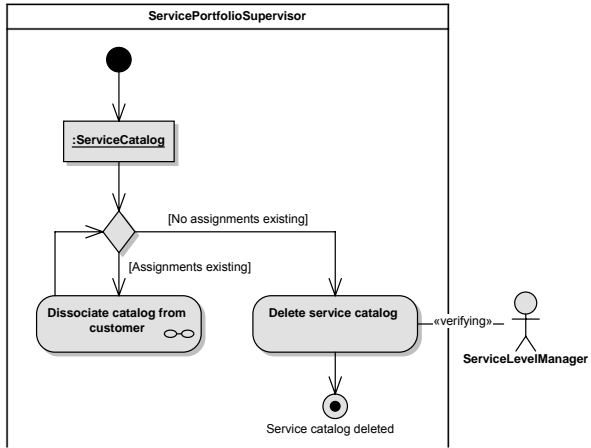


Abbildung 5.28: Referenz-Workflow 10 (*Delete service catalog*)

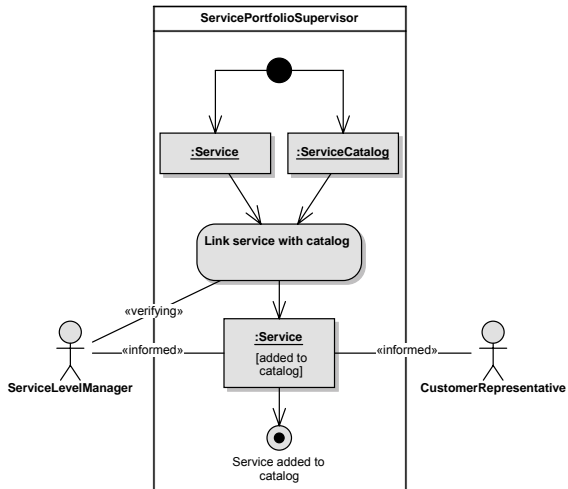


Abbildung 5.29: Referenz-Workflow 11 (*Add service to catalog*)

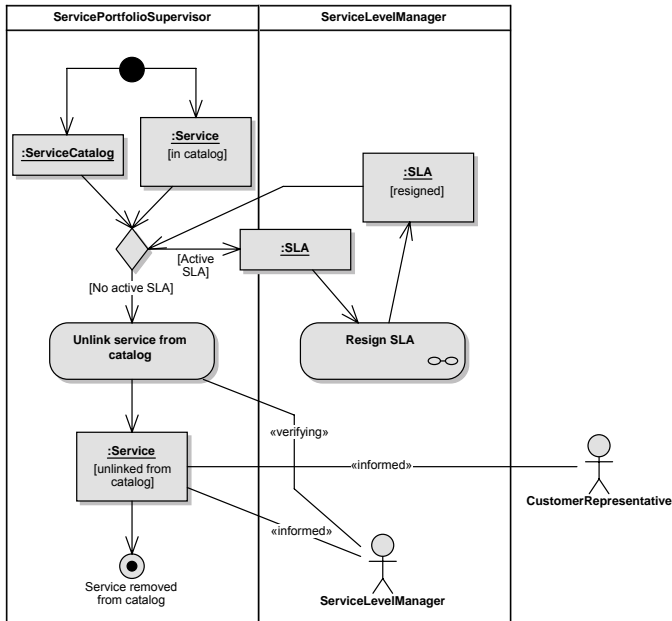
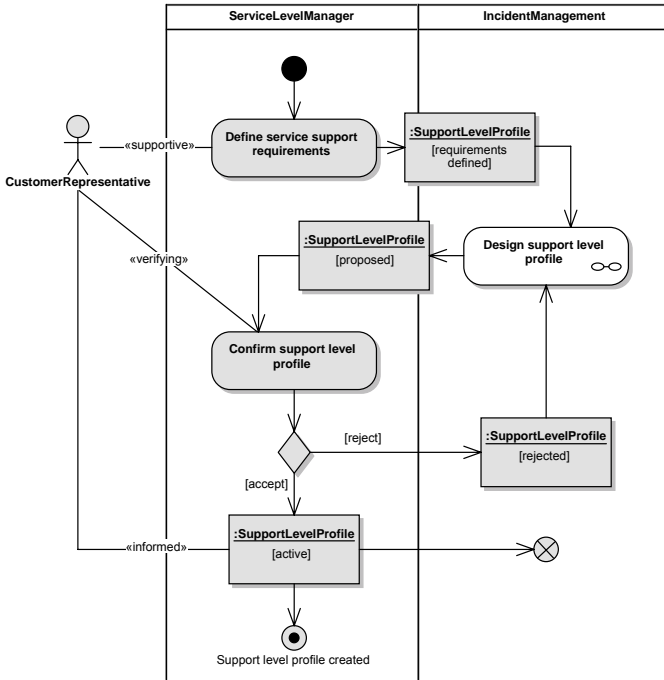


Abbildung 5.30: Referenz-Workflow 12 (*Remove service from catalog*)

eigenständig zu administrieren, um auf diese aus Dienstvereinbarungen heraus zu referenzieren. Entsprechend steht das Informationsartefakt **SupportLevel-Profile** im Mittelpunkt dieses Funktionsteilbereichs und seiner Workflows.

Anwendungsfall 13 – Primärszenario Der in Abbildung 5.31 dargestellte Referenz-Workflow geht davon aus, dass neben dem Service-Level-Manager als hauptsächlich operativ verantwortliche Rolle zusätzlich der Prozessmanager des *Incident Management*-Prozesses und ein Kundenvertreter in den Entwurf eines neuen Support-Level-Profils eingebunden sind. Dabei liefert der Kundenvertreter die maßgeblichen Anforderungen an die Dienstunterstützung, auf deren Basis der Service-Level-Manager ein entsprechendes **SupportLevel-Profile**-Artefakt initialisiert.

Wichtig ist allerdings, dass die Definition des Support-Level-Profils in der Verantwortung des *Incident Managements* liegt – ganz ähnlich wie die Planung,

Abbildung 5.31: Referenz-Workflow 13 (*Create support level profile*)

Implementierung und Freigabe neuer oder geänderter Dienste im Kontext der Anwendungsfälle 1 und 2 vom *Change Management* verantwortet wird.

Anwendungsfall 14 – Primärszenario Die Änderung eines Support-Level-Profils gleicht in den meisten Punkten dem Workflow zur Erstellung, wobei zwischen der Akzeptanz des in Verantwortung des *Incident Managements* geänderten Support-Level-Profils durch den Service-Level-Manager und dem Abschluss des Workflows einschließlich Information der Kundenvertreter überprüft wird, ob die Änderungen Einfluss auf bestehende Dienstvereinbarungen haben. In diesem Fall wird der Workflow zum Anwendungsfall 20 (*Revise SLA*) angestoßen, um die betreffenden Dienstvereinbarungen anzupassen. Abbildung 5.32 zeigt den Referenz-Workflow zu Anwendungsfall 14.

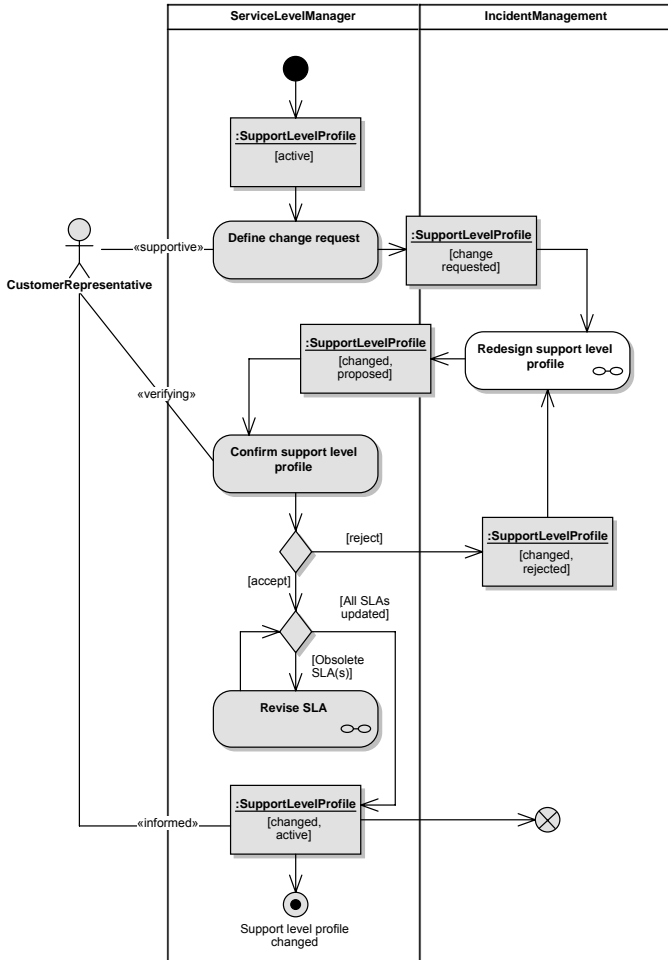


Abbildung 5.32: Referenz-Workflow 14 (*Change support level profile*)

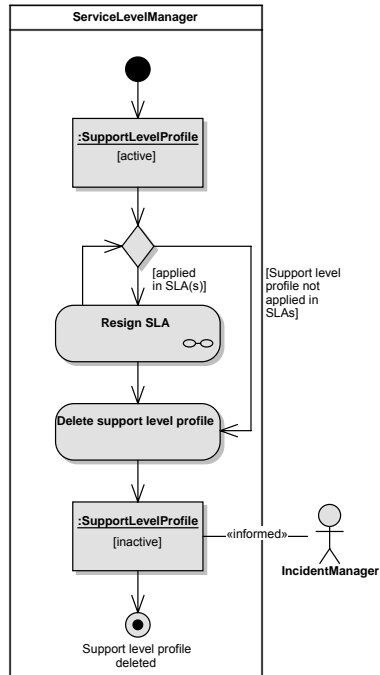


Abbildung 5.33: Referenz-Workflow 15 (*Delete support level profile*)

Anwendungsfall 15 – Primärszenario Abbildung 5.33 illustriert den Referenz-Workflow zu Anwendungsfall 15, der berücksichtigt, dass das Löschen eines Support-Level-Profiles nur ermöglicht werden darf, wenn keine Dienstvereinbarungen (mehr) existieren, die sich auf dieses beziehen.

5.3.3 Funktionsbereich II: Stakeholder-Management

Der zweite Hauptfunktionsbereich, das *Stakeholder-Management*, kann in die folgenden beiden Teilbereiche eingeteilt werden:

- *Kundenverwaltung*
- *Dienstbringerverwaltung*

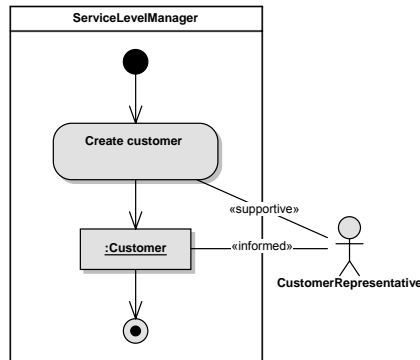


Abbildung 5.34: Referenz-Workflow 4 (*Create new customer*)

5.3.3.1 Kundenverwaltung

Die Anwendungsfälle 4 (*Create new customer*), 5 (*Change customer*) und 6 (*Delete customer*) bilden den Funktionsteilbereich der *Kundenverwaltung*, der für die Bereitstellung von Dienstkatalogen und damit auch für den Abschluss von Dienstvereinbarungen benötigt wird. Es handelt sich um einen vergleichsweise wenig komplexen funktionalen Teilbereich, da im Wesentlichen SLM-relevante Informationen zum Kundenumfeld gemanagt werden müssen. Welche Informationen das konkret sind, wurde bereits durch das Informationsmodell geklärt. Zentrales Informationsartefakt ist natürlich das **Customer**-Artefakt.

Die Abbildungen 5.34, 5.35 und 5.36 illustrieren nun die drei zugehörigen Referenz-Workflows für diesen Funktionsteilbereich. Da alle diese Workflows eine einfache Struktur aufweisen und nie mehr als zwei Rollen des Organisationsmodells involviert sind, kann an dieser Stelle auf eine differenzierte Beschreibung jeder Workflow-Aktivität und einer Abgrenzung zwischen Primär- und Sekundärszenarien verzichtet werden. Es folgen einige Erläuterungen im Hinblick auf die durch die Ablaufdiagramme repräsentierten Primärszenarien.

Anwendungsfälle 4 und 5 – Primärszenarien Beim Anlegen eines Kunden-Datensatzes (*Create customer*) im Rahmen von Anwendungsfall 4 wird ein Mindestmaß an für das Service-Level-Management zu einem Kunden relevanten Informationen gemäß Informationsmodell erfasst und im Informationsartefakt **Customer** gespeichert. Während der Service-Level-Manager für dieser Aktivität operativ verantwortlich ist, ist er zugleich auf die Unterstützung

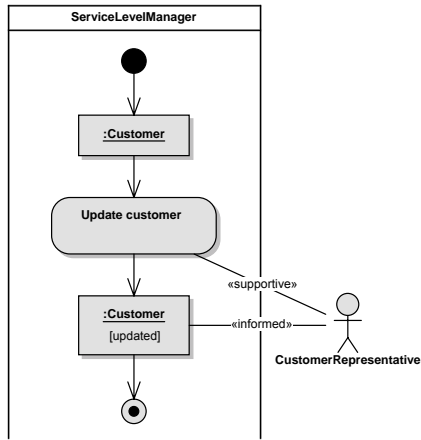


Abbildung 5.35: Referenz-Workflow 5 (*Change customer*)

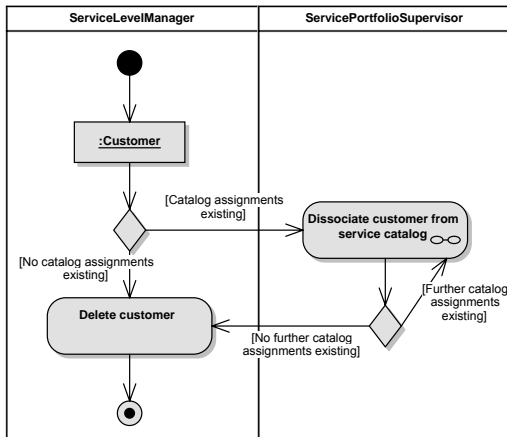


Abbildung 5.36: Referenz-Workflow 6 (*Delete customer*)

durch den betreffenden Kundenvertreter angewiesen, der einen Großteil der benötigten Informationen bereitstellen muss. Analoges gilt für die Aktivität *Update customer* des Anwendungsfalls 5.

Anwendungsfall 6 – Primärszenario Vor dem Löschen eines Kunden-Datensatzes (*Delete customer*) müssen alle bestehenden Zuweisungen von Dienstkatalogen an diesen Kunden aufgehoben werden. Die entsprechende Aktivität *Dissociate customer from service catalog* löst den Workflow zum Anwendungsfall 9 aus und ist daher als *composite activity* gekennzeichnet.

5.3.3.2 Diensterbringerverwaltung

Der zweite Funktionsteilbereich im *Stakeholder-Management* umfasst die Primäranwendungsfälle 16 (*Create delivery party*), 17 (*Change delivery party data*) und 18 (*Delete delivery party*). Sie realisieren die Anforderung F3 des Funktionsspektrums, die fordert, dass Informationen zu internen IT-Fachabteilungen sowie externen Lieferanten vorgehalten werden müssen, also zu den „Vertragspartnern“ des Providers im Zusammenhang mit internen operativen Vereinbarungen (OLAs) und externen Lieferantenverträgen (UCs). Die Informationsartefakte *InternalDeliveryParty* und *ExternalSupplier* stehen im Fokus dieses Funktionsteilbereichs.

Anwendungsfälle 16 und 17 – Primärszenarien Die Referenz-Workflows zu diesen Anwendungsfällen, dargestellt in den Abbildungen 5.37 und 5.38, differenzieren zwischen dem Anlegen bzw. Ändern einer internen Diensterbringerpartei und eines externen Zulieferers. Sie sind strukturell und semantisch ähnlich – mit dem Unterschied, dass Workflow 16 eine neue Instanz eines Informationsartefakts erzeugt, während Workflow 17 auf einer bestehenden Änderungen durchführt.

Anwendungsfall 18 – Primärszenario Beim Löschen eines Diensterbringers müssen gemäß Abhängigkeitsgraph aus der Anforderungsanalyse die Abhängigkeiten des Anwendungsfalls 18 von den Anwendungsfällen 24 (*Resign OLA*) und 27 (*Resign UC*) berücksichtigt werden. Der in Abbildung 5.39 dargestellte Referenz-Workflow garantiert das.

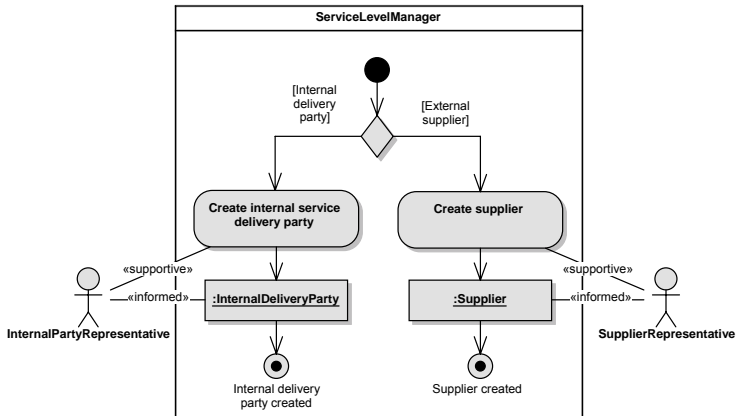


Abbildung 5.37: Referenz-Workflow 16 (*Create delivery party*)

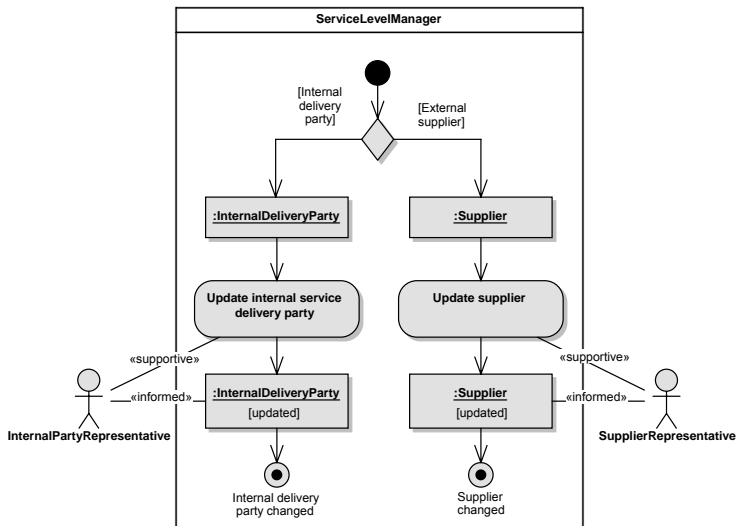


Abbildung 5.38: Referenz-Workflow 17 (*Change delivery party data*)

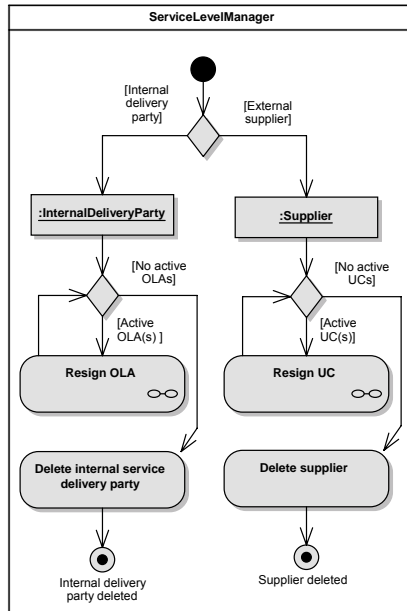


Abbildung 5.39: Referenz-Workflow 18 (Delete delivery party)

5.3.4 Funktionsbereich III: Vereinbarungsmanagement

Dieses Unterkapitel führt den Hauptfunktionsbereich III, das *Vereinbarungsmanagement*, ein, in dem es darum geht, einige der Kernaufgabenbereiche des Service-Level-Managements zu adressieren und primäre SLM-Zielsetzungen (vgl. Abschnitt 3.2.1) zu unterstützen. Er umfasst die folgenden Funktionsbereiche:

- *SLA-Verwaltung*
- *OLA-Verwaltung*
- *UC-Verwaltung*
- *Konfliktanalyse*

5.3.4.1 SLA-Verwaltung

Der Funktionsteilbereich der *SLA-Verwaltung* umfasst die zentralen Primäranwendungsfälle 19 (*Close SLA*), 20 (*Revise SLA*) und 21 (*Resign SLA*), die den Lebenszyklus von Dienstvereinbarungen kennzeichnen. Dieser Lebenszyklus ist nicht mit dem generischen Dienstlebenszyklus (vgl. Abschnitt 3.2.2) zu verwechseln. So fallen Abschluss und Änderung von Dienstvereinbarungen (Anwendungsfälle 19 und 20) in die initiale bzw. die einer Dienstanpassung nachgelagerte Verhandlungsphase eines Dienstes, während die Aufhebung einer Dienstvereinbarung in die Dienstlebenszyklusphase der Auflösung fällt. Im Mittelpunkt dieses Funktionsteilbereichs steht das Informationsartefakt `ServiceLevelAgreement` des Informationsmodells der Prozess-Sicht.

Anwendungsfall 19 – Primärszenario Abbildung 5.40 leitet die Auseinandersetzung mit dem Funktionsteilbereich *SLA-Verwaltung* mit der Definition eines Referenz-Workflows zu Anwendungsfall 19 ein. Darin enthalten sind folgende Aktivitäten:

- Dienstauswahl (*Select service from service catalog*): Der Dienst, über dessen Bereitstellung eine Dienstvereinbarung abgeschlossen werden soll, wird aus einem für den jeweiligen Kunden gültigen Dienstkatalog ausgewählt. Dies liegt in der Verantwortung des Kundenvertreters, der einen neuen Dienst beauftragen möchte.
- Initialisierung einer neuen Dienstvereinbarung (*Initialize new SLA*): In der operativen Verantwortung des Providers liegt es, einen ersten zwar bereits auf den konkret angefragten Dienst und den anfragenden Kunden ausgerichtete, aber im Hinblick auf die Diensterbringung und den zu realisierenden Leitungswert generische Standard-Dienstvereinbarung zu generieren, indem eine Art *Default*-Instanz des `ServiceLevelAgreement`-Artefakts initialisiert wird.
- Verhandlung der Inhalte (*Negotiate contents*): Ausgehend von dieser Standard-Dienstvereinbarung müssen in Verhandlungen konkrete Ausprägungen für alle Bestandteile der Dienstvereinbarung geführt werden. Das Informationsmodell gibt mit den Klassen an Informationsanforderungen an das Artefakt `ServiceLevelAgreement` vor, welches diese Bestandteile und die im Rahmen dieser Aktivität zu berücksichtigenden Aspekte sind. Die eigentliche Verhandlung findet zwischen Service-Level-Manager und Kundenvertreter statt, wobei durch den Service-Level-Manager aus

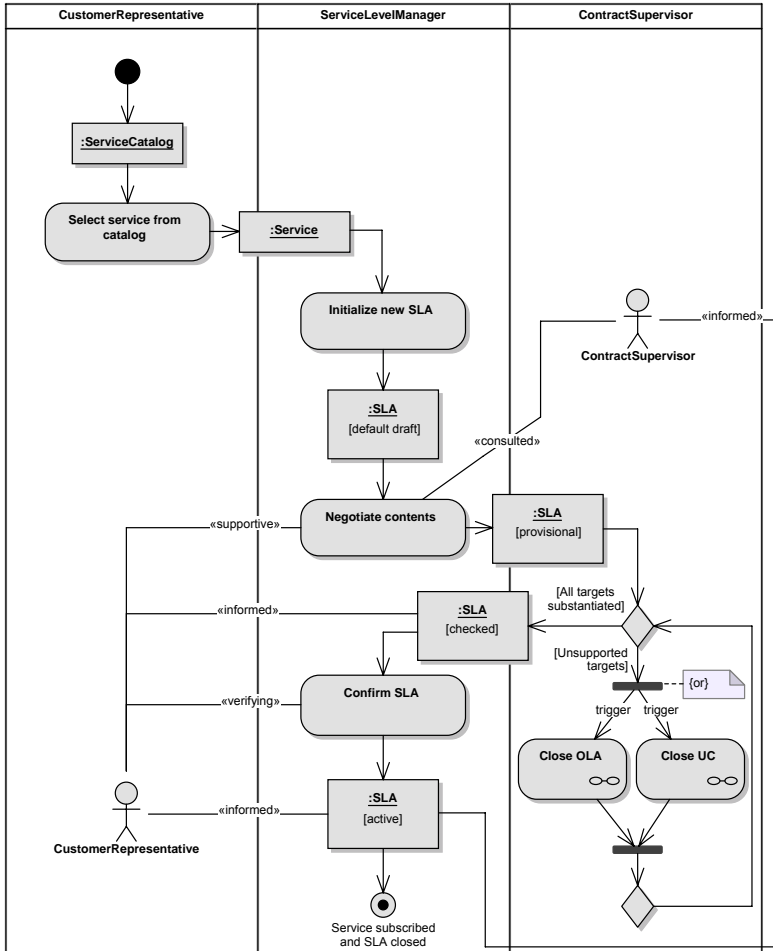


Abbildung 5.40: Referenz-Workflow 19 (*Close SLA*)

den Rollen des Service-Level-Managements mindestens noch der Kontraktverwalter beratend hinzugezogen werden kann, um bestehende operative Vereinbarungen und Lieferantenverträge frühzeitig zu berücksichtigen und die entstehende Dienstvereinbarung sinnvoll auf diese auszurichten. Hier wird auch deutlich, dass die Trennung zwischen den beiden Rollen Service-Level-Manager und Kontraktverwalter ein Stück weit die Idee des Vier-Augen-Prinzips verfolgt – wobei eine Ausfüllung der Rollen in Personalunion natürlich nie ausgeschlossen werden kann. Das Ergebnis der Aktivität muss eine prinzipiell vollständige Dienstvereinbarung sein, die zur Überprüfung an den Kontraktverwalter übergeben wird.

- Bestätigung der Dienstvereinbarung (*Confirm SLA*): Eine Dienstvereinbarung wird durch den Kontraktverwalter zurück in die operative Verantwortung des Service-Level-Managers überführt, wenn sichergestellt ist, dass sie sich in das bestehende Geflecht von Verträgen und Vereinbarungen integrieren lässt, ohne maßgebliche Konflikte zu erzeugen. Gegebenenfalls annotiert der Kontraktverwalter aber auch erforderliche Anpassungen im bisherigen Entwurf, damit dieses Ziel erreicht wird. Das so überprüfte *ServiceLevelAgreement*-Artefakt fließt in die Aktivität der Bestätigung ein, die nicht nur vom Service-Level-Manager durchgeführt, sondern auch vom Kundenvertreter verifiziert werden muss. Erst dadurch wird die Dienstvereinbarung abschließend finalisiert und zum vorgesehenen Termin aktiviert, was auch die Bereitstellung des Dienstes impliziert.

Anwendungsfall 19 – Sekundärszenarien Zu diesem Anwendungsfall sind grundsätzlich eine Reihe von Sekundärszenarien denkbar, die eine Abweichung vom Referenz-Workflow bedingen oder sinnvoll machen können:

1. Die Verhandlungen zwischen Kundenvertreter und Service-Level-Manager über die Dienstbeauftragung scheitern, und es kommt nicht zum erforderlichen Ergebnis eines vorläufigen Entwurfs der Dienstvereinbarung zur Überprüfung durch den Kontraktverwalter. In diesem Fall wird eine Funktion zur *hierarchischen Eskalation* des Vorgangs an einer dem Service-Level-Manager weisungsbefugten Person innerhalb der Organisation (zum Beispiel zum *Chief Information Officer*) benötigt.
2. Der gegebenenfalls nach der Überprüfung durch den Kontraktverwalter angepasste Entwurf der Dienstvereinbarung wird durch den Kundenvertreter nicht akzeptiert. In diesem Fall erfolgt eine Rückkehr von der Aktivität *Confirm SLA* in die Aktivität *Negotiate contents*.

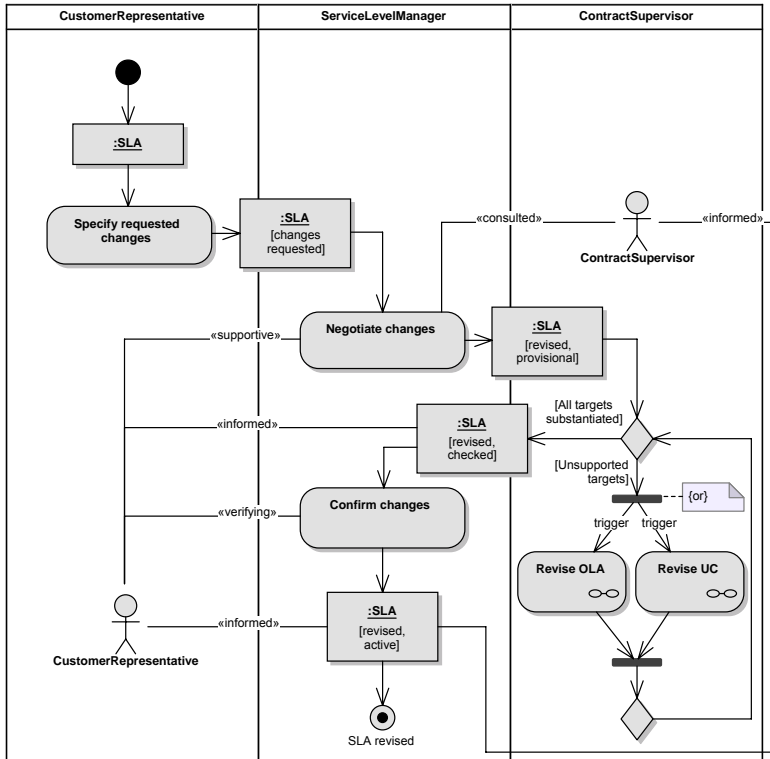
3. Die Dienstanforderung durch den Kunden ist besonders zeitkritisch. Unter Zustimmung durch den Kundenvertreter tritt die Standard-Dienstvereinbarung gegebenenfalls unter Berücksichtigung einiger weniger Kern-Anforderungen in Kraft, bevor ausführliche Verhandlungen geführt und eine Überprüfung durch den Kontraktverwalter durchgeführt werden. Die Vorgaben hinsichtlich Nutz- und Leistungswert werden so lange als nicht bindend betrachtet, bis die übersprungenen Prozessaktivitäten nachgeholt und die Dienstvereinbarung zum regulären Abschluss gebracht wurde.

Anwendungsfall 20 – Primärszenario Der Referenz-Workflow zu Anwendungsfall 20 (*Revise SLA*) weist eine im direkten Vergleich zu Anwendungsfall 19 (*Close SLA*) analoge und durch vergleichbare Aktivitäten geprägte Struktur auf. Abbildung 5.41 zeigt diesen Workflow, der vor diesem Hintergrund keiner zusätzlichen Erläuterung bedarf.

Anwendungsfall 21 – Primärszenario Wie die Abbildung 3.12 im Kapitel 3 zeigt, ist Anwendungsfall 21 (*Resign SLA*) im Abhängigkeitsgraph nahezu eine Senke – mit Ausnahme der trivialen Abhängigkeit von Anwendungsfall 19 (*Close SLA*), die ausdrückt, dass nur vorhandene, abgeschlossene Dienstvereinbarungen aufgehoben werden können. Dass keine weiteren Abhängigkeiten zu berücksichtigen sind, erklärt den einfachen Workflow zur Auflösung von Dienstvereinbarungen, der in Abbildung 5.42 dargestellt ist. Unter der operativen Verantwortung des Service-Level-Managements für den gesamten Workflow muss der betreffende Kundenvertreter die Aufhebung einer Dienstvereinbarung bestätigen (verifizieren), und sowohl der Kontraktverwalter als auch der *Change Manager* müssen über die Terminierung der Dienstvereinbarung und somit auch der verbundenen Diensterbringung informiert werden.

5.3.4.2 OLA-Verwaltung

Die Primäranwendungsfälle 22 (*Close OLA*), 23 (*Revise OLA*) und 24 (*Resign OLA*) bilden den Funktionsteilbereich der *OLA-Verwaltung*. In diesem Aufgabenbereich des Service-Level-Managements geht es darum, operative Vereinbarungen mit an der Diensterbringung beteiligten Provider-internen Parteien vom Abschluss über Änderungszyklen bis hin zur Auflösung zu verwalten – ganz ähnlich also zum Funktionsteilbereich *SLA-Verwaltung*. Das Informationsartefakt **OperationalLevelAgreement** steht hierbei stets im Vordergrund.

Abbildung 5.41: Referenz-Workflow 20 (*Revise SLA*)

Anwendungsfälle 22 bis 24 – Primärszenarien Die Referenz-Workflows zu diesen Anwendungsfällen sind in den Abbildungen 5.43, 5.44 und 5.45 abgebildet. Die enthaltenen Aktivitäten und ihre Semantik sind eng verwandt mit den Aktivitäten aus den Workflows zum SLA-Lebenszyklus, also aus den Anwendungsfällen 19 bis 21. Daher kann auf eine detaillierte Beschreibung der einzelnen Primärszenarien verzichtet werden.

Die beiden umfangreichsten Aktivitäten dieses Funktionsteilbereichs sind die Verhandlung der OLA-Inhalte (*Negotiate contents*) im Kontext von Workflow 22 sowie die Verhandlung von Änderungen an diesen Inhalten (*Negotiate changes*) im Rahmen von Workflow 23. Beide Aktivitäten verfolgen das Ziel

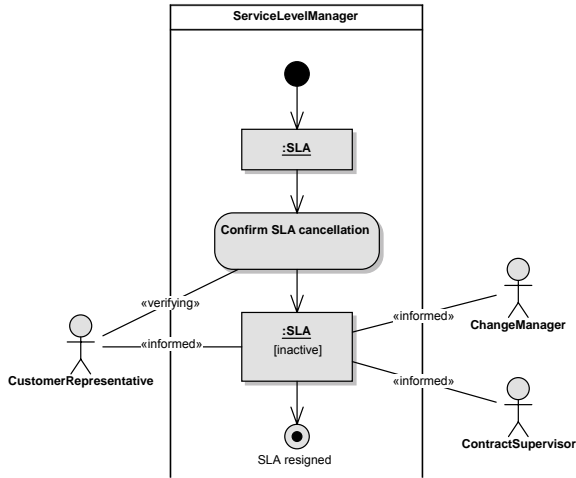


Abbildung 5.42: Referenz-Workflow 21 (*Resign SLA*)

festzulegen, welche Funktionalitäten, technischen Systeme oder Dienstleistungen durch die interne Dienstbringerpartei erbracht werden müssen. Entscheidender Unterschied im Vergleich zu Zielen der Dienstbringung, wie sie im Rahmen von Dienstvereinbarungen mit IT-Kunden festgelegt werden, ist, dass sich operative Vereinbarungen nicht auf Dienste aus dem Dienstportfolio (vgl. Informationsartefakt **ServicePortfolio**) beziehen müssen. Einfache Beispiele für operative Ziele der Dienstbringung sind die Bereitstellung eines 10-Gigabit-Ethernet-Links an einem festgelegten Netzknoten oder der Betrieb eines E-Mail-Servers mit POP3- und IMAP-Funktionalität und weiteren festgelegten Funktions-/Leistungsmerkmalen. An dieser Stelle sei noch einmal darauf hingewiesen, dass das Grundkonzeptmodell nicht zwangsläufig die Administration eines Subdienstportfolios pro interner Dienstbringerpartei vorsieht. Die Möglichkeit einer entsprechenden Erweiterung wurde aber im Abschnitt 5.2.3.2 des Informationsmodells der Prozess-Sicht diskutiert. Ihre Umsetzung muss sich nicht zwangsläufig auf die Referenz-Workflows zur OLA-Verwaltung auswirken, wohl aber auf einzelne Aktivitäten und ihre Durchführung.

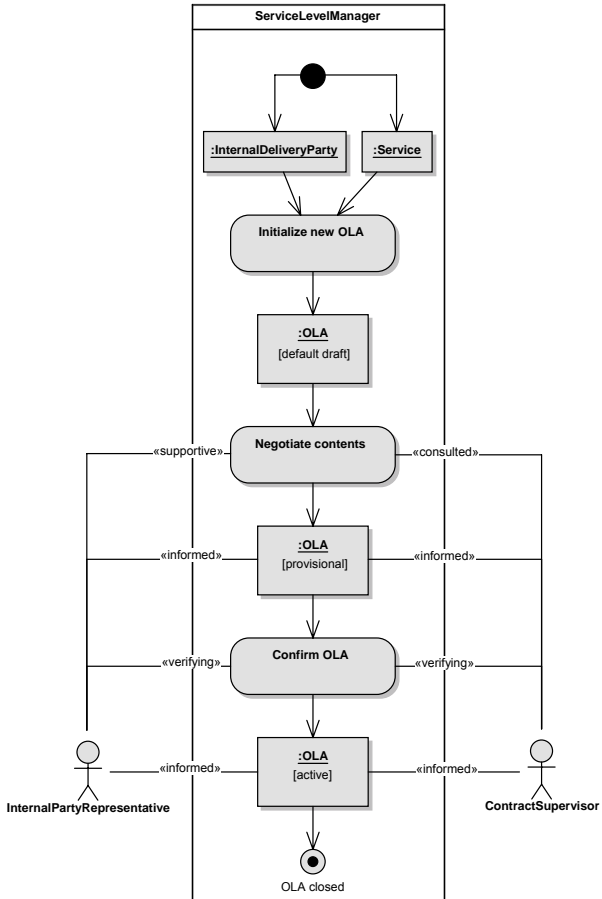


Abbildung 5.43: Referenz-Workflow 22 (*Close OLA*)

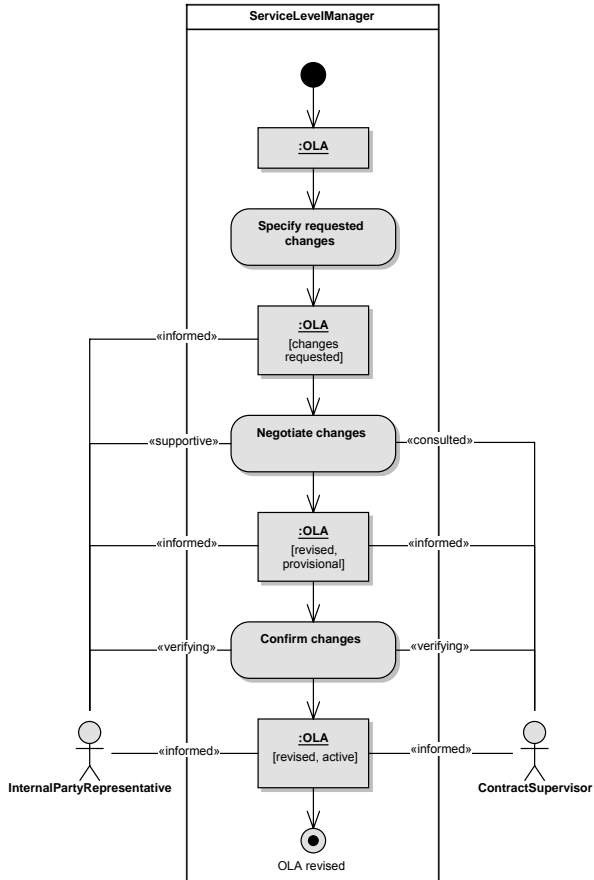
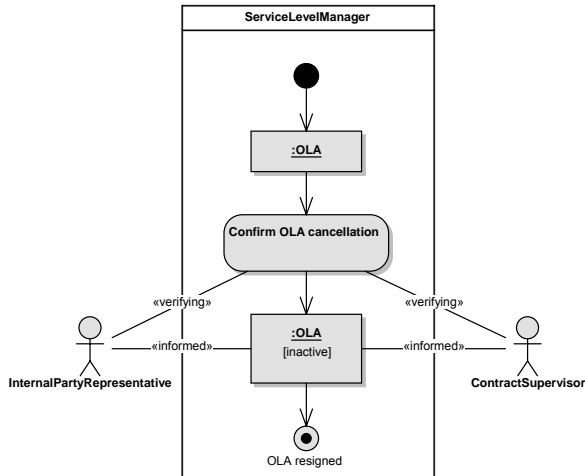


Abbildung 5.44: Referenz-Workflow 23 (*Revise OLA*)

Abbildung 5.45: Referenz-Workflow 24 (*Resign OLA*)

5.3.4.3 UC-Verwaltung

Der Aufgabenbereich der *UC-Verwaltung* steht in enger Analogie sowohl zur *OLA-Verwaltung* als auch zur *SLA-Verwaltung*, da ein Vertrag zwischen dem Provider und einem externen Lieferanten bzw. Subdienst-Provider letztlich einer Dienstvereinbarung entspricht, bei der der betrachtete Provider in der Rolle des Kunden auftritt. Die Primäranwendungsfälle 25 (*Close UC*), 26 (*Revise UC*) und 27 (*Resign UC*) fallen in diesen Funktionsteilbereich.

Anwendungsfälle 25 bis 27 – Primärszenarien Inhaltlich und strukturell sind die Referenz-Workflows zu diesen Anwendungsfällen, dargestellt in den Abbildungen 5.46, 5.47 und 5.48 nahezu identisch mit den Workflows aus dem Funktionsteilbereich *OLA-Verwaltung*. Die einzigen Unterschiede bestehen im Hinblick auf die eingebundene Dienstbringer-Rolle, die hier natürlich ein *ExternalSupplierRepresentative* ist, sowie auf das zentrale Informationsartefakt, welches eine Instanz der Artefaktklasse *UnderpinningContract* ist.

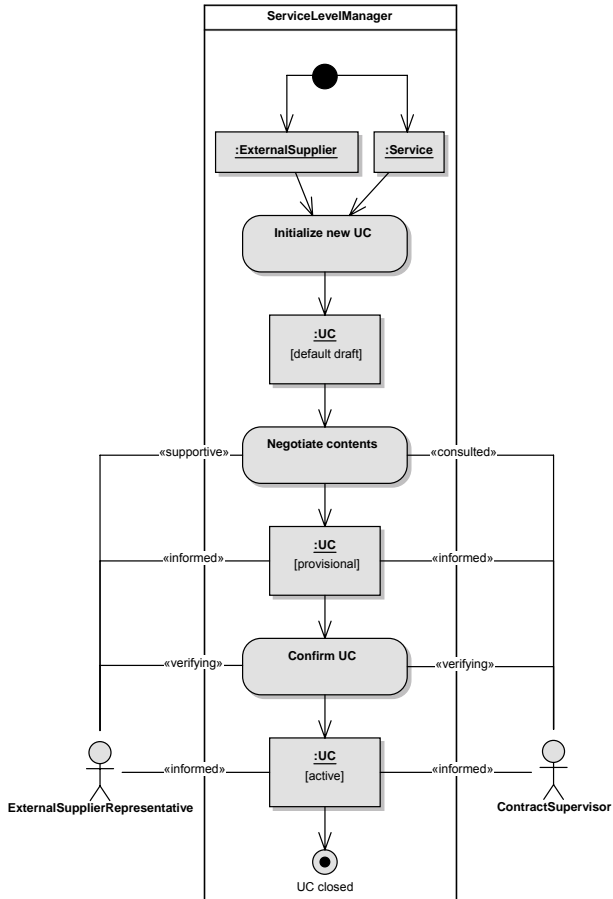


Abbildung 5.46: Referenz-Workflow 25 (*Close UC*)

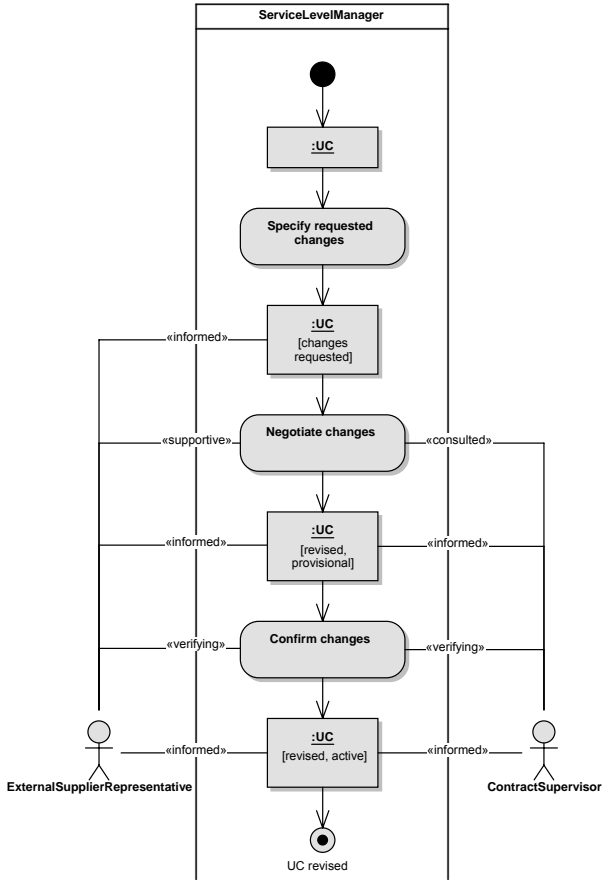


Abbildung 5.47: Referenz-Workflow 26 (*Revise UC*)

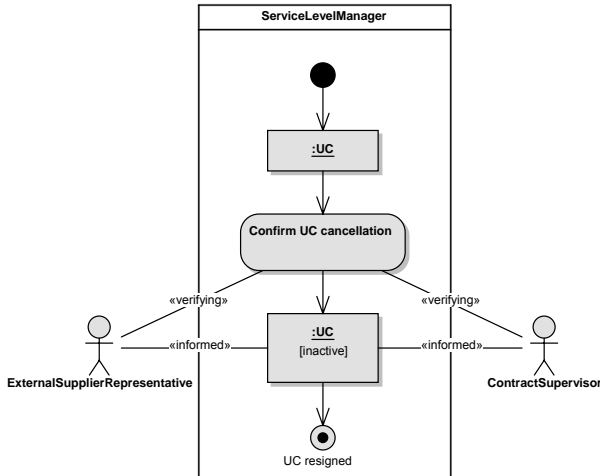


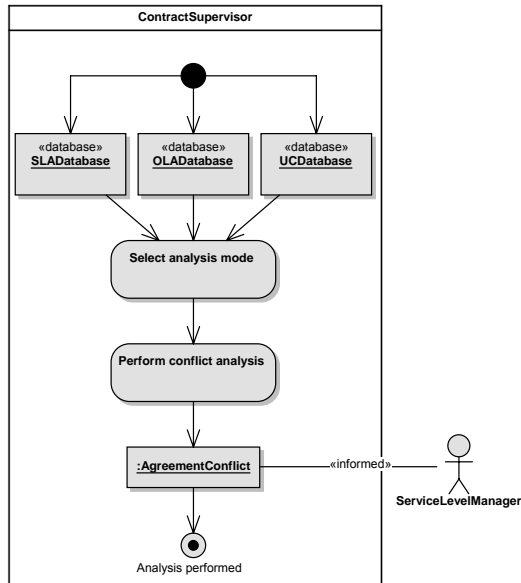
Abbildung 5.48: Referenz-Workflow 27 (*Resign UC*)

5.3.4.4 Konfliktanalyse

Der Funktionsteilbereich der *Konfliktanalyse* umfasst nur einen einzigen Primäranwendungsfall, nämlich den Anwendungsfall 28 (*Analyze agreements*). Übergeordnetes Ziel ist es, durch die Erkennung von konfliktären Zieldefinitionen in Dienstvereinbarungen (SLAs) und unterstützenden Vereinbarungen und Verträgen (OLAs, UCs), einen Beitrag zu einem widerspruchsfreien Service-Level-Management und zu einer konsistenten Dienstkette zu leisten.

Beispielhaft und ohne dadurch weitere Möglichkeiten auszuschließen, seien hier drei mögliche und sinnvolle Varianten (Modi) der Konfliktanalyse aufgeführt:

1. Gesamtanalyse: Alle identifizierbaren Abhängigkeiten zwischen SLAs, OLAs und UCs werden analysiert, und Konflikte werden dargestellt.
→ Sinnvoll im Kontext eines generellen SLM-Reviews, zum Beispiel im Rahmen interner Audits
2. Einzelanalyse: Konflikte zwischen einem ausgewählten SLA und allen unterstützenden OLAs und UCs werden ermittelt und dargestellt.
→ Sinnvoll nach Abschluss oder Anpassung eines SLAs

Abbildung 5.49: Referenz-Workflow 28 (*Analyze agreements*)

3. Selektive Analyse: Abhängigkeiten zwischen beliebigen ausgewählten SLAs, OLAs und UCs werden ermittelt und Konflikte dargestellt.
→ Bietet maximale Freiheit und Flexibilität in der Definition und Eingrenzung des Analyseraums, nicht angepasst auf spezifische Situationen oder Vorbedingungen

Anwendungsfall 28 – Primärszenario Der Referenz-Workflow zu diesem Anwendungsfall ist in Abbildung 5.49 dargestellt und umfasst folgende Aktivitäten:

- Auswahl des Analyse-Modus (*Select analysis mode*): Der Analyse-Modus (Gesamtanalyse, Einzelanalyse oder selektive Analyse) wird ausgewählt.
- Durchführen der Analyse (*Perform analysis*): Die eigentliche Abhängigkeits- und Konfliktanalyse wird durchgeführt. Obwohl sich erst

das Funktionsmodell der System-Sicht mit der Automatisierbarkeitsanalyse von Workflows und Aktivitäten auseinandersetzt, kann an dieser Stelle schon festgestellt werden, dass die Durchführung der Konfliktanalyse ohne geeigneten IT-Support schon bei vergleichsweise wenigen Vereinbarungen und Verträgen mit gegenseitigen Abhängigkeiten nicht mehr mit vertretbarem Zeitaufwand realisierbar ist.

Das Ergebnis eines Analyselaufs ist eine neue Instanz der Artefaktklasse **AgreementConflict**.

5.3.5 Funktionsbereich IV: Messdaten- und Berichtsmanagement

Es sei daran erinnert, dass die Funktionsbereiche III und IV als der *klassische Betrachtungsbereich* der Disziplin Service-Level-Management eingeordnet wurden. Während sich die Teilbereiche und Anwendungsfälle des Funktionsbereichs III mit dem Management der verschiedenen Arten von Verträgen und Vereinbarungen sowie dem Management von Konflikten zwischen diesen Vereinbarungen beschäftigen und sich damit vor allem der konsistenten Beschreibung eines an Kundenanforderungen orientierten *Soll-Zustands* der erbrachten Dienste verschreibt, beleuchtet der Funktionsbereich IV nun die andere Seite, die Ermittlung, Beschreibung und Kommunikation des tatsächlich erreichten *Ist-Zustands*. Er kann in die folgenden zwei Funktionsteilbereiche zerlegt werden:

- *Messung und Aufzeichnung*
- *Berichtswesen*

Diese beiden Funktionsteilbereiche, ihre enthaltenen Anwendungsfälle und die zugehörigen Referenz-Workflows werden im Folgenden beschrieben und beschließen das Funktionsmodell der Prozess-Sicht.

5.3.5.1 Messung und Aufzeichnung

Der Funktionsteilbereich *Messung und Aufzeichnung* besteht aus nur einem Primäranwendungsfall, dem gleichnamigen Anwendungsfall 29 (*Measure and record*), was jedoch nicht zu falschen Schlüssen bezüglich der Komplexität der Aufgaben in diesem Bereich verleiten sollte. Denn dass die Messung von Dienst(güte)parametern (Dienst-Monitoring) alles andere als eine triviale Angelegenheit ist, beweisen zahlreiche Studien und Forschungsarbeiten, die sich

mit diesem Thema auseinandersetzen. Dazu zählt unter anderem auch die Dissertation von Sailer [11], die sich mit der Konzeption einer Service-MIB beschäftigt.

Anwendungsfall 29 – Primärszenario Abbildung 5.50 illustriert den Referenz-Workflow zum Anwendungsfall 29, der von der Komplexität des Dienst-Monitorings abstrahiert und festlegt, welche grundlegenden Aktivitäten aus Sicht des Service-Level-Managements eine Rolle spielen.

Grundsätzlich muss zu jedem angebotenen (End-)Dienst spezifiziert werden, *was* und vor allem *wie* seine Leistung gemessen werden soll. Das Ergebnis dieser Festlegungen wird in einer Instanz des Datensatz-Artefakts **Service-MeasurementSpecification** festgehalten. Das Informationsmodell differenziert im Zusammenhang mit diesem Artefakt zwischen den Informationsklassen **MeasurementMetrics** (zu messende Leistungsparameter) und **MeasurementRules** (Messvorschriften). Die Verantwortung der Definition der Messspezifikation liegt beim Dienstportfolioverwalter. Er wird in dieser Aktivität vom Service-Level-Manager unterstützt, und um sicherzustellen, dass die Messspezifikation geeignet ist, den Zielerreichungsgrad der in bestehenden Dienstvereinbarungen festgelegten Zusicherungen zu bestimmen, kann der Kontraktverwalter zusätzlich beratend hinzugezogen werden.

Im Rahmen der Aktivität *Define measurement specification* werden also Festlegungen über die zu messenden Parameter, Messvorschriften und -verfahren sowie gegebenenfalls über die Installation und Konfiguration technischer Messeinrichtungen getroffen. Dies bildet den Ausgangspunkt dafür, dass die erforderlichen Messungen zum betreffenden Dienst während der Betriebsphase des Dienstes im Rahmen der Aktivität *Perform and record measurements* durchgeführt werden können. Die erfolgreich durchgeführten und laufenden Messungen liefern Messergebnisse, die in strukturierter Weise aufgezeichnet und damit dokumentiert werden müssen. Die so entstehenden Aufzeichnungen (**ServiceMeasurementResults**) bilden die Grundlage für das interne und externe Berichtswesen über die Dienstleistung und -qualität.

5.3.5.2 Berichtswesen

Die Effektivität und Qualität des Service-Level-Managements steht und fällt mit der Fähigkeit, aussagekräftige Berichte (*Reports*) zu generieren. Wie bereits in der Anforderungsanalyse erläutert sind primär zwei Arten von Berichten für das Service-Level-Management relevant:

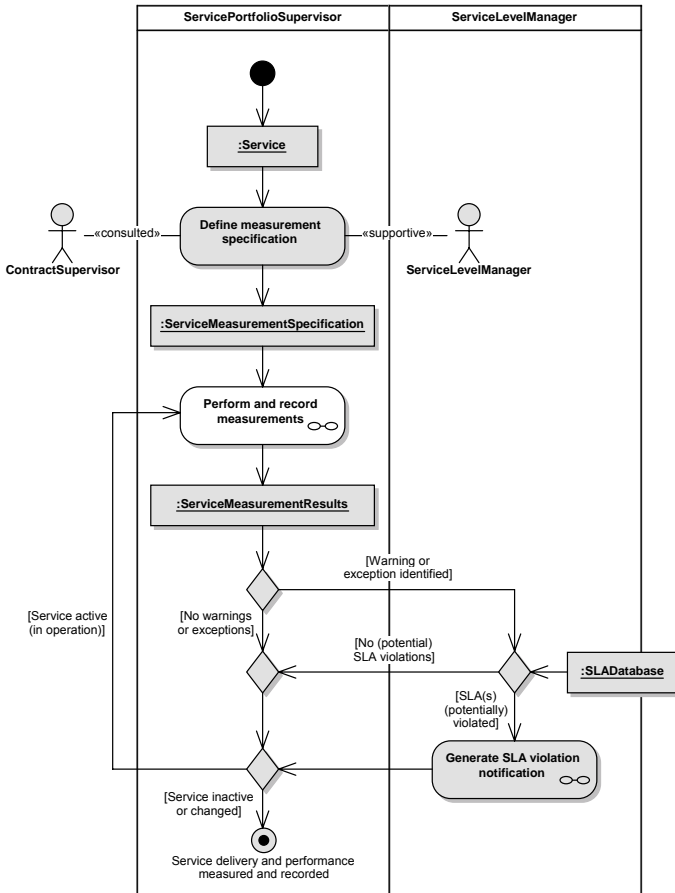


Abbildung 5.50: Referenz-Workflow 29 (*Measure and record*)

- IT-Provider-interne Berichte, die an die an der Dienstbringung beteiligten (internen) Parteien sowie an die Prozessverantwortlichen im Dienstmanagement gerichtet sind, und
- externe Berichte, die den IT-Kunden zugänglich gemacht werden.

Im Rahmen des Informationsmodells wurden bereits die Artefaktklassen `InternalReport` und `ExternalReport` für diese beiden Berichtsarten sowie `InternalReportSpecification` und `ExternalReportSpecification` zur Realisierung von Berichtsspezifikationen eingeführt und ihre Informationsanforderungen definiert. Sie stehen im Mittelpunkt dieses Funktionsteilbereichs, dem die Primäranwendungsfälle 30 (*Create internal report*), 31 (*Create external report*) und 32 (*Create SLA violation notification*) zugeordnet sind. Letzterer ergänzt das reguläre Berichtswesen und stützt sich auf ein weiteres bereits definiertes Informationsartefakt, die `SLAViolationNotification`.

Wichtig im Kontext der Anwendungsfälle 30 und 31 ist vorab vor allem die Feststellung, dass Service-Level-Berichte nichts anderes sind als in sinnvoller Weise aufbereitete und auf den Adressaten ausgerichtete Sichten auf Messaufzeichnungen und darauf aufbauende Aggregationen, Interpretationen und Beschreibungen von Folgen bzw. Konsequenzen. Ein grundlegendes Unterscheidungsmerkmal externer und interner Berichte ist neben der Auswahl der enthaltenen Informationen die Tatsache, dass sich externe Berichte stets auf eine Dienstvereinbarung mit einem Kunden beziehen, während sich interne Berichte direkt auf einen Dienst des Dienstportfolios beziehen.

Anwendungsfall 30 – Primärszenario Abbildung 5.51 zeigt zunächst den Referenz-Workflow zu Anwendungsfall 30, dessen Ziel die Erstellung eines internen Berichts über die Bereitstellung eines bestimmten (End-)Dienstes aus dem Dienstportfolio in einem zurückliegenden Zeitraum ist.

Grundlage für jeden internen Bericht ist eine entsprechende Berichtsspezifikation, also eine Instanz des Informationsartefakts `InternalReportSpecification`, die Vorgaben darüber macht, aus welchen Elementen und Berichtsinhalten jeder konkrete interne Bericht besteht. Die Erstellung einer solchen Spezifikation im Rahmen der Aktivität *Define internal report specification* findet unter Berücksichtigung des Dienstes statt, über den berichtet wird, der zu diesem Dienst gehörenden Messspezifikation sowie der operativen Vereinbarungen, Lieferantenverträge und Dienstvereinbarungen, die sich auf diesen Dienst oder Teile dieses Dienstes (auch Subdienste, Komponenten, realisierende informationstechnische Systeme) beziehen.

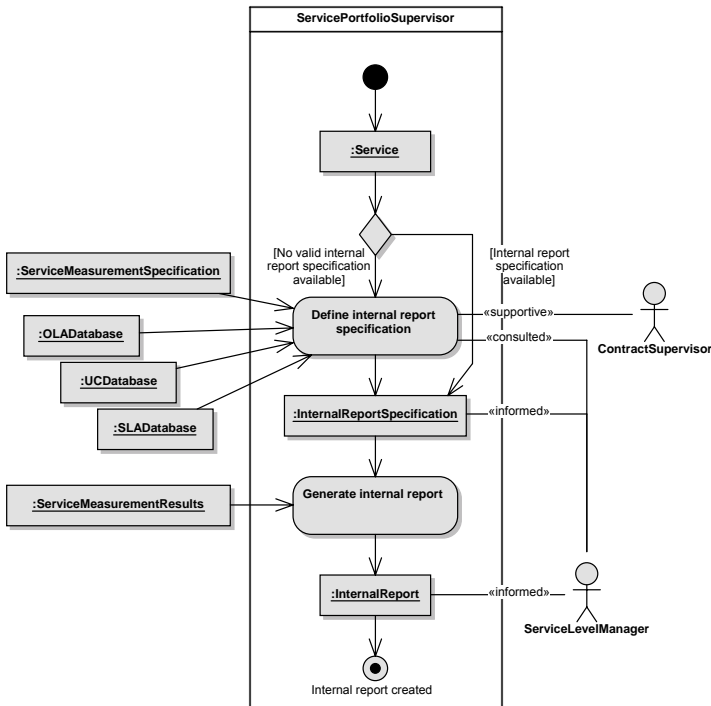


Abbildung 5.51: Referenz-Workflow 30 (*Create internal report*)

Die Spezifikation des internen Berichts (**InternalReportSpecification**) sowie die konkreten im Berichtszeitraum erhobenen Messdaten zum betreffenden Dienst (**ServiceMeasurementResults**) fließen in die zweite Hauptaktivität dieses Anwendungsfalls, die eigentliche Generierung des Berichts (*Generate internal report*) ein. Diese umfasst sowohl die spezifikationsgemäße Aufbereitung und Aggregation von Messdaten als auch – falls entsprechend spezifiziert – ihre Interpretation und die Ergänzung des Berichts um Elemente, die nicht direkt aus den Messdatensätzen gewonnen oder abgeleitet werden können.

Anwendungsfall 31 – Primärszenario In großen Teilen analog zu Anwendungsfall 30 gestaltet sich der Referenz-Workflow zu Anwendungsfall 31 (*Create external report*), dargestellt in Abbildung 5.52. Strukturelle und inhaltliche

Abweichungen ergeben sich in den folgenden Punkten:

- In die Aktivität zur Festlegung der Spezifikation des externen Berichts (*Define external report specification*) fließt die Dienstvereinbarung ein, auf die sich der externe Bericht beziehen wird (SLA), die Messspezifikation des assoziierten Dienstes (*ServiceMeasurementSpecification*) sowie die Spezifikation des internen Berichts zu diesem Dienst (*InternalReportSpecification*). Auf Basis dieser Eingaben wird eine Instanz der Artefaktklasse *ExternalReportSpecification* gemäß den Anforderungen aus dem Informationsmodell erstellt.
- In die Aktivität der Berichtserstellung (*Generate external report*) fließen neben der zugehörigen Berichtsspezifikation auch wieder die für den Berichtszeitraum relevanten Messdatensätze (*ServiceMeasurementResults*) und zusätzlich der interne Bericht (*InternalReport*) zum betreffenden Dienst mit ein.
- Da die Gesamtverantwortung für die Kundenkommunikation beim Service-Level-Manager liegt und die Korrektheit der erstellten externen Berichte in vielerlei Hinsicht einen kritischen Faktor darstellt, wird der erstellte externe Bericht erst im Rahmen einer dritten Aktivität *Confirm and deliver external report* durch den Service-Level-Manager verifiziert und damit die Autorisierung zur Übermittlung an den jeweiligen Kundenvertreter erteilt.

Bemerkung: Die Anwendungsfälle 31 und 32 liefern ein hohes Automatisierungspotenzial, welches im Rahmen des Funktionsmodells der System-Sicht vertiefend diskutiert wird.

Anwendungsfall 32 – Primärszenario Um den Anforderungen an ein kunden- und qualitätsorientiertes Dienstmanagement gerecht zu werden, müssen reguläre externe Berichte in bestimmten Fällen durch zusätzliche Benachrichtigungen ergänzt werden. Anwendungsfall 32, dessen Referenz-Workflow in Abbildung 5.53 dargestellt ist, komplettiert den Funktionsteilbereich *Berichtswesen*, indem er diesen Fall und diese Anforderung ans Service-Level-Management abdeckt.

Wie bereits im Rahmen der funktionalen Anforderungsanalyse deutlich gemacht wurde, entsteht eine Notwendigkeit der außerberichtsmäßigen Benachrichtigung eines IT-Kunden, wenn ein in einer Dienstvereinbarung mit diesem

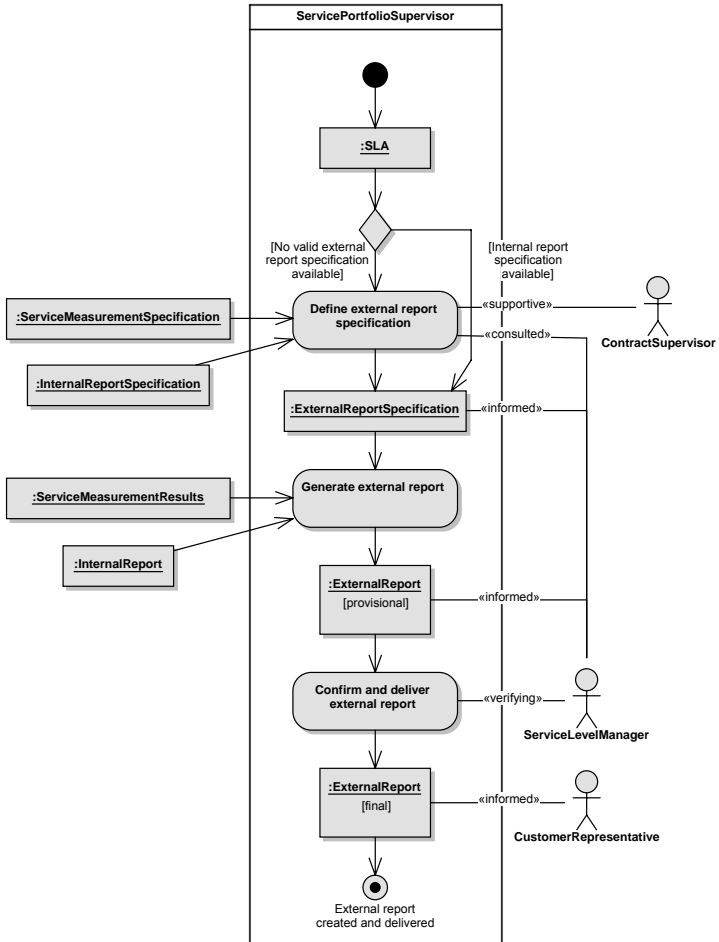


Abbildung 5.52: Referenz-Workflow 31 (*Create external report*)

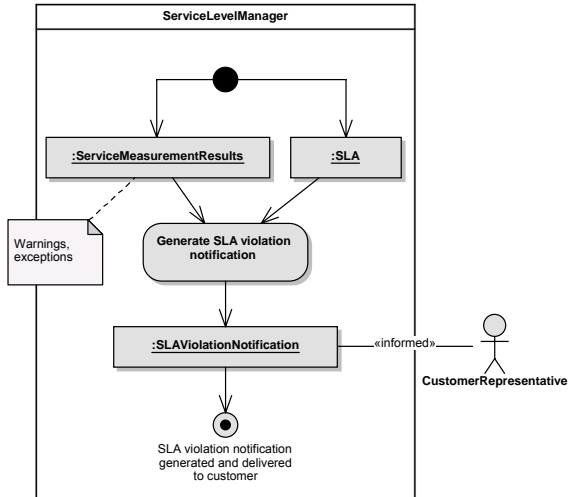


Abbildung 5.53: Referenz-Workflow 32 (*Create SLA violation notification*)

Kunden festgelegtes Ziel der Dienstbringung (*Service Delivery Target*) oder Service-Level-Ziel nicht eingehalten wurde oder wenn abzusehen ist, dass es nicht eingehalten werden kann. Dieses Vorgehen deckt sich auch mit dem *Code of Practice* (Umsetzungsempfehlungen) für Service-Level-Management des Standards ISO/IEC 20000 [31].

Die funktionale Prozess-Sicht auf diesen Anwendungsfall – gegeben durch seinen Workflow – ist nicht sehr komplex und umfasst die Erzeugung einer Instanz der Artefaktklasse *SLAViolationNotification* auf Basis einer Dienstvereinbarung (SLA) und von Messergebnissen (*ServiceMeasurementResults*). Von entscheidender Wichtigkeit ist allerdings, im Vorfeld die Ereignisse und Situationen zu ermitteln, die zur Auslösung des Anwendungsfalls führen, also zu erkennen, wann eine Dienstvereinbarung verletzt zu werden droht bzw. wann dies bereits geschehen ist. Dies liegt allerdings außerhalb des operativen Betrachtungsbereichs von Service-Level-Management und gehört zu den Anforderungen, die an ein effektives Dienst-Monitoring und Ereignismanagement (*Event Management*) gestellt werden.

Prinzipiell werden im *Event Management* drei Typen von Ereignissen unterschieden: Informationen (*informational events*), Warnungen (*warnings*) und Ausnahmen (*exceptions*). Speziell Warnungen und Ausnahmen stellen aus

Sicht des Service-Level-Managements eine wichtige Quelle für die Feststellung drohender oder tatsächlicher SLA-Verletzungen dar. Dabei steht und fällt die Klassifikation dieser Ereignistypen mit der auf Dienstvereinbarungen ausgerichteten Definition von sinnvollen Kriterien und Schwellwerten. Grundsätzlich gilt aber:

- Warnungen sollten generiert und an das Service-Level-Management kommuniziert werden, wenn bestimmte Schwellenwerte relevanter Dienstmetriken über- bzw. unterschritten wurden, die eine – durch geeignete Gegenmaßnahmen noch abzuwehrende – Gefahr für die Einhaltung einer oder mehrerer Dienstvereinbarungen darstellen.
- Ausnahmen sind zu generieren und an das Service-Level-Management zu kommunizieren, wenn Schwellenwerte von Dienstmetriken über- bzw. unterschritten wurden und dadurch ein Status erreicht wurde, der zwangsläufig zu einer nicht mehr abzuwehrenden Einhaltung einer oder mehrerer Dienstvereinbarungen geführt hat oder führen wird.

5.3.6 Zusammenfassung

Im Rahmen des Funktionsmodells der Prozess-Sicht wurde der Gesamtaufgabenkomplex Service-Level-Management auf Basis der Partitionierung des Funktionsspektrums aus der Anforderungsanalyse in die vier Funktionsbereiche *Dienstinformationsmanagement*, *Stakeholder-Management*, *Vereinbarungsmanagement* und *Messdaten- und Berichtsmanagement* zerlegt, denen die Primäranwendungsfälle 1 bis 32 zugeordnet werden können. Die vorgestellten Workflows helfen, ein besseres und detailliertes Verständnis über die durch die Primäranwendungsfälle gegebenen funktionalen Anforderungen aufzubauen. Doch vor allem schlagen sie die Brücke zwischen der Anforderungsanalyse und der System-Sicht der Architektur. Durch den Entwurf von Referenz-Workflows wurde zudem eine feiner-granulare und strukturiertere Sicht auf jeden Anwendungsfall sowie auf die Zusammenhänge und Abhängigkeiten zwischen verschiedenen Anwendungsfällen erzeugt. Es wurde gezeigt, wie sich die Rollen und Verantwortungsmodelle des Organisationsmodells und die Informationsartefakte des Informationsmodells in den Kontext der Primäranwendungsfälle und der mit ihnen verbundenen Aktivitäten einordnen.

Die hier nicht adressierten Primäranwendungsfälle 33 bis 38 spielen im nachfolgenden Unterkapitel, also im Kontext des Kommunikationsmodells der Prozess-Sicht, eine wichtige Rolle. Auf ihrer Basis werden dort vor allem die

durch ein Managementsystem zu unterstützenden Kommunikationsbeziehungen zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen analysiert (Interprozess-Kommunikation). Gleichzeitig fließen die Ergebnisse des Funktionsmodells in das nächste Unterkapitel ein und helfen, die Intraprozesskommunikation zu formalisieren.

5.4 Kommunikationsmodell

Das Kommunikationsmodell ist das letzte der vier Teilmodelle, die die Prozess-Sicht der Managementarchitektur bilden. Sein Ziel besteht darin, alle Kommunikationsbeziehungen im Zusammenhang mit der Disziplin Service-Level-Management strukturiert darzustellen.

Fragestellungen dieses Moduls: Die Fragen, die durch das Kommunikationsmodell der Prozess-Sicht vorrangig beantwortet werden müssen, lauten:

- Welche (grundsätzlichen) Kommunikationsbeziehungen ergeben sich auf Basis der Workflows aus dem Funktionsmodell zwischen den Rollen aus dem Organisationsmodell?
- Welche Kommunikations-/Interaktionsbeziehungen bestehen zwischen Service-Level-Management und anderen Disziplinen des dienstorientierten IT-Managements?

Die erste Frage ergibt sich aus der Tatsache, dass das Funktionsmodell der Prozess-Sicht primär die Kontroll- und Objektflüsse darstellt, die innerhalb der Anwendungsfälle zum Tragen kommen, dabei jedoch nicht beschreibt, welche Art der Kommunikation zwischen den an einem Anwendungsfall beteiligten Akteuren zur Durchführung oder zum Abschluss einer bestimmten Aktivität erforderlich ist. Anders gesagt wird zwar der grundsätzliche (mögliche) Ablauf eines Anwendungsfalls beschrieben, die dazu notwendige Interaktion zwischen Rollen jedoch nur oberflächlich spezifiziert.

Die Frage nach den Kommunikationsbeziehungen zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen ist von maßgeblicher Wichtigkeit für die Architektur und für den Entwurf eines konkreten IT-gestützten Managementsystems mit zugehöriger Plattform, weil es grundsätzlich nicht sinnvoll wäre, einzelne Dienstmanagementdisziplinen in Isolation zu betrachten. Vielmehr muss das komplexe Beziehungsgefüge eines Gesamt-Managementsystems für IT-Dienste berücksichtigt werden.

Aus diesen beiden Fragestellungen ergeben sich die nun folgenden Bestandteile des Kommunikationsmodells der Prozess-Sicht: das *Intraprozess-Kommunikationsmodell* zur Beantwortung der ersten Frage sowie das *Interprozess-Kommunikationsmodell* zur Beantwortung der zweiten Frage.

5.4.1 Intraprozess-Kommunikationsmodell

Das Fundament für das Intraprozess-Kommunikationsmodell wird durch die Kontroll- und Objektflüsse in den Referenz-Workflows des Funktionsmodells gebildet. Insofern baut dieses Teilkapitel stark auf dem vorherigen auf, setzt das streng-deduktive Vorgehen der Arbeit konsequent fort und komplettiert die Prozess-Sicht der Managementarchitektur.

Ziel des Intraprozess-Kommunikationsmodells ist es, erforderliche Interaktionen zwischen Akteuren im Rahmen eines SLM-unterstützenden Managementsystems zu identifizieren und zu klassifizieren. Die *Identifikation* muss vor dem Hintergrund der Zielsetzung erfolgen, ein vollständiges Bild über die Intraprozesskommunikation zu zeichnen. Die *Klassifikation* soll helfen, jede einzelne Kommunikationsbeziehung aus dem Gesamtgefüge besser zu verstehen, um letztlich Möglichkeiten und Grenzen einer IT-Unterstützung ableiten zu können und dadurch eine Brücke zur System-Sicht der Architektur zu schlagen.

5.4.1.1 Identifikation von Kommunikationsbeziehungen

Mit dem gegebenen Funktionsmodell lässt sich die Identifikation von Kommunikationsbeziehungen in einfacher Weise durchführen. Jede der beiden folgenden Bedingungen impliziert und bedingt eine Kommunikation zwischen zwei Akteuren des Managementsystems:

- Ein Kontroll- und/oder Objektfluss reicht vom operativen Verantwortungsbereich (in den Workflow-Diagrammen dargestellt als UML-*Swimlane*) einer Rolle in den operativen Verantwortungsbereich einer anderen Rolle. Die Grenze zwischen zwei *Swimlanes* wird also überschritten.
→ In diesem Fall wechselt die Verantwortung innerhalb des betreffenden Anwendungsfalls, was immer ein Mindestmaß an Kommunikation und Informationsaustausch bedingt.
- Eine Rolle des Organisationsmodells, in den Referenz-Workflows dargestellt durch ein Akteuren-Symbol, steht zu einer Aktivität oder einem Informationsartefakt außerhalb seines eigenen operativen Verantwortungsbereiches (also außerhalb seiner eigenen *Swimlane*) über eines der Verantwortungsmodelle, *informed*, *supportive*, *consulted* oder *verifying* in Beziehung.

→ Diese Situation impliziert ebenfalls eine Kommunikationsbeziehung zwischen zwei Rollen, deren Ausprägung natürlich maßgeblich vom jeweiligen Verantwortungsmodell abhängt.

5.4.1.2 Klassifikation der Kommunikationsbeziehungen

Wurden mit Hilfe der genannten Bedingungen alle Kommunikationsbeziehungen aus dem Funktionsmodell extrahiert, müssen diese geeignet klassifiziert werden. Das zu diesem Zweck anzuwendende Klassifizierungsschema sollte so beschaffen sein, dass sich aus der Klassifizierung einer Kommunikationsbeziehung zwischen zwei Akteuren Aussagen über Charakteristik und Komplexität dieser Beziehung ableiten lassen, die für einen späteren Systementwurf einen Nutzen darstellen.

Differenzierungskriterien Kommunikation zwischen zwei Akteuren kann sehr verschiedene Ausprägungen annehmen. Zur Ermittlung und Differenzierung dieser verschiedenen möglichen Ausprägungen werden zunächst drei wichtige Kriterien beschrieben, die helfen können, verschiedene Kommunikationsbeziehungen gegeneinander abzugrenzen:

1. Kommunikationsrichtung: Differenzierung zwischen unidirektionaler Kommunikation, einfach-bidirektionaler Kommunikation (genau eine Nachricht pro Richtung, in der Regel eine Anfrage und eine Antwort) und mehrfach-bidirektionaler Kommunikation (mehr als eine Nachricht pro Richtung)
2. Strukturiertheit der Nachricht(en): Differenzierung zwischen strukturierten Nachrichten (durch ein Informationsmodell standardisierte Signale oder Artefakte) und unstrukturierten Nachrichten (kein zugrunde liegendes standardisiertes Informations-/Datenmodell)
3. Semantikgehalt der Nachricht(en): Differenzierung zwischen semantiken Nachrichten (Signale oder Daten definierter einfacher Typen) und semantikhreichen Nachrichten (Informationsartefakte wie Datensätze, Dokumentationen und Berichte)

Kommunikationsmuster Legt man diese Differenzierungskriterien zugrunde, so ergeben sich in der Theorie zwölf verschiedene Klassen zur Einordnung von Kommunikationsbeziehungen, von denen allerdings nur fünf für diese Arbeit

		<i>unidirektional</i>	<i>einfach-bidirektional</i>	<i>mehrfach-bidirektional</i>
<i>strukturiert</i>	<i>semantikarm</i>	K_1	–	–
	<i>semantikreich</i>	K_2	K_3	K_4
<i>unstrukturiert</i>	<i>semantikarm</i>	–	–	–
	<i>semantikreich</i>	–	–	K_5

Tabelle 5.14: Festlegung von Kommunikationsmustern mit Hilfe der Differenzierungskriterien

sinnvoll und relevant sind. Tabelle 5.14 zeigt, welche Ausprägungen der Differenzierungskriterien für welche dieser fünf Klassen jeweils spezifisch sind. Diese fünf Klassen stellen die Kommunikationsmuster dar, die bei der Formalisierung des Intraprozesskommunikationsmodells im Abschnitt 5.4.1.3 zur Anwendung kommen.

Bemerkung: Alle Kommunikationsmuster werden unter der Annahme betrachtet, dass jeweils genau ein Sender mit einem Empfänger kommuniziert. Dies stellt im Grundsatz keine Beschränkung der Allgemeinheit dar, da sich komplexere 1-zu-n-Kommunikationsbeziehungen auf 1-zu-1-Kommunikationsbeziehungen reduzieren/abbilden lassen. Zudem entspricht diese Annahme der Umsetzung in den Referenz-Workflows des Funktionsmodells.

Die fünf relevanten Kommunikationsmuster sind:

K_1 Signalisierung (unidirektional, strukturiert, semantikarm)

Typische Kommunikationsformen für dieses Muster: Senden einer Steueranweisung, Übergabe eines *Tokens* im Rahmen eines Kontrollflusses und zur Ablaufkoordination

Auftreten im Rahmen des Funktionsmodells: Wann immer die operative Verantwortung innerhalb eines Kontrollflusses zu einem Anwendungsfall

von einer Rolle auf eine andere wechselt, muss eine Signalisierung vom bisher verantwortlichen an den neu verantwortlichen Akteur stattfinden.

K₂ Informationsübermittlung (unidirektional, strukturiert, semantisch)

Typische Kommunikationsformen für dieses Muster: Übermitteln eines Informationsartefakts, Benachrichtigung eines Akteurs über das Ergebnis der Durchführung einer Aktivität

Auftreten im Rahmen des Funktionsmodells: Wenn ein Informationsartefakt im Rahmen eines Objektflusses die Grenze einer *Swimlane* überschreitet, also vom operativen Verantwortungsbereich einer Rolle in den einer anderen Rolle wechselt, so entspricht diese Kommunikationsbeziehung dem Muster *K₂*. Dies gilt ebenfalls für die Assoziation eines Akteurs über das Verantwortungsmodell *informiert (informed)*, repräsentiert durch den gleichnamigen Assoziations-Stereotypen, mit einem Informationsartefakt oder einer Aktivität außerhalb seines eigenen operativen Verantwortungsbereichs.

K₃ Anfrage-Antwort-Kommunikation (einfach-bidirektional, strukturiert, semantisch)

Typische Kommunikationsform für dieses Muster: Anfrage einer spezifischen Aktion oder Information mit entsprechender Reaktion bzw. Antwort durch einen anderen Akteur

Auftreten im Rahmen des Funktionsmodells: Diese Art der Kommunikationsbeziehung ist relevant für jede Assoziation eines Akteurs über das Verantwortungsmodell *verifizierend (verifying)* mit einer Aktivität außerhalb seines eigenen operativen Verantwortungsbereichs. Zudem kann sie für Assoziationen der Stereotypen *beratend hinzugezogen (consulted)* und *unterstützend (supportive)* relevant sein, wenn diese insbesondere den Anforderungen einfacher Bidirektionalität und Strukturiertheit entsprechen. Die Anfrage geht dabei stets vom Akteur in der operativ verantwortlichen Rolle aus.

K₄ Standardisierter Dialog (mehrfach-bidirektional, strukturiert, semantisch)

Typische Kommunikationsformen für dieses Muster: Kommunikation oder Verhandlung unter Zuhilfenahme eines standardisierten Abfragebaums

Auftreten im Rahmen des Funktionsmodells: Standardisierter Dialoge zwischen zwei Akteuren können ebenfalls Rahmen der Verantwor-

tungsmodelle *beratend hinzugezogen (consulted)* und *unterstützend (supportive)* zum Einsatz kommen. Wenn eine einfache Anfrage-Antwort-Kommunikation nach dem Muster K_3 nicht ausreicht, die Kommunikation zwischen den beiden Akteuren aber trotzdem einem festgelegten Ablauf (Pfad in einem zugehörigen Abfrage-/Verlaufsbaum) folgt, also nicht völlig „frei“ stattfindet (vgl. Kommunikationsmuster K_5), kann sie in diese Klasse eingeordnet werden.¹

K_5 Freier Dialog (mehrfach-bidirektional, unstrukturiert, semantischreich)

Typische Kommunikationsformen für dieses Muster: Kommunikation oder Verhandlung im Sinne eines „freien Gesprächs“ (oder Schriftwechsels) und insbesondere ohne die Unterstützung durch standardisierte Abfragebäume oder Verlaufsprotokolle

Auftreten im Rahmen des Funktionsmodells: Auch dieses Kommunikationsmuster ist für Assoziationen der Stereotypen *beratend hinzugezogen (consulted)* und *unterstützend (supportive)* relevant und kommt zum Einsatz, wenn weder K_3 noch K_4 realisierbar erscheinen.

Bemerkung: Ob eine Assoziation vom Typ *beratend hinzugezogen (consulted)* oder *unterstützend (supportive)* zwischen einem Akteur und einer Aktivität außerhalb seines eigenen operativen Verantwortungsbereichs eine Kommunikationsbeziehung nach dem Muster K_3 , K_4 oder K_5 impliziert, kann nur auf der Basis fundierter Kenntnis über die Aktivität und die Anforderungen an den Informationsaustausch entschieden werden. Die Differenzierungskriterien helfen hierbei, eine eindeutige Beurteilung ist aber nicht in allen Fällen möglich. Für die Klassifikation im nächsten Abschnitt stellt dies eine besondere Herausforderung dar.

Die Kommunikationsmuster K_1 bis K_5 variieren in ihrem Potenzial zur Automatisierung und in ihrer zeitlichen Ausdehnung. Zudem kann zwischen synchroner und asynchroner Kommunikation unterschieden werden. Grundsätzlich kann im Falle asynchroner Kommunikation mit oder ohne Empfangsbestätigung gearbeitet werden. Diese Details eher technischer Natur werden an

¹Kommunikationsbeziehungen des Typs *Standardisierter Dialog (K_4)* spielen in der Praxis des Dienstmanagements vor allem im Rahmen der Entstörung (*Incident Management, Service Desk*) eine große Rolle. Die in diesem Bereich eingesetzten Werkzeuglösungen (*Trouble Ticket-Systeme*) wie zum Beispiel das *Open Ticket Request System (OTRS)* unterstützen Kommunikation nach diesem Muster, indem sie das Anlegen von Abfragebäumen und deren Verwendung im Kontext eines Tickets unterstützen.

dieser Stelle jedoch nicht weiter betrachtet. Somit beziehen sich die Kommunikationsmuster ausschließlich auf die aus Prozess-Sicht relevante *logische* Aktorenkommunikation.

5.4.1.3 Überblick über die Kommunikationsbeziehungen

Tabelle 5.15 stellt das Ergebnis der Analyse der Intraprozess-Kommunikationsbeziehungen im Service-Level-Management auf Basis des Funktionsmodells der Architektur und unter Anwendung des gerade beschriebenen Identifikations- und Klassifizierungsansatzes dar. Die erste Spalte gibt den Anwendungsfall an, in den die Kommunikationsbeziehung eingebettet ist und aus dessen Referenz-Workflow sie extrahiert wurde. Die zweite Spalte spezifiziert die Kommunikationsbeziehung unter Angabe der beteiligten Aktoren, des Kommunikationsmusters und der Kommunikationsrichtung. Die dritte und letzte Spalte gibt den Kontext der Kommunikationsbeziehung an, also die Aktivität, im Rahmen derer oder in Folge welcher die Kommunikation stattfindet. Speziell bei Kommunikationsbeziehungen nach dem Muster K_2 (Informationsübermittlung) wird hier zusätzlich das Informationsartefakt angegeben, das übermittelt wird.

Zur Verkürzung der Darstellung werden für die Rollen aus dem Organisationsmodell folgende Abkürzungen verwendet:

- Service-Level-Manager: SLMgr
- Dienstportfolioverwalter: SPSup
- Kontraktverwalter: ConSup
- Vertreter einer internen Diensterbringerpartei: IPRep
- Kundenvertreter: CusRep
- Lieferantenvertreter: SupRep

Fallen zwei (oder mehr) Kommunikationsbeziehungen im selben Kontext zusammen, so wird die resultierende Kommunikationsbeziehung nach dem höheren Kommunikationsmuster klassifiziert. Dies ist sehr häufig im Hinblick auf die Kommunikationsmuster K_1 (Signalisierung) und K_2 (Informationsübermittlung) der Fall, da der Kontroll- und der primäre Objektfluss eines Workflows meistens zusammenfallen.

<i>Anwendungsfall</i>	<i>Kommunikation</i>	<i>Kontext</i>
1 (<i>Create new service</i>)	SLMgr $\overset{K_5}{\leftrightarrow}$ CusRep	<i>Identify key functional service requirements</i>
	SLMgr $\overset{K_2}{\rightarrow}$ CusRep	<i>Formal requirements engineering \rightarrow :Service [requirements designed]</i>
	SLMgr $\overset{K_2}{\rightarrow}$ ChangeMgmt	<i>Formal requirements engineering \rightarrow :Service [requirements designed]</i>
	ReleaseMgmt $\overset{K_2}{\rightarrow}$ SLMgr	<i>Release service \rightarrow :Service [released]</i>
	ReleaseMgmt $\overset{K_2}{\rightarrow}$ SPSup	<i>Release service \rightarrow :Service [released]</i>
	SPSup $\overset{K_3}{\rightarrow}$ SLMgr	<i>Insert service into portfolio</i>
	SPSup $\overset{K_2}{\rightarrow}$ SLMgr	<i>Insert service into portfolio \rightarrow :Service [active]</i>
	SPSup $\overset{K_2}{\rightarrow}$ CusRep	<i>Insert service into portfolio \rightarrow :Service [active]</i>
2 (<i>Change service</i>)	SLMgr $\overset{K_2}{\rightarrow}$ ChangeMgmt	<i>Define change request \rightarrow :Service [change requested]</i>
	ChangeMgmt $\overset{K_2}{\rightarrow}$ SLMgr	<i>Assess and design change \rightarrow :Service [change designed]</i>
	ReleaseMgmt $\overset{K_2}{\rightarrow}$ SLMgr	<i>Release change \rightarrow :Service [changed]</i>
	ReleaseMgmt $\overset{K_2}{\rightarrow}$ SPSup	<i>Release change \rightarrow :Service [changed]</i>
	SPSup $\overset{K_2}{\rightarrow}$ SLMgr	<i>Update service portfolio \rightarrow :Service [changed, active]</i>
	SPSup $\overset{K_2}{\rightarrow}$ CusRep	<i>Update service portfolio \rightarrow :Service [changed, active]</i>
3 (<i>Delete service</i>)	SPSup $\overset{K_2}{\rightarrow}$ ChangeMgmt	<i>Remove service from portfolio \rightarrow :Service [inactive]</i>

4 (<i>Create new customer</i>)	SLMgr $\xleftrightarrow{K_4}$ CusRep	<i>Create customer</i>
	SLMgr $\xrightarrow{K_2}$ CusRep	<i>Create customer</i> \rightarrow :Customer
5 (<i>Change customer</i>)	SLMgr $\xleftrightarrow{K_4}$ CusRep	<i>Update customer</i>
	SLMgr $\xrightarrow{K_2}$ CusRep	<i>Update customer</i> \rightarrow :Customer [updated]
6 (<i>Delete customer</i>)	SLMgr $\xrightarrow{K_2}$ SPSup	<i>ActivityInitial</i> \rightarrow :Customer
	SPSup $\xrightarrow{K_1}$ SLMgr	<i>Dissociate customer from service catalog</i>
8 (<i>Assign catalog to customer</i>)	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Confirm assignment</i> \rightarrow :ServiceCatalog [assigned]
	SPSup $\xrightarrow{K_2}$ CusRep	<i>Confirm assignment</i> \rightarrow :ServiceCatalog [assigned]
9 (<i>Dissociate catalog from customer</i>)	SPSup $\xrightarrow{K_2}$ SLMgr	<i>ActivityInitial</i> \rightarrow :SLA
	SLMgr $\xrightarrow{K_2}$ SPSup	<i>Resign SLA</i> \rightarrow :SLA [resigned]
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Confirm dissociation</i> \rightarrow :ServiceCatalog [dissociated]
	SPSup $\xrightarrow{K_2}$ CusRep	<i>Confirm dissociation</i> \rightarrow :ServiceCatalog [dissociated]
10 (<i>Delete service catalog</i>)	SPSup $\xleftrightarrow{K_3}$ SLMgr	<i>Delete service catalog</i>
11 (<i>Add service to catalog</i>)	SPSup $\xleftrightarrow{K_3}$ SLMgr	<i>Link service with catalog</i>
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Link service with catalog</i> \rightarrow :Service [added to catalog]
	SPSup $\xrightarrow{K_2}$ CusRep	<i>Link service with catalog</i> \rightarrow :Service [added to catalog]

12 (<i>Remove service from catalog</i>)	$\text{SPSup} \xrightarrow{K_2} \text{SLMgr}$	<i>ActivityInitial</i> → :SLA
	$\text{SLMgr} \xrightarrow{K_2} \text{SPSup}$	<i>Resign SLA</i> → :SLA [resigned]
	$\text{SPSup} \xleftrightarrow{K_3} \text{SLMgr}$	<i>Unlink service from catalog</i>
	$\text{SPSup} \xrightarrow{K_2} \text{SLMgr}$	<i>Unlink service from catalog</i> → :Service [unlinked from catalog]
	$\text{SPSup} \xrightarrow{K_2} \text{CusRep}$	<i>Unlink service from catalog</i> → :Service [unlinked from catalog]
13 (<i>Create support level profile</i>)	$\text{SLMgr} \xleftrightarrow{K_5} \text{CusRep}$	<i>Define service support requirements</i>
	$\text{SLMgr} \xrightarrow{K_2} \text{IncidentMgmt}$	<i>Define service support requirements</i> → :SupportLevelProfile [requirements defined]
	$\text{IncidentMgmt} \xrightarrow{K_2} \text{SLMgr}$	<i>Design support level profile</i> → :SupportLevelProfile [proposed]
	$\text{SLMgr} \xleftrightarrow{K_3} \text{CusRep}$	<i>Confirm support level profile</i>
	$\text{SLMgr} \xrightarrow{K_2} \text{IncidentMgmt}$	<i>Confirm support level profile</i> → :SupportLevelProfile [rejected]
	$\text{SLMgr} \xrightarrow{K_2} \text{IncidentMgmt}$	<i>Confirm support level profile</i> → :SupportLevelProfile [active]
	$\text{SLMgr} \xrightarrow{K_2} \text{CusRep}$	<i>Confirm support level profile</i> → :SupportLevelProfile [active]
14 (<i>Change support level profile</i>)	$\text{SLMgr} \xleftrightarrow{K_4} \text{CusRep}$	<i>Define change request</i>

	SLMgr $\xrightarrow{K_2}$ IncidentMgmt	Define change request \rightarrow :SupportLevelProfile [change requested]
	IncidentMgmt $\xrightarrow{K_2}$ SLMgr	Redesign support level profile \rightarrow :SupportLevelProfile [changed, proposed]
	SLMgr $\xleftrightarrow{K_3}$ CusRep	Confirm support level profile
	SLMgr $\xrightarrow{K_2}$ IncidentMgmt	Confirm support level profile \rightarrow :SupportLevelProfile [changed, rejected]
	SLMgr $\xrightarrow{K_2}$ IncidentMgmt	Confirm support level profile \rightarrow :SupportLevelProfile [changed, active]
	SLMgr $\xrightarrow{K_2}$ CusRep	Confirm support level profile \rightarrow :SupportLevelProfile [changed, active]
15 (Delete support level profile)	SLMgr $\xrightarrow{K_2}$ IncidentMgmt	Delete support level profile \rightarrow :SupportLevelProfile [inactive]
16 (Create delivery party)	SLMgr $\xleftrightarrow{K_4}$ IPRep	Create internal service delivery party
	SLMgr $\xrightarrow{K_2}$ IPRep	Create internal service delivery party \rightarrow :InternalDeliveryParty
	SLMgr $\xleftrightarrow{K_4}$ SupRep	Create supplier
	SLMgr $\xrightarrow{K_2}$ SupRep	Create supplier \rightarrow :Supplier
17 (Change delivery party)	SLMgr $\xleftrightarrow{K_4}$ IPRep	Update internal service delivery party
	SLMgr $\xrightarrow{K_2}$ IPRep	Update internal service delivery party \rightarrow :InternalDeliveryParty [updated]
	SLMgr $\xleftrightarrow{K_4}$ SupRep	Update supplier
	SLMgr $\xrightarrow{K_2}$ SupRep	Update supplier \rightarrow :Supplier [updated]

19 (<i>Close SLA</i>)	CusRep $\xrightarrow{K_2}$ SLMgr	<i>Select service from catalog</i> → :Service
	SLMgr $\xleftrightarrow{K_5}$ CusRep	<i>Negotiate contents</i>
	SLMgr $\xleftrightarrow{K_5}$ ConSup	<i>Negotiate contents</i>
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Negotiate contents</i> → :SLA [provisional]
	ConSup $\xrightarrow{K_2}$ SLMgr	<i>Check SLA</i> → :SLA [checked]
	ConSup $\xrightarrow{K_2}$ CusRep	<i>Check SLA</i> → :SLA [checked]
	SLMgr $\xleftrightarrow{K_3}$ CusRep	<i>Confirm SLA</i>
	SLMgr $\xrightarrow{K_2}$ CusRep	<i>Confirm SLA</i> → :SLA [active]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm SLA</i> → :SLA [active]
20 (<i>Revise SLA</i>)	CusRep $\xrightarrow{K_2}$ SLMgr	<i>Specify requested changes</i> → :SLA [changes requested]
	SLMgr $\xleftrightarrow{K_5}$ CusRep	<i>Negotiate changes</i>
	SLMgr $\xleftrightarrow{K_5}$ ConSup	<i>Negotiate changes</i>
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Negotiate changes</i> → :SLA [revised, provisional]
	ConSup $\xrightarrow{K_2}$ SLMgr	<i>Check SLA</i> → :SLA [revised, checked]
	ConSup $\xrightarrow{K_2}$ CusRep	<i>Check SLA</i> → :SLA [revised, checked]
	SLMgr $\xleftrightarrow{K_3}$ CusRep	<i>Confirm changes</i>
	SLMgr $\xrightarrow{K_2}$ CusRep	<i>Confirm changes</i> → :SLA [revised, active]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm changes</i> → :SLA [revised, active]
21 (<i>Resign SLA</i>)	SLMgr $\xleftrightarrow{K_3}$ CusRep	<i>Confirm SLA cancellation</i>
	SLMgr $\xrightarrow{K_2}$ CusRep	<i>Confirm SLA cancellation</i> → :SLA [inactive]

	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm SLA cancellation</i> → :SLA [inactive]
	SLMgr $\xrightarrow{K_2}$ ChangeMgmt	<i>Confirm SLA cancellation</i> → :SLA [inactive]
22 (<i>Close OLA</i>)	SLMgr $\overset{K_5}{\leftrightarrow}$ IPRep	<i>Negotiate contents</i>
	SLMgr $\overset{K_5}{\leftrightarrow}$ ConSup	<i>Negotiate contents</i>
	SLMgr $\xrightarrow{K_2}$ IPRep	<i>Negotiate contents</i> → :OLA [provisional]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Negotiate contents</i> → :OLA [provisional]
	SLMgr $\overset{K_3}{\leftrightarrow}$ IPRep	<i>Confirm OLA</i>
	SLMgr $\overset{K_3}{\leftrightarrow}$ ConSup	<i>Confirm OLA</i>
	SLMgr $\xrightarrow{K_2}$ IPRep	<i>Confirm OLA</i> → :OLA [active]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm OLA</i> → :OLA [active]
23 (<i>Revise OLA</i>)	SLMgr $\xrightarrow{K_2}$ IPRep	<i>Specify requested changes</i> → :OLA [changes requested]
	SLMgr $\overset{K_5}{\leftrightarrow}$ IPRep	<i>Negotiate changes</i>
	SLMgr $\overset{K_5}{\leftrightarrow}$ ConSup	<i>Negotiate changes</i>
	SLMgr $\xrightarrow{K_2}$ IPRep	<i>Negotiate changes</i> → :OLA [revised, provisional]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Negotiate changes</i> → :OLA [revised, provisional]
	SLMgr $\overset{K_3}{\leftrightarrow}$ IPRep	<i>Confirm changes</i>
	SLMgr $\overset{K_3}{\leftrightarrow}$ ConSup	<i>Confirm changes</i>
	SLMgr $\xrightarrow{K_2}$ IPRep	<i>Confirm changes</i> → :OLA [revised, active]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm changes</i> → :OLA [revised, active]
24 (<i>Resign OLA</i>)	SLMgr $\overset{K_3}{\leftrightarrow}$ IPRep	<i>Confirm OLA cancellation</i>
	SLMgr $\overset{K_3}{\leftrightarrow}$ ConSup	<i>Confirm OLA cancellation</i>

	SLMgr $\xrightarrow{K_2}$ IPRep	<i>Confirm OLA cancellation</i> → :OLA [inactive]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm OLA cancellation</i> → :OLA [inactive]
25 (<i>Close UC</i>)	SLMgr $\xleftrightarrow{K_5}$ ESRep	<i>Negotiate contents</i>
	SLMgr $\xleftrightarrow{K_5}$ ConSup	<i>Negotiate contents</i>
	SLMgr $\xrightarrow{K_2}$ ESRep	<i>Negotiate contents</i> → :UC [provisional]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Negotiate contents</i> → :UC [provisional]
	SLMgr $\xleftrightarrow{K_3}$ ESRep	<i>Confirm UC</i>
	SLMgr $\xleftrightarrow{K_3}$ ConSup	<i>Confirm UC</i>
	SLMgr $\xrightarrow{K_2}$ ESRep	<i>Confirm UC</i> → :UC [active]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm UC</i> → :UC [active]
26 (<i>Revise UC</i>)	SLMgr $\xrightarrow{K_2}$ ESRep	<i>Specify requested changes</i> → :UC [changes requested]
	SLMgr $\xleftrightarrow{K_5}$ ESRep	<i>Negotiate changes</i>
	SLMgr $\xleftrightarrow{K_5}$ ConSup	<i>Negotiate changes</i>
	SLMgr $\xrightarrow{K_2}$ ESRep	<i>Negotiate changes</i> → :UC [revised, provisional]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Negotiate changes</i> → :UC [revised, provisional]
	SLMgr $\xleftrightarrow{K_3}$ ESRep	<i>Confirm changes</i>
	SLMgr $\xleftrightarrow{K_3}$ ConSup	<i>Confirm changes</i>
	SLMgr $\xrightarrow{K_2}$ ESRep	<i>Confirm changes</i> → :UC [revised, active]
	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm changes</i> → :UC [revised, active]
27 (<i>Resign UC</i>)	SLMgr $\xleftrightarrow{K_3}$ ESRep	<i>Confirm UC cancellation</i>
	SLMgr $\xleftrightarrow{K_3}$ ConSup	<i>Confirm UC cancellation</i>
	SLMgr $\xrightarrow{K_2}$ ESRep	<i>Confirm UC cancellation</i> → :UC [inactive]

	SLMgr $\xrightarrow{K_2}$ ConSup	<i>Confirm UC cancellation</i> → :UC [inactive]
28 (<i>Analyze agreements</i>)	ConSup $\xrightarrow{K_2}$ SLMgr	<i>Perform conflict analysis</i> → :AgreementConflict
29 (<i>Measure and record</i>)	SPSup $\xleftrightarrow{K_4}$ ConSup	<i>Define measurement specification</i>
	SPSup $\xleftrightarrow{K_4}$ SLMgr	<i>Define measurement specification</i>
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Perform and record measurements</i> → :ServiceMeasurementResults [warnings, exceptions]
30 (<i>Create internal report</i>)	SPSup $\xleftrightarrow{K_5}$ ConSup	<i>Define internal report specification</i>
	SPSup $\xleftrightarrow{K_5}$ SLMgr	<i>Define internal report specification</i>
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Define internal report specification</i> → :InternalReportSpecification
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Generate internal report</i> → :InternalReport
31 (<i>Create external report</i>)	SPSup $\xleftrightarrow{K_5}$ ConSup	<i>Define external report specification</i>
	SPSup $\xleftrightarrow{K_5}$ SLMgr	<i>Define external report specification</i>
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Define external report specification</i> → :ExternalReportSpecification
	SPSup $\xrightarrow{K_2}$ SLMgr	<i>Generate external report</i> → :ExternalReport
	SPSup $\xleftrightarrow{K_3}$ SLMgr	<i>Confirm and deliver external report</i>

	SPSup $\xrightarrow{K_2}$ CusRep	<i>Confirm and deliver external report</i> → :ExternalReport
32 (<i>Create SLA violation notification</i>)	SLMgr $\xrightarrow{K_2}$ CusRep	<i>Generate SLA violation notification</i> → :SLAViolationNotification

Tabelle 5.15: Übersicht über alle Intraprozess-Kommunikationsbeziehungen

5.4.1.4 Auswertung der Kommunikationsbeziehungen

Tabelle 5.15 liefert eine gute und belastbare Basis für systementwurfsrelevante Überlegungen im Rahmen der System-Sicht der Architektur, wenn es unter anderem darum geht, die Unterstützung der Intraprozesskommunikation durch geeignete Plattform-Komponenten im Rahmen eines IT-gestützten Managementsystems für Service-Level-Management zu fördern.

Mit dieser Zielsetzung vor Augen sollen die wichtigsten Ergebnisse und Schlussfolgerungen aus dem Intraprozesskommunikationsmodell (der Prozess-Sicht) im Folgenden kurz aufgeführt werden:

- Das mit Abstand am häufigsten instanziierte Kommunikationsmuster ist K_2 (Informationsübermittlung). Dies überrascht zwar nicht, macht aber deutlich, dass dem Thema *Austausch von Informationsartefakten* bzw. *Zugriff auf eine gemeinsame Informationsbasis* im Rahmen eines Systementwurfs eine hohe Bedeutung zukommen muss.
- Kommunikationsbeziehungen nach den Mustern K_3 (Anfrage-Antwort-Kommunikation) und K_4 (standardisierter Dialog) machen mehr als ein Fünftel aller identifizierten Kommunikationsbeziehungen aus. Sie sind damit für ein IT-gestütztes Managementsystem von großer Relevanz, und so ist auf der Ebene der System-Sicht der Architektur zu untersuchen, wie ein adäquater IT-Support für diese Kommunikationsmuster aussehen kann.
- Obwohl Service-Level-Management im direkten Vergleich mit vielen anderen Dienstmanagementprozessen „von sich aus“ deutlich weniger strukturiert ist und nach oberflächlicher Analyse scheinbar nur geringes Automatisierungspotenzial bietet, zeigt sich spätestens mit diesem Teilkapitel, dass eine Architektur wie die in dieser Arbeit vorgestellte insbesondere durch ein differenziertes Informations- und Funktionsmodell einen

entscheidenden Beitrag zur Formalisierung als Voraussetzung für die IT-Unterstützung der Intraprozesskommunikation leisten kann.

5.4.2 Interprozess-Kommunikationsmodell

Die zweite Komponente des Kommunikationsmodells der Prozess-Sicht ist das *Interprozess-Kommunikationsmodell*. Es beschäftigt sich mit Kommunikationsvorgängen zwischen der Disziplin (bzw. dem Managementprozess) Service-Level-Management und anderen Dienstmanagementdisziplinen/-prozessen. Daraus ergeben sich Anforderungen an die Realisierung der Schnittstellen zwischen Managementprozessen und somit auch der Schnittstellen zwischen Managementsystemen zur Förderung eines integriert-kooperativen Ansatzes im Dienstmanagement.

Um die Fragestellungen nach Interprozess-Kommunikationsbeziehungen und resultierenden Schnittstellenanforderungen adressieren zu können, muss aber zunächst eine Annahme darüber getroffen werden, welche anderen Dienstmanagementdisziplinen überhaupt existieren. Diese sollte allerdings nicht die Allgemeingültigkeit der Architektur gefährden. Hilfreich bei der Festlegung sind die in Kapitel 4 vorgestellten ITSM-Prozessrahmenwerke sowie der umfassende Standard ISO/IEC 20000. Letzterer liefert mit seinem Prozess-Rahmenwerk eine Basis, die für die Zwecke dieser Arbeit und insbesondere das Interprozess-Kommunikationsmodell einen sehr guten Ausgangspunkt bildet.

5.4.2.1 Ein Referenz-Prozessrahmenwerk

Die folgenden Prozesse in vier Kategorien (Prozessgruppen) werden im Rahmen des Kommunikationsmodells der Architektur berücksichtigt. Sie entsprechen im Kern den in ISO/IEC 20000 spezifizierten Prozessen und Prozessgruppen – mit drei geringfügigen Unterschieden:

- Der Prozess *Business Relationship Management* aus ISO/IEC 20000 wird im Kontext dieses Prozessrahmenwerks als *Customer Relationship Management* eingeführt, um eine konsistente Terminologie im Sinne der Begriffsbildung fortzusetzen.
- *Release Management* wird im Kontext von *Change Management* und *Configuration Management* und somit als Teil der Prozessgruppe *Control Processes* betrachtet.

- Die Prozesse *Incident Management* und *Problem Management* werden im Prozessrahmenwerk von ISO/IEC 20000 zur Gruppe der *Resolution Processes* zusammengefasst. Um die Funktion des Prozesses *Incident Management*, eine Anwender-Schnittstelle nicht nur für Störungsmeldungen, sondern auch für Anfragen (*Service Requests*, vgl. Kapitel 5.2.2) bereitzustellen, wird die entsprechende Prozessgruppe hier allgemeiner als *Service Support Processes* referenziert. Diese Nomenklatur findet auch im Rahmen der *IT Infrastructure Library* in der Version 2 Anwendung.

Eine kurze einführende Erklärung und Zieldefinition zu jedem Prozess wird in den Abschnitten 5.4.2.3 bis 5.4.2.6 gegeben, die sich mit den Interprozess-Kommunikationsbeziehungen aus Sicht des Service-Level-Managements befassen.

5.4.2.2 Quellen für Anforderungen an die Interprozesskommunikation

Zur Identifizierung relevanter Interprozess-Kommunikationsbeziehungen im Rahmen der Managementarchitektur stehen drei Quellen zur Verfügung:

1. Die Primäranwendungsfälle 33 bis 38 aus der funktionalen Anforderungsanalyse (vgl. Kapitel 3.3)
2. Im Rahmen der Modellierung der Referenz-Workflows zu den Anwendungsfällen 1 bis 32 (Funktionsmodell der Prozess-Sicht) explizit definierte Kontroll- und Objektflüsse zwischen Service-Level-Management und *Change Management*, *Release Management* und *Incident Management*
3. Anforderungen an Prozess-Schnittstellen aus ISO/IEC 20000 und anderen Prozessrahmenwerken

Alle drei Quellen gemeinsam bilden eine Analysegrundlage und fließen in die Synthese des Interprozess-Kommunikationsmodells ein.

5.4.2.3 Schnittstellen zu Prozessen der Dienstplanung und -bereitstellung

Als erster großer Block werden die Zusammenhänge (Kommunikationsbeziehungen) zwischen Service-Level-Management und den Prozessen der Dienstplanung und -bereitstellung (*Service Delivery Processes*) untersucht. Gelegentlich werden diese auch als die „taktischen Prozesse“ bezeichnet, um sie von

<i>Prozessgruppe</i>	<i>Prozesse</i>
Dienstplanung und -bereitstellung (<i>Service Delivery Processes</i>)	Service-Level-Management
	Kapazitätsmanagement (<i>Capacity Management</i>)
	Verfügbarkeits- und Kontinuitätsma- nagement (<i>Availability and Continuity Manage- ment</i>)
	Management der Informationssicher- heit (<i>Information Security Management</i>)
	Budgetierung und Abrechnung (<i>Budgeting and Accounting</i>)
	Berichtswesen (<i>Service Reporting</i>)
Beziehungsmanagement (<i>Relationship Processes</i>)	Management der Kundenbeziehungen (<i>Customer Relationship Management</i>)
	Management der Lieferantenbeziehun- gen (<i>Supplier Relationship Management</i>)
Dienstkontrolle und -freigabe (<i>Control and Release Processes</i>)	Änderungsmanagement (<i>Change Management</i>)
	Konfigurationsmanagement (<i>Configuration Management</i>)
	Soft- und Hardwareverteilung (<i>Release Management</i>)
Dienstunterstützung (<i>Service Support Processes</i>)	Störungsmanagement (<i>Incident Management</i>)
	Problemmangement (<i>Problem Management</i>)

Tabelle 5.16: Grundlegendes Referenz-Prozessrahmenwerk in Anlehnung an ISO/IEC 20000

den eher operativen Prozessen wie *Incident Management* abzugrenzen. Bezogen auf den generischen Dienstlebenszyklus (vgl. Kapitel 3.2.2) tangieren diese Prozesse vor allem die Phasen der Planung, Verhandlung und Bereitstellung von Diensten.

Capacity Management Die Bereitstellung ausreichender Kapazitäten zur Realisierung der angebotenen Dienste ist essenziell für eine qualitätsorientierte Dienstleistung. Aus ITIL heraus entspringt die Unterscheidung zwischen dem klassischen Kapazitätsbegriff auf Ressourcenebene und dem Begriff der *Dienstkapazität*. Ziel muss es sein, die erforderliche Dienstkapazität entsprechend der Kundenanforderungen korrekt zu ermitteln und im zweiten Schritt sinnvoll auf Anforderungen an die Ressourcenkapazität (zum Beispiel erforderliche Server- und Netzkapazitäten) abzubilden. Was unter der Kapazität eines Dienstes genau verstanden werden kann, lässt sich nicht allgemein festlegen und hängt stark von dessen Charakteristik ab. So ist die Kapazität eines Applikationsdienstes beispielsweise gekennzeichnet durch seinen Parallelitätsgrad (Wie viele Anwender können den Dienst gleichzeitig nutzen?) und die maximalen Antwortzeiten. Dagegen hängt die Kapazität eines Dienstes wie *Bereitstellung und Einrichtung eines Standard-EDV-Arbeitsplatzes* von der Reaktionszeit auf eine entsprechende Anfrage und die Zeit für die Dienstleistung in Abhängigkeit von der Zahl der Anfragen in einem bestimmten Zeitraum ab.

Ziel dieses Prozesses: Sicherstellen, dass der Provider stets über ausreichende Kapazitäten verfügt, um gegenwärtige und künftige vereinbarte Kundenanforderungen zu erfüllen

Aufgabe des Service-Level-Managements ist es, beim Abschluss einer Dienstvereinbarung mit einem Kunden die Anforderungen an die Dienstkapazität zu bestimmen und in Form geeigneter Service-Level-Ziele zu quantifizieren. Dies muss in enger Kopplung an den Prozess *Capacity Management* und in Übereinstimmung mit der aktuellen Kapazitätsplanung geschehen. Für Service-Level-Ziele, die eine Änderung dieser Planung nach sich ziehen würden, müssen zuvor entsprechende Zusicherungen vom *Capacity Management* eingefordert werden. Operative Vereinbarungen können verwendet werden, um diese zu manifestieren.

Die Informationsflüsse zwischen dem Service-Level-Management (SLM) und dem *Capacity Management* (CapM) sind im Ergebnis:

- SLM → CapM: Neue und geänderte Anforderungen hinsichtlich der Dienstkapazität auf Basis von allgemeinen (im Rahmen des Dienstent-

wurfs) und spezifischen (durch SLAs festgelegte) Kundenanforderungen an die jeweiligen IT-Dienste

- SLM ← CapM: Kapazitätsplanung und Zusicherungen von Dienstkapazitäten

Availability/Continuity Management Dieser Prozess beschäftigt sich mit den wichtigen Qualitätsaspekten der *Verfügbarkeit* und der *Kontinuität* von IT-Diensten. Als Verfügbarkeit gilt hierbei die Fähigkeit eines Dienstes, seine erforderliche Funktion zu einem bestimmten Zeitpunkt oder über ein bestimmtes Zeitintervall zu erfüllen, während mit dem Begriff der Kontinuität die Gewährleistung einer (gegebenenfalls eingeschränkten) Verfügbarkeit in und nach einem Katastrophenfall adressiert wird.

Ziel dieses Prozesses: Sicherstellen, dass dem Kunden gegenüber vereinbarte Verpflichtungen zur Dienstkontinuität und -verfügbarkeit unter allen Umständen eingehalten werden können

SLA-relevante Zusicherungen hinsichtlich der Dienstverfügbarkeit und Dienstkontinuität müssen in enger Abstimmung zwischen dem Service-Level-Management und dem Prozess *Availability and Continuity Managements* erfolgen. Die Verantwortlichkeiten für die Erhebung von Daten zur Bestimmung der (durchschnittlichen) Dienstverfügbarkeit müssen festgelegt werden.

Die Informationsflüsse zwischen dem Service-Level-Management (SLM) und dem *Availability/Continuity Management* (ACM) sind im Einzelnen:

- SLM → ACM: Allgemeine sowie kundenspezifische Mindestanforderungen an die Dienstverfügbarkeit und -kontinuität auf Basis geschäftlicher Anforderungen, Anfragen bezüglich der Zusicherung bestimmter Verfügbarkeitsniveaus
- SLM ← ACM: Verfügbarkeits- und Kontinuitätszusagen in dienst- und kundenspezifischer Ausprägung, Daten und Berichte zur tatsächlich erreichten Dienstverfügbarkeit und -kontinuität

Information Security Management Gegenstand des Managements der Informationssicherheit ist der Schutz von Informationen und informationsspeichernder, -übertragender und -verarbeitender Systeme.

Ziel dieses Prozesses: Effektives Management der Informationssicherheit über alle Aktivitäten des Dienstmanagements hinweg

Anforderungen an die Informationssicherheit können je nach Dienst und Kunde sehr unterschiedlich ausfallen. Bei der Bereitstellung eines Dienstes an einen Kunden muss nicht immer zwangsläufig die größtmögliche Informationssicherheit gewährleistet werden, sondern ein angemessenes Maß. Und dieses richtet sich in erster Linie danach, wie sensibel die im Rahmen der Dienstleistung anfallenden, verarbeiteten und ausgetauschten Informationen im Kontext der Geschäftsprozesse sind. Folgende Aspekte der Informationssicherheit sind in diesem Zusammenhang zu berücksichtigen: Informationsverfügbarkeit (Schutz der Information vor Verlust/Löschung), Informationsvertraulichkeit (Schutz der Information vor unberechtigtem Zugriff) und Informationsintegrität (Schutz der Information vor unautorisierten Änderungen).

Die Informationsflüsse zwischen dem Service-Level-Management und dem *Information Security Management* sind im Einzelnen:

- SLM → *Information Security Management*: Allgemeine sowie kunden- und dienstspezifische Mindestanforderungen an die Informationssicherheit, Input für den Dienst-Informationssicherheitsplan (vgl. Kapitel 5.2.2 – Grundkonzeptmodell, Konzeptklasse *ServiceSecurityPlan*) auf Basis von SLAs
- SLM ← *Information Security Management*: Sicherheitszusagen in dienst- und kundenspezifischer Ausprägung, Berichte über sicherheitsrelevante Vorfälle und (potenzielle) sicherheitsbezogene SLA-Verletzungen

Budgeting and Accounting Auch die finanzielle Planung der Erbringung von IT-Diensten (Budgetierung) sowie die Ermittlung der tatsächlich entstandenen Kosten (Kostenrechnung) innerhalb einer Abrechnungsperiode gehören zu den Aufgaben, die durch ein Dienstmanagementsystem abgedeckt werden.

Ziel dieses Prozesses: Planung und Ermittlung der Kosten für die Dienstleistung

Wie stark die Beziehung zwischen Service-Level-Management und dem Prozess *Budgeting and Accounting* ausgeprägt ist, hängt vor allem maßgeblich davon ab, ob eine Weiterverrechnung (*Charging*) der Kosten für die erbrachten IT-Dienste an die IT-Kunden stattfindet oder nicht.

Die Informationsflüsse zwischen Service-Level-Management und *Budgeting and Accounting* sind im Einzelnen:

- SLM → *Budgeting and Accounting*: Anfragen bezüglich kostenbeeinflussender Kundenanforderungen, Anforderungen an die Leistungsverrechnung von Diensten (Kunden, Abrechnungseinheiten, Verrechnungsschlüssel)
- SLM ← *Budgeting and Accounting*: Dienstbezogene Budgetplanung, dienstbezogene Kostenrechnung, Leistungsverrechnung für Dienstanspruchnahme durch einen Kunden (vgl. Kapitel 5.2.2 – Grundkonzeptmodell, Konzeptklasse *ServiceInvoice*)

5.4.2.4 Schnittstellen zu Beziehungsprozessen

Die Gruppe der *Relationship Processes* beinhaltet zwei Prozesse, die sich mit Aspekten der Etablierung und Pflege der Beziehungen zu Kunden und externen Lieferanten beschäftigen. Dabei wird implizit von einem Domänenmodell ausgegangen (obwohl dieses nicht explizit durch den Standard ISO/IEC 20000 definiert wird), welches genau den Domänenfestlegungen aus dem Organisationsmodell der Prozess-Sicht dieser Arbeit entspricht.

Customer Relationship Management Die Provider-Kunden-Beziehung ist ein essenzieller Faktor für ein erfolgreiches Service-Level-Management. Dienstvereinbarungen stellen den wichtigsten Beitrag zur Untermauerung dieser Beziehung dar. Die Abgrenzung zwischen Service-Level-Management und *Customer Relationship Management* (CRM) ist allerdings nicht trivial und hängt vom festgelegten Betrachtungsbereich (*Scope*) der beiden Prozesse ab. Zu den Aufgaben des *Customer Relationship Managements* gehören gemäß ISO/IEC 20000 die Identifikation und Dokumentation der IT-Kunden und die Durchführung sogenannter *Service Review Meetings*.

Ziel dieses Prozesses: Etablierung und Pflege einer guten Beziehung zwischen Provider und Kunde basierend auf dem Verständnis der Kundenanforderungen und kundenspezifischen Erfolgsfaktoren

Die Frage nach der Abgrenzung stellt sich aus Sicht des Service-Level-Managements und im Hinblick auf die Realisierung eines IT-gestützten Managementsystems für diese Disziplin vor allem im Hinblick auf die Anwendungsfälle 4 (*Create customer*), 5 (*Change customer data*) und 6 (*Delete customer*).

Je nach gewähltem Ansatz – von der strikten Ausgliederung dieser Aktivitäten aus dem Service-Level-Management an das CRM und der Trennung dieser Prozesse und ihrer unterstützenden Managementsysteme bis hin zur Einbettung dieser Kern-Aktivitäten und der resultierenden Informationsbasis in das Service-Level-Management – ergeben sich unterschiedliche und unterschiedlich stark ausgeprägte Kommunikationsbeziehungen.

Im Folgenden einige mögliche und sinnvolle Informationsflüsse zwischen Service-Level-Management und *Customer Relationship Management* unter der Annahme, dass ein Ansatz gewählt wird, der weder eine vollständige Integration CRM-bezogener Aufgabenbereiche in das Service-Level-Management, noch eine (zu) strikte Trennung verfolgt. Angenommen wird, dass die grundlegende Kundenverwaltung (Identifikation der Kunden und Ermittlung ihrer Basisdaten) sowie Maßnahmen zur Verbesserung der Provider-Kunden-Beziehung, die nicht in direktem Zusammenhang mit SLM-spezifischen Aktivitäten stehen, vom CRM übernommen werden. Die Anwendungsfälle 4 bis 6 bleiben als Teil eines SLM-unterstützenden Managementsystems erhalten, wobei im Hinblick auf die Informationsbasis *CustomerDatabase* eine enge Kopplung mit dem CRM sinnvoll ist.

- SLM → *Customer Relationship Management*: Kundenspezifische Informationen hinsichtlich Dienstanforderungen und abgeschlossener SLAs, SLM-spezifische Ergänzungen der Kunden-Basisdaten
- SLM ← *Customer Relationship Management*: Basisdaten über Kunden, Ergebnisse aus Kundengesprächen/Meetings

Supplier Relationship Management Die Provider-Lieferanten-Beziehung basiert auf Lieferantenverträgen (UCs) und spielt somit für die Realisierung einer nahtlosen Dienstkette aus Sicht des Service-Level-Managements ebenfalls eine wichtige Rolle.

Ziel dieses Prozesses: Management der Lieferantenbeziehungen zur Sicherstellung einer nahtlosen Dienstleistung

Hier sind es die Anwendungsfälle 16 (*Create delivery party*), 17 (*Change delivery party data*) und 18 (*Delete delivery party*), die die Abgrenzung erschweren – analog zum Abgrenzungsproblem im Zusammenhang mit dem *Customer Relationship Management*. Daher werden im Hinblick auf das *Supplier Relationship Management* (SRM) analoge Annahmen getroffen: Die grundlegende Lieferantenidentifikation und -verwaltung und alle Aktivitäten ohne direkten

Bezug zu Service-Level-Management – wie zum Beispiel die Durchführung von Lieferantenaudits – fallen ins SRM, für die Anwendungsfälle 16 bis 18 und verbundene Aktivitäten wird eine enge Kopplung gefordert.

Die Informationsflüsse zwischen Service-Level-Management und *Supplier Relationship Management* sind demnach:

- SLM → *Supplier Relationship Management*: Lieferantenspezifische Informationen hinsichtlich Dienstangeboten und abgeschlossener UCs, SLM-spezifische Ergänzungen der Lieferanten-Basisdaten
- SLM ← *Supplier Relationship Management*: Basisdaten über Lieferanten, Ergebnisse aus Lieferanten-Audits

5.4.2.5 Schnittstellen zu Kontroll- und Freigabeprozessen

Die dritte Prozessgruppe, deren Zusammenhänge mit dem Service-Level-Management im Rahmen des Kommunikationsmodells der Prozess-Sicht analysiert werden, ist die der Kontroll- und Freigabeprozesse (*Control and Release Processes*). Durch diese Prozesse primär tangierte Dienstlebenszyklusphasen sind die Dienstbereitstellung und die Dienstanpassung.

Change Management Änderungen an Infrastruktur und IT-Diensten stellen für Provider aufgrund der vielfältigen Abhängigkeiten zwischen Infrastrukturkomponenten eine der größten Herausforderungen dar.

Ziel dieses Prozesses: Sicherstellen, dass alle Veränderungen an Infrastruktur und Diensten in standardisierter Weise bewertet, autorisiert und implementiert werden

Grundsätzliche Aufgaben und Aktivitäten des *Change Managements* sind die Erfassung und Filterung aller Änderungsanfragen, Bewertung, Kategorisierung und Autorisierung geplanter Veränderungen, Planung und Koordination der Entwicklung und Implementierung von Änderungen sowie die Durchführung von *Post Implementation Reviews* zur Beurteilung des Ergebnisses und der Konformität jeder Änderung mit den Soll-Vorgaben aus der Planungsphase.

Die Anforderungsanalyse hat den Anwendungsfall 36 (*Calculate SLA-based change category*) für ein Managementsystem für Service-Level-Management geliefert, der natürlich im Zusammenhang mit diesem Prozess steht. Ziel dieses Anwendungsfalls ist es, das *Change Management* bei der SLA-konformen Bestimmung der Kategorie einer geplanten Veränderung zu unterstützen. Die

Kategorie ergibt sich aus Dringlichkeit und Auswirkung (zusammengefasst zur Priorität) sowie dem Risiko der Veränderung. Entsprechend müssen Dienstvereinbarungen Anhaltspunkte zur Bestimmung dieser drei Größen im Hinblick auf die Änderung an einem bestimmten Dienst enthalten. Zur Bestimmung der Kategorie müssen gegebenenfalls Ziele und Festlegungen mehrerer SLAs über den gleichen Dienst korreliert werden.

Die Informationsflüsse zwischen Service-Level-Management und *Change Management* sind demnach:

- SLM → Change Management: Aus SLAs abgeleitete dienstspezifische Einflussfaktoren auf die Dringlichkeit, die Auswirkung und das Risiko geplanter Veränderungen
- SLM ← Change Management: Änderungsplanung, Kategorisierungsanfragen

Release Management Während sich das *Change Management* mit der Bewertung, Autorisierung, Planung und Entwicklung einzelner Änderungen befasst, liegt es in der Verantwortung des *Release Managements*, freigegebene Änderungen zu paketieren, gemeinsam zu testen und in kontrollierter Weise auszurollen.

Ziel dieses Prozesses: Verteilung einer oder mehrerer Änderungen in Form eines *Releases* in die Live-Umgebung einschließlich Release-Planung und -Dokumentation

In diesen Kontext ist der Anwendungsfall 37 (*Determine transition constraints*) einzuordnen. Denn sowohl für die Release-Planung als auch die tatsächliche Durchführung (*Rollout*) können sich aus Dienstvereinbarungen einschränkende Randbedingungen wie festgelegte Wartungsfenster ergeben.

Vor diesem Hintergrund ergeben sich die folgenden wesentlichen Informationsflüsse zwischen Service-Level-Management und *Release Management*:

- SLM → Release Management: Randbedingungen für die Planung und Durchführung konkreter Releases, die sich aus bestehenden SLAs ergeben
- SLM ← Release Management: Release-Richtlinie, Release-Planung, Anfragen zu Randbedingungen der Transition von Diensten, Informationen zur Freigabe/Ausrollung von Diensten

Configuration Management Im Mittelpunkt dieses Prozesses steht der Aufbau und die Pflege einer Konfigurationsmanagementdatenbank (*Configuration Management Database*, CMDB), die alle prozessübergreifend relevanten Informationen zu Konfigurationselementen (*Configuration Items*, CIs) enthält – darunter Infrastrukturkomponenten und Dienste – und ihre Beziehungen dokumentiert.

Ziel dieses Prozesses: Dokumentation und Kontrolle von Dienst- und Infrastrukturkomponenten sowie Pflege relevanter Konfigurationsinformationen

Die Informationsflüsse zwischen Service-Level-Management und *Configuration Management* umfassen mindestens:

- SLM → Configuration Management: Dienstinformationen (insbesondere Dienstspezifikationen)
- SLM ← Configuration Management: Informationen zu Abhängigkeiten zwischen Diensten und Subdiensten bzw. Diensten und Infrastrukturkomponenten

5.4.2.6 Schnittstellen zu Prozessen der Dienstunterstützung

Die Prozesse des Bereichs *Service Support* sind vor allem während der Betriebsphase eines Dienstes relevant. Beide enthaltenen Prozesse beschäftigen sich mit unerwünschten und ungeplanten Dienst- und Infrastrukturzuständen. Mit dem *Incident Management* wird außerdem eine Schnittstelle zum Dienst-anwender geschaffen.

Incident Management Aufgabe des *Incident Managements* ist es, die vereinbarte und zugesicherte Dienstqualität nach Feststellung einer Störung durch den Provider bzw. Meldung einer Störung durch einen Kunden oder Anwender im Rahmen des vereinbarten Support-Levels wiederherzustellen.

Ziel dieses Prozesses: Incidents schnellstmöglich beheben, um negative Auswirkungen auf die Geschäftsprozesse zu vermeiden

Bei der Priorisierung der Maßnahmen zur Entstörung muss vom *Incident Management* im Sinne eines geschäftsorientierten Vorgehens berücksichtigt werden, wie sich die zeitgleich bestehenden Dienststörungen auf die Geschäftsprozesse der Kunden auswirken. Diese Entscheidung soll aber nicht quasi „zur

Laufzeit“ während der Prozessdurchführung getroffen werden, sondern durch sinnvoll verhandelte SLAs weitgehend vorgegeben sein. So wird ein Kunde für einen geschäftskritischen IT-Dienst sowohl strengere Service-Level-Ziele als auch ein höheres Support-Level, gekennzeichnet durch eine schnellere Reaktion und kürzere zugesicherte Lösungszeiten, abschließen als für einen Dienst, der im Störfall nur eine geringe negative Auswirkung auf einen Geschäftsprozess hat.

Dies erklärt die Notwendigkeit der Anwendungsfälle 33 (*Calculate SLA-based incident urgency*) und 34 (*Determine resolution constraints*), deren Ziel es aus Sicht des Service-Level-Managements ist, das *Incident Management* auf Basis bestehender Dienstvereinbarungen sowohl bei der Priorisierung von Störungen (Anwendungsfall 33) als auch der Ablaufkoordination ihrer Wiederherstellung (Anwendungsfall 34) zu unterstützen.

Die zentrale Rolle des Service-Level-Managements sowie die Wichtigkeit, im Rahmen eines Managementsystems auch und gerade die Beziehungen zu den anderen Dienstmanagementdisziplinen zu berücksichtigen, kommen hier besonders gut zum Ausdruck. Informationen, die zwischen Service-Level-Management und *Incident Management* fließen, sind:

- SLM → *Incident Management*: Informationen zur SLA-basierten Priorisierung von Störungen, Randbedingungen für die Entstörung, anzuwendendes Support-Level in Abhängigkeit vom gestörten Dienst und betroffenen Kunden
- SLM ← *Incident Management*: Informationen zu Störungen, Ausfallzeiten und SLA-kritischen Ereignissen insbesondere im Hinblick auf tatsächliche oder potenzielle SLA-Verletzungen

Problem Management Im Unterschied zum *Incident Management* verfolgt *Problem Management* nicht vorrangig das Ziel einer schnellstmöglichen Wiederherstellung im Fehlerfall, sondern die Lösung der den Störungen zugrunde liegenden Ursachen zur Vermeidung weiterer Störungen. Während *Incident Management* also primär reaktiv agiert, kommt durch das *Problem Management* eine proaktive/präventive Komponente in den Bereich der Dienstunterstützung.

Ziel dieses Prozesses: Störungen schnellstmöglich beheben, um negative Auswirkungen auf die Geschäftsprozesse zu vermeiden

Ein Problem ist nach ITIL [48] und ISO/IEC 20000 [52] definiert als die *unbekannte* Ursache für eine oder mehrere Störungen. Die Philosophie, die hinter der Differenzierung zwischen *Incident Management* und *Problem Management* bzw. zwischen Störungen und Problemen steht, kann schnell falsch interpretiert werden. Insbesondere *wird* aus einer Störung kein Problem, sondern mehrere Störungen können darauf hindeuten, dass es ein Problem *gibt*. So klar diese beiden Prozesse durch ihre Definitionen auch von einander abzugrenzen sind, so schwierig kann es in der operativen Praxis sein, bestimmte Aufgaben in der Fehlerbehandlung – zum Beispiel die Ermittlung der Ursache einer Störung – genau einem der beiden Prozesse zuzuordnen.

Welcher Zusammenhang besteht nun zwischen Service-Level-Management und *Problem Management*? Die Antwort auf diese Frage lautet: In der Tendenz ein ganz ähnlicher wie zwischen Service-Level-Management und *Incident Management*, da natürlich auch im *Problem Management* priorisiert werden muss und sich die Effektivität des Prozesses daran messen lassen muss, welchen Beitrag er zur Erreichung der Service-Level-Ziele geleistet hat.

Die Informationsflüsse sind im Einzelnen:

- SLM \rightarrow *Problem Management*: Informationen zur Geschäftskritikalität von Diensten, vereinbarte Service-Level-Ziele zur korrekten Bestimmung der Dringlichkeit von Problemen
- SLM \leftarrow *Problem Management*: Problemberichte, Nichteinhaltung vereinbarter Service-Level-Ziele nach bzw. aufgrund eines Problems

5.4.3 Zusammenfassung

Das Intraprozess-Kommunikationsmodell hat primär auf Basis von Organisations- und Funktionsmodell alle Kommunikationsbeziehungen innerhalb des Service-Level-Managements identifiziert und klassifiziert. Zu diesem Zweck wurden sowohl eine Identifikationsmethodik als auch ein formaler Klassifikationsansatz entwickelt. Letzterer legt, basierend auf verschiedenen Differenzierungskriterien, fünf Klassen von Kommunikationsmustern fest. Das Ergebnis ist ein vollständiger und strukturierter Überblick über die Aktorenkommunikation innerhalb eines Managementsystems für Service-Level-Management.

Im Rahmen des Interprozess-Kommunikationsmodells wurden die vielfältigen Kommunikationsbeziehungen und damit Abhängigkeiten zwischen dem

Service-Level-Management und anderen Dienstmanagementdisziplinen und -prozessen analysiert und strukturiert dargestellt. Dabei hat sich bestätigt, dass Service-Level-Management sowohl wichtige Informationen *an* andere Prozesse bereitstellen muss, die diese zur effektiven Ausführung benötigen, dass Service-Level-Management aber auch selbst auf Informationen *von* anderen Prozessen angewiesen ist, um die durch die Primäranwendungsfälle gegebenen Anforderungen und im Rahmen des Funktionsmodells der Prozess-Sicht entwickelten Workflows und Aktivitäten durchführen zu können.

Gestützt wurde die Kommunikationsflussanalyse auf ein Referenz-Prozessrahmenwerk, welches sich im Wesentlichen mit dem Prozess-Rahmenwerk aus ISO/IEC 20000 deckt. Damit werden auch die in anderen Ansätzen (zum Beispiel ITIL, MOF oder COBIT) vorgeschlagenen und gängigsten Prozesse im Kern abgedeckt, ohne dass die Arbeit in diesem Punkt zu spezifisch wird. Im Ergebnis ist das Kommunikationsmodell auf dieser Betrachtungsebene der Architektur schon detailliert genug, um die Kommunikationsbeziehungen zu verstehen, aber noch zu abstrakt, um auf ihrer Grundlage ohne weitere Verfeinerungsschritte ein IT-gestütztes Managementsystem zu entwerfen, das die Beziehungen geeignet abbildet. Mit diesen Eigenschaften schlägt es die Brücke zwischen der Anforderungsanalyse und der System-Sicht der Architektur.

5.5 Zusammenfassung und nächste Schritte

In diesem Kapitel wurde, basierend auf den funktionalen Anforderungen und Anwendungsfällen aus Kapitel 3, der erste Haupt-Bestandteil der Managementarchitektur zur Realisierung eines integrierten, IT-gestützten Ansatzes im Service-Level-Management vorgestellt. Im Mittelpunkt standen hierbei Aspekte, die aus Prozess-Sicht Kernanforderungen an ein entsprechendes Managementsystem realisieren.

5.5.1 Ergebnisse der Prozess-Sicht

Im *Organisationsmodell* wurden Domänen und enthaltene Rollen spezifiziert, die für ein SLM-unterstützendes Managementsystem relevant sind. Die drei betrachteten Organisationsdomänen sind in Übereinstimmung mit zahlreichen Modellen, Rahmenwerken und der gängigen Praxis die Provider-, die Kunden- und die Lieferantendomäne. Zudem wurden die für das Funktionsmodell benötigten Verantwortungsmodelle definiert.

Im *Informationsmodell* wurden die erforderlichen Informationsartefakte definiert, die zur Speicherung SLM-relevanter Informationen verwendet werden. Ein Konzeptmodell bildete hierfür die Grundlage. Zu jedem identifizierten Artefakt wurden Klassen von Informationsanforderungen an die im Rahmen der System-Sicht zu entwickelnden Datenmodelle angegeben.

Im *Funktionsmodell* wurden funktionale Aspekte eines Managementsystems für Service-Level-Management auf der Basis der Primäranwendungsfälle adressiert. Dazu wurden zu allen Anwendungsfällen, die durch ein solches Managementsystem abgedeckt werden sollen, Referenz-Workflows mit dem Ziel der Darstellung der Aktivitäten, Kontroll- und Objektflüsse in den identifizierten Funktions(teil)bereichen angegeben. Die vier Haupt-Funktionsbereiche sind das Dienstinformationsmanagement, das Stakeholder-Management, das Vereinbarungmanagement und das Messdaten- und Berichtsmanagement.

Schließlich wurden im *Kommunikationsmodell* alle relevanten Kommunikationsbeziehungen des Managementsystems umfassend analysiert und beschrieben. Dabei wurde zwischen einem Intraprozess-Kommunikationsmodell, das die Kommunikationsbeziehungen zwischen Akteuren innerhalb des Service-Level-Managements abdeckt, und einem Interprozess-Kommunikationsmodell differenziert, welches die Zusammenhänge zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen adressiert.

5.5.2 Abgrenzung

Die Überlegungen der Prozess-Sicht enden dort, wo die enthaltenen Teilmolelle einen Detaillierungsgrad erreicht haben, der ausreicht, um den Prozess *Service-Level-Management* in allen einbezogenen Facetten zu verstehen. Dazu wurde auf dieser Ebene ein formales Prozessmodell entwickelt. Ein solches stellt eine unverzichtbare Grundlage für den Entwurf eines IT-gestützten Managementsystems dar.

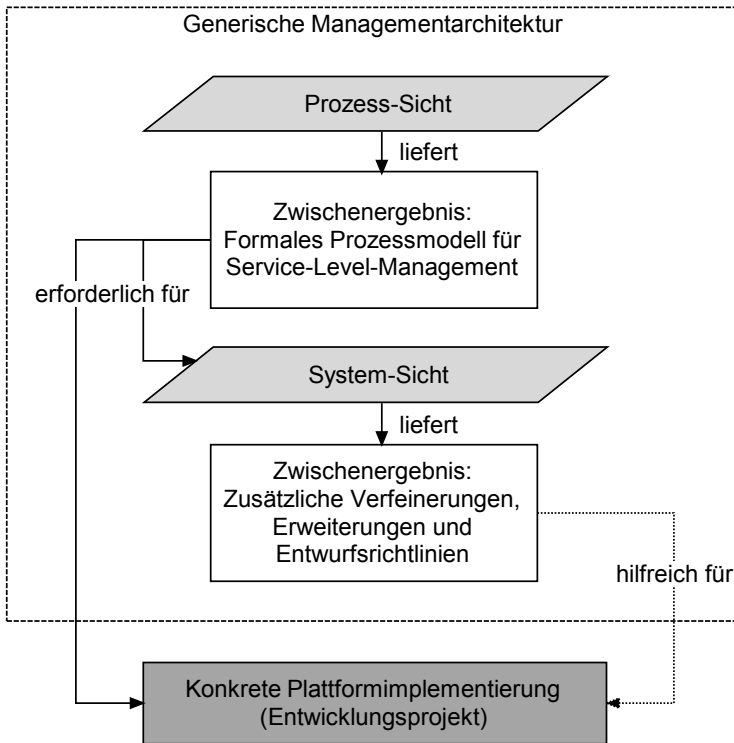


Abbildung 5.54: Abgrenzung von Prozess- und System-Sicht der Managementarchitektur

Ziel der System-Sicht ist es, die Ergebnisse der Prozess-Sicht im Hinblick auf das Ziel *Automatisierung* zu ergänzen. Es ist aber wichtig zu verstehen, dass beide Ebenen Gegenstand einer *generischen* Managementarchitektur sind. Das

<i>Teilmodell</i>	<i>Kapitel 5 (Prozess-Sicht)</i>	<i>Kapitel 6 (System-Sicht)</i>
Organisationsmodell	Domänen, Rollen, Verantwortungsmodelle	Interaktivitätsanalyse, Interaktion und Bediensysteme
Informationsmodell	Konzeptmodell, Informationartefakte, Informationsanforderungen	Datenmodelle, Dokumentstrukturen (Schemata)
Funktionsmodell	Referenz-Workflows, Kontrollflüsse, Informationsflüsse	Automatisierbarkeitsanalyse, Workflow-Erweiterungen, Systementwurfsrichtlinien
Kommunikationsmodell	Intraprozesskommunikation, Interprozesskommunikation	Umsetzbarkeit von Kommunikationsmustern, Managementsystem-Interoperabilität

Tabelle 5.17: Abgrenzung: Prozess-Sicht vs. System-Sicht der Architektur

grenzt die Ebene der System-Sicht insbesondere auch „nach unten“ ab: Alle Elemente der System-Sicht beschäftigen sich mit *technologieunabhängigen* Aspekten des Entwurfs eines IT-gestützten Managementsystems für Service-Level-Management. Die Entscheidungen über die Auswahl und den Einsatz von Technologien (zum Beispiel: Welche Programmiersprache? Welches Datenbankkonzept?) werden erst im Rahmen eines konkreten Entwicklungsprojekts einer Managementplattformimplementierung getroffen, die außerhalb des Betrachtungsbereichs der generischen Architektur liegen. Abbildung 5.54 visualisiert diese Abgrenzungsaspekte.

Die bisher erzielten Ergebnisse bilden also den Ausgangspunkt für das nachfolgende Kapitel und fließen als Eingaben in die Teilmodelle der System-Sicht ein. Tabelle 5.17 stellt die wesentlichen Inhalte der Teilmodelle aus Prozess-Sicht (dieses Kapitel) den intendierten Inhalten der Teilmodelle aus System-Sicht gegenüber, die Gegenstand des nächsten Kapitels sind.

Kapitel 6

System-Sicht der Managementarchitektur

Inhalt dieses Kapitels

6.1 Organisationsmodell	301
6.1.1 Systembenutzerrollen	302
6.1.2 Nutzerschnittstellen und Systeminteraktivität . . .	304
6.1.3 Zugriffskontrolle	308
6.1.4 Zusammenfassung	312
6.2 Informationsmodell	313
6.2.1 Auswahl einer Modellierungssprache	314
6.2.2 Methodisches Vorgehen	315
6.2.3 Datenmodellierung	319
6.2.4 Zusammenfassung	330
6.3 Funktionsmodell	332
6.3.1 Methodischer Ansatz	333
6.3.2 Automatisierung vom Managementfunktionen . . .	338
6.3.3 Zusammenfassung	347
6.4 Kommunikationsmodell	348
6.4.1 Realisierung der Intraprozesskommunikation . . .	349
6.4.2 Realisierung der Interprozesskommunikation . . .	352
6.4.3 Zusammenfassung	356

Ziel dieses Kapitels ist es, die Prozessmodelle des vorangegangenen Kapitels in Modelle der System-Sicht zu überführen und um zusätzliche Entwurfsrichtlinien und -kriterien zu erweitern, um auf diese Weise eine Spezifikation für ein SLM-unterstützendes, IT-gestütztes Managementsystem zu entwickeln, die den durch die (Primär-)Anwendungsfälle gegebenen Anforderungen gerecht wird. Entsprechend dem Ordnungs- und Entwicklungsrahmen mit der horizontalen Partitionierung der beiden mittleren Architekturebenen orientiert sich

auch dieses Kapitel wieder an den vier Teilmodellen, die bereits die Grundlage für die Modelle aus Prozess-Sicht gebildet haben.

Abbildung 6.1 zeigt, dass dieses Kapitel den dritten der vier Kernbestandteile der Managementarchitektur und Ausgangspunkt für den letzten Schritt, die Anleitung zur Überführung in konkrete Softwaretechnologie in Form einer Managementplattform, bildet.

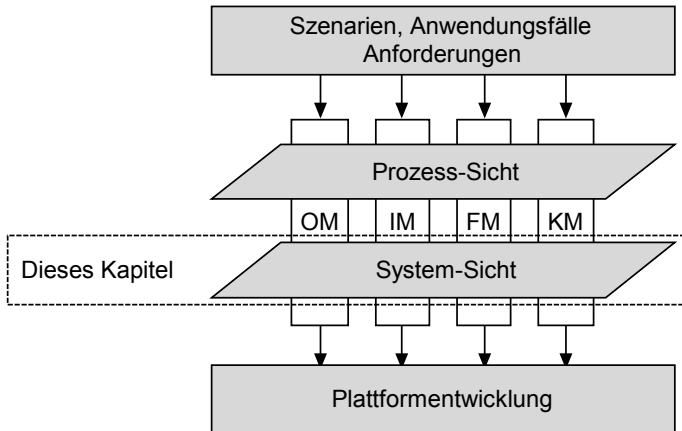


Abbildung 6.1: Einordnung dieses Kapitels in den Ordnungsrahmen

Automatisierungsaspekte stehen in diesem Kapitel – im Gegensatz zum vorherigen – im Vordergrund. Ein Schwerpunkt des Kapitels liegt nicht zuletzt deswegen auf dem Informationsmodell, in dessen Kontext Datenmodelle für SLM-relevante Informationsartefakte in XML-Notation angegeben werden.

Zusätzlich zu Modellverfeinerungen und -erweiterungen werden in diesem Kapitel *Entwurfsrichtlinien* entwickelt und angegeben, die in der Summe eine Art Leitfaden ergeben, der spezifische Aspekte einer Systemimplementierung adressiert und Empfehlungen darüber ausgibt, wie die Architekturteilmodelle beider Sichten (Prozess- und System-Sicht) in diesem Kontext anzuwenden sind.

6.1 Organisationsmodell

Um die Fragestellungen zu ermitteln, die durch das Organisationsmodell der System-Sicht beantwortet werden müssen, müssen die Fragestellungen des Organisationsmodells der Prozess-Sicht auf das Ziel *Automatisierung* projiziert werden.

Zur Erinnerung: Das Organisationsmodell der Prozess-Sicht umfasst

- relevante Organisationsdomänen im Service-Level-Management,
- zugehörige Rollen sowie
- anzuwendende Verantwortungsmodelle.

Aus Sicht der Software- und Systementwicklung sind im Zusammenhang mit solchen Organisationsaspekten unter anderem folgende Punkte von Interesse:

- Abbildung von realen Rollen auf Systembenutzerrollen
- Analyse der Anforderungen an Nutzerschnittstellen und Systeminteraktivität
- Kontrolle und Restriktion der Zugriffe auf Managementfunktionen

Fragestellungen dieses Moduls: Die Fragen, die durch das Organisationsmodell der System-Sicht in erster Linie beantwortet werden müssen, lauten somit:

- Auf welche Benutzerrollen muss der Entwurf eines IT-gestützten Managementsystems für Service-Level-Management ausgerichtet werden?
- Welche Interaktionen müssen zwischen dem Managementsystem und seinen Benutzern über geeignete Schnittstellen ermöglicht werden? Welche konkreten Entwurfskriterien ergeben sich daraus speziell für die Interaktionskanäle und Bedienoberflächen?
- Welcher Ansatz kann zielführend zur Administration und Durchsetzung von Zugriffsrechten auf Systemfunktionen und Informationen eingesetzt werden?

Diese drei Punkte liefern die Struktur für dieses Teilkapitel.

6.1.1 Systembenutzerrollen

Das Organisationsmodell der Prozess-Sicht liefert ein differenziertes Rollenmodell für die Disziplin Service-Level-Management, das auch im Rahmen des Funktions- und Kommunikationsmodells gleicher Ebene zur Anwendung gekommen ist. Es liegt auf der Hand, dass die gleichen Rollen, wie sie in diesem Organisationsmodell eingeführt wurden, auch als Benutzerrollen im Rahmen eines Systementwurfs berücksichtigt werden müssen. Zwei zusätzliche Überlegungen sind allerdings für diesen Teil des Organisationsmodells der System-Sicht anzustellen:

1. Wie hoch ist der Instanzierungsgrad einer Benutzerrolle? Wie viele Systembenutzer füllen also in Abhängigkeit vom betrachteten Szenario parallel die gleiche Rolle aus?

→ Die Kenntnis hierüber kann für einen Systementwurf von Bedeutung sein, da bei Rollen, die einen höhern Instanzierungsgrad als 1 haben, die Betrachtung der eingenommenen Rolle allein beispielsweise im Bereich der Zugriffsrechteverwaltung nicht ausreichend sein kann und eine weitere Differenzierung an dieser Stelle erforderlich ist.

2. Welche zusätzlichen systemrelevanten Rollen werden benötigt?

→ Für einen Systementwurf müssen nicht nur Benutzerrollen – wie durch das Prozess-Organisationsmodell vorgegeben – sondern auch jegliche weiteren Rollen betrachtet werden sein, die in allen Lebenszyklusphasen des Managementsystems erforderlich sind. Hierunter fällt vor allem eine geeignet zu definierende Administrator-Rolle.

6.1.1.1 Instanzierungsgrad von Systembenutzerrollen

Zur ersten Frage ist zu sagen, dass die Rollen *Vertreter einer internen Dienstbringerpartei*, *Kundenvertreter*, *Dienstanwender* und *Lieferantenvertreter* einen Instanzierungsgrad d mit $d \geq 1$ haben. So kann es in einem beliebigen Szenario beispielsweise einen oder mehrere Kunden und somit auch einen oder mehrere Kundenvertreter geben. Bei allen anderen Rollen (zum Beispiel *Service-Level-Manager*) kann davon ausgegangen werden, dass sie in einem Managementsystem zu jeder Zeit von genau einer Person ausgefüllt werden und somit einen Instanzierungsgrad von $d = 1$ aufweisen.

Ein Instanziierungsgrad von größer als 1 bedingt eine differenzierte Benutzerkontenführung für die entsprechenden Rollen als Teil der Funktionalität des IT-gestützten Managementsystems. Beispielsweise muss für jede Instanz der Systembenutzerrolle *Kundenvertreter* bekannt sein, welchem Kunden (Verweis auf das Informationsartefakt *Customer*) sie zuzuordnen ist. Nur so können Zugriffe auf Managementsystemfunktionen und Informationen effektiv gesteuert werden.

Systementwurfsrichtlinie 1: Zu realisierende Systembenutzerrollen sind die Rollen des Organisationsmodells der Prozess-Sicht. Ihr Instanziierungsgrad ist beim Systementwurf zu berücksichtigen und für jede Rolle mit einem Instanziierungsgrad größer als 1 eine differenzierte Benutzerkontenführung zu implementieren.

6.1.1.2 Zusätzliche systemrelevante Rollen

Ein IT-gestütztes Managementsystem für Service-Level-Management basiert auf einer Softwareplattform, welche wie jedes andere Softwaresystem verschiedene Phasen von der System-Anforderungsanalyse bis zur Inbetriebnahme durchläuft (vgl. V-Modell XT [34]). Und auch in der Nutzungsphase können sich Phasen des Betriebs und der Wartung, Modifikation, Anpassung oder Erweiterung abwechseln. Das ist der Grund dafür, dass die alleinige Betrachtung von Systembenutzerrollen nicht ausreicht. Vielmehr müssen mindestens drei weitere Rollen als dauerhaft systemrelevant angesehen werden, nämlich die des Administrators, die des Unterstützers und die des Entwicklers.

Managementsystem-Administrator Der Managementsystem-Administrator (*Super User*) ist verantwortlich für alle anfallenden Aufgaben, die ausgeführt werden müssen, um einen anforderungskonformen Betrieb des Managementsystem zu gewährleisten. Dies schließt folgende (aus Sicht des Service-Level-Managements nicht-funktionale und somit administrative) Aufgabenbereiche ein: Verwaltung von Benutzerkonten, Zugriffen auf Systemfunktionen und Informationen, einfache Konfigurations-, Anpassungs-, Wartungs- und Entstörungsmaßnahmen, Dokumentation und Überwachung der Systemaktivitäten.

Managementplattform-Unterstützer Der Managementplattform-Unterstützer (*Platform Supporter*) ist Ansprechpartner und Bezugsquelle für Informationen, Dokumentation und Unterstützung im Fehlerfall

oder bei Schwierigkeiten im Umgang (beinhaltet sowohl die Verwendung als auch die Administration) mit der Managementplattform.

Managementplattform-Entwickler Der Managementplattform-Entwickler (*Platform Developer*) ist verantwortlich für die (Weiter-)Entwicklung der Managementplattform unter Berücksichtigung erkannter Fehler und neuer oder geänderter Anforderungen an das IT-gestützte Managementsystem und die darunter liegende Plattform.

Diese Rollen werden nicht als Systembenutzerrollen deklariert, da sie nicht in die Nutzungsphase des Managementsystems fallen und aus Sicht des Service-Level-Managements keine funktionale Bedeutung haben.

Systementwurfsrichtlinie 2: Zusätzlich zu den Systembenutzerrollen sind die Rollen Managementsystem-Administrator, Managementplattform-Unterstützer und Managementplattform-Entwickler als systemrelevante Rollen bei der Implementierung einer Managementplattform zu berücksichtigen.

6.1.2 Nutzerschnittstellen und Systeminteraktivität

Ein IT-gestütztes Managementsystem für Service-Level-Management hat die Aufgabe, Verantwortungsträger effektiv bei der Durchführung ihrer SLM-bezogenen Aufgaben zu unterstützen. Entsprechend spielen Interaktionen zwischen Systembenutzern und System eine wichtige Rolle. Adäquate Nutzerschnittstellen in Form geeigneter Bediensysteme (*Frontends*) bereitzustellen gehört zu den Zielen der Implementierung der zugrunde liegenden Plattform.

Zuvor muss versucht werden, auf Basis der verschiedenen Architekturteilmodelle der Prozess-Sicht die Anforderungen hierfür konkret zu analysieren, um den Plattformentwurf in diesem Aspekt zu erleichtern. Zu diesem Zweck wird zunächst untersucht, welche Rollen wie mit dem System interagieren müssen, um was für Interaktionen es sich dabei handelt und welche Schlüsse daraus für den Entwurf von Nutzerschnittstellen zu ziehen sind. Dies geschieht im Rahmen einer Interaktivitätsanalyse.

6.1.2.1 Interaktivitätsanalyse

Die Interaktivitätsanalyse wird in erster Linie abgestützt durch das Organisations-, das Funktions- und das Kommunikationsmodell der Prozess-

Sicht. Aus dem Organisationsmodell werden die Verantwortungsmodelle *verantwortlich (responsible)*, *informiert (informed)*, *unterstützend (supportive)*, *beratend hinzugezogen (consulted)* und *verifizierend (verifying)* herangezogen, um zu analysieren, wie stark die einzelnen Rollen im Hinblick auf diese Verantwortungsmodelle jeweils in die Anwendungsfälle eingebunden sind.

Die nachfolgenden Zusammenhänge ergeben sich sinnvollerweise für die verschiedenen Verantwortungsmodelle im Hinblick auf die *Notwendigkeit einer dedizierten Benutzerschnittstelle*:

- *Verantwortlich*: Eine Nutzerschnittstelle in Form eines Frontends ist in der Regel obligatorisch, um die betreffende Aktivität zu steuern, Informationen aus dem System zu beziehen, Ergebnisse zu dokumentieren oder sonstige im Zusammenhang mit der Aktivität durch die Plattform zu unterstützende Aktionen zu vollziehen.
- *Informiert*: Eine Nutzerschnittstelle in Form eines Frontends kann für einen effektiven Bezug von Informationen sinnvoll sein, wenn auch in vielen Fällen auf alternative bestehende Kanäle zurückgegriffen werden kann wie die automatisierte Benachrichtigung per E-Mail.
- *Unterstützend, beratend hinzugezogen* oder *verifizierend*: Eine Nutzerschnittstelle in Form eines Frontends kann für eine effektive Unterstützung einer Aktivität sinnvoll sein, wenn sie auch in vielen Fällen verzichtbar ist. Aspekte wie Häufigkeit der Aktivität und Zeitkritikalität ihrer Durchführung sind wichtige Faktoren, die bei dieser Entscheidung berücksichtigt werden sollten.

Systementwurfsrichtlinie 3: Die Entscheidung darüber, ob für eine bestimmte Systembenutzerrolle ein eigenes Frontend als Teil des Oberflächenbausteins der Managementplattform entwickelt und bereitgestellt wird oder ob auf bestehende Kommunikationskanäle zurückgegriffen wird, sollte in Abhängigkeit von der Interaktivität der jeweiligen Rolle mit dem Managementsystem – ablesbar aus dem Ergebnis der Interaktivitätsanalyse – getroffen werden.

Bemerkung: Im Falle der Verantwortungsmodelle *informiert (informed)* und *unterstützend (supportive)* kann der Rückgriff auf bestehende Kommunikationskanäle gegebenenfalls auch durch die Anbindung an bestehende, übergeordnete Tools mit Kooperationsunterstützung und Benachrichtigungsfunktionalität realisiert werden.

Generische Ticket-Systeme wie beispielsweise das *Remedy Action Request System* [90] können hierfür eine geeignete Basis-Plattform liefern.

Tabelle 6.1 zeigt das Ergebnis der Interaktivitätsanalyse für die drei wichtigsten Verantwortungsmodelle *verantwortlich*, *informiert* und *unterstützend*, aufgeschlüsselt nach den primären Systembenutzerrollen. Die Referenz-Workflows der Primärszenarien aller funktionalen Anwendungsfälle, wie sie durch das Funktionsmodell der Prozess-Sicht gegeben sind, wurden zu diesem Zweck im Hinblick darauf analysiert, welche Rolle des Organisationsmodells pro Verantwortungsmodell in wie viele Workflow-Aktivitäten eingebunden ist. Dabei wird im Falle des Verantwortungsmodells *responsible* zwischen einfachen und komponierten Aktivitäten differenziert. Der Eintrag 39/8 in Spalte 1 bedeutet also: Die Rolle Service-Level-Manager ist für insgesamt 47 Aktivitäten, davon 39 einfache und 8 komponierte, operativ verantwortlich.

Im Falle der beiden anderen Verantwortungsmodelle wird jeweils unterschieden, wie häufig die betrachtete Rolle Ziel und wie oft Ausgangspunkt der dahinter stehenden Kommunikationsbeziehung ist. Der Eintrag 23/53 in Spalte 2 bedeutet demnach, dass der Service-Level-Manager über alle Workflows insgesamt 23 mal über Ergebnisse von Aktivitäten aus dem Verantwortungsbereich anderer Aktoren informiert wird und selbst 53 mal aus seinem eigenen Verantwortungsbereich heraus andere Aktoren informiert. Dies schließt Objektflüsse über *Swimlane*-Grenzen mit ein. Der Eintrag 15/1 weist schließlich darauf hin, dass der Service-Level-Manager 15 mal von anderen Aktoren unterstützt wird und einmal selbst unterstützend auftritt.

Die Aussagekraft der Tabelle 6.1 muss mit Vorsicht bewertet werden. Denn obwohl sie zweifellos einen guten Ausgangspunkt darstellt, um ein Bild von der Nutzer-System-Interaktivität zu erhalten, muss darauf hingewiesen werden, dass folgende Punkte nicht berücksichtigt werden:

- Verschiedene Anwendungsfälle haben jeweils eine unterschiedliche Instanzierungshäufigkeit, deren konkrete Ausprägung maßgeblich vom jeweiligen Einsatzszenario abhängt. Um dies in der Interaktivitätsanalyse zu berücksichtigen, müsste man alle Anwendungsfälle in Relation zueinander gewichten und die sich ergebende Gewichtungsfunktion auf die Werte der Tabelle 6.1 anwenden.
- Einige Referenz-Workflows enthalten Verzweigungen, von denen in einer Ablaufinstanz nicht alle Äste zur Ausführung kommen müssen. Umgekehrt können in einigen Workflows Ablaufzyklen entstehen, die die

	<i>Verantwortlich/ Composite</i>	<i>Informiert/ informierend</i>	<i>Unterstützt/ unterstützend</i>
<i>Service-Level-Manager</i>	39/8	23/53	15/1
<i>Dienstportfolio-verwalter</i>	15/6	5/21	3/0
<i>Kontraktverwalter</i>	2/4	15/3	0/2
<i>Anderer Prozessmanager</i>	1/10	10/7	0/0
<i>Interner Vertreter</i>	0/0	8/0	0/7
<i>Kundenvertreter</i>	2/0/0	16/2	0/12
<i>Dienstanwender</i>	0/0/0	0/0	0/0
<i>Lieferantenvertreter</i>	0/0/0	8/0	0/7

Tabelle 6.1: Interaktivitätsanalyse

gleiche Aktivität mehrfach zur Ausführung bringen, bevor der Zyklus verlassen wird. In solchen Fällen ist die tatsächliche Nutzer-System-Interaktivität im Kontext des betreffenden Anwendungsfalls nicht deterministisch und kann von Instanz zu Instanz teilweise erheblich schwanken. Wie ein typisches Ablaufverhalten für einen Workflow aussieht, hängt wieder stark vom Einsatzszenario ab.

- Die Verantwortungsmodelle *beratend hinzugezogen (consulted)* und *verifizierend (verifying)* wurden nicht explizit betrachtet. Eine Rolle, die besonders häufig beratend hinzugezogen wird, ist der Kontraktverwalter.

Im Ergebnis allerdings wird doch ein sehr deutliches Bild davon gezeichnet, welche Rollen in welcher Verantwortung wie stark in das Managementsystem involviert sind, da sich insbesondere die Granularität der Aktivitäten über alle Referenz-Workflows hinweg auf einem gleichen Niveau bewegt und komponierte Aktivitäten (*composite activities*) als solche in der Ergebnistabelle der Interaktivitätsanalyse berücksichtigt werden.

6.1.2.2 Schlussfolgerungen für Nutzerschnittstellen

Nachdem die Interaktivitätsanalyse durchgeführt wurde, müssen nun noch die richtigen Schlussfolgerungen im Hinblick auf ein konkretes Design eines IT-gestützten Managementsystems für Service-Level-Management gezogen werden. Das ist das Ziel dieses Abschnitts.

Grundsätzlich lassen sich die Systembenutzerrollen aufgrund der Ergebnisse der Interaktivitätsanalyse in vier Klassen einteilen:

1. Starke Einbindung im Hinblick auf alle Verantwortungsmodelle: Service-Level-Manager, Dienstportfolio-Verwalter, Kontrakt-Verwalter
2. Starke Einbindung im Hinblick auf die Verantwortungsmodelle *informiert* und *unterstützend*, schwache Einbindung im Hinblick auf das Verantwortungsmodell *verantwortlich*: Kundenvertreter
3. Mittelmäßige Einbindung im Hinblick auf die Verantwortungsmodelle *informiert* und *unterstützend*, keine Einbindung im Hinblick auf das Verantwortungsmodell *verantwortlich*: Interner Vertreter, Lieferantenvertreter
4. Keine Einbindung im Hinblick auf alle Verantwortungsmodelle: Dienst-anwender

Daraus ergibt sich im Hinblick auf die Notwendigkeit und Realisierung von Steuer-Frontends die nachfolgende Entwurfsrichtlinie.

Systementwurfsrichtlinie 4: Für die Systembenutzerrollen Service-Level-Manager, Dienstportfolio-Verwalter und Kontrakt-Verwalter sind Steuer-Frontends zu implementieren, die Schnittstellen zu folgenden Funktionalitäten jeweils in Konformität zum Funktionsmodell der Prozess-Sicht realisieren: Workflow-Initialisierung und -Steuerung, Zugriff auf relevante Informationsartefakte, Anstoßen eines Objektflusses, Abruf bzw. Anfrage von Informationsobjekten.

6.1.3 Zugriffskontrolle

Nachdem der letzte Abschnitt den Aspekt der Schnittstellenrealisierung zwischen IT-gestütztem Managementsystem und Systembenutzern adressiert hat,

also der Frage nachgegangen ist, über *welche Arten von Nutzerschnittstellen* in welchen Fällen eine Nutzer-System-Interaktivität realisiert werden kann, beschäftigt sich dieses Teilkapitel damit, *welche Zugriffe* über diese Nutzerschnittstellen in welchem Kontext zulässig sind. Zwei Arten der Zugriffskontrolle müssen in diesem Zusammenhang unterschieden werden:

- Kontrolle funktionaler Systemzugriffe: Welcher Akteur darf wann auf welche Managementsystemfunktionen (zum Beispiel Ändern einer bestimmten Komponente einer Dienstvereinbarung) zugreifen, also Aktionen durchführen, die unter die Kontrolle des IT-gestützten Managementsystems fallen?
- Kontrolle informationeller Systemzugriffe: Welcher Akteur darf wann und wie auf welche Informationsartefakte zugreifen, die Teil des Informationssystems für Service-Level-Management sind (vgl. nächstes Kapitel)?

Dass zwischen beiden Arten ein enger Zusammenhang besteht, liegt auf der Hand. In der Regel fällt ein Informationszugriff ja nur im Kontext eines funktionalen Systemzugriffs überhaupt an.

6.1.3.1 Kontrolle funktionaler Systemzugriffe

In die Entscheidung über die Legitimation funktionaler Zugriffe auf das Managementsystem fließen nicht nur die Systembenutzerrollen und weiteren systemrelevanten Rollen, sondern abermals auch die verschiedenen Verantwortungsmodelle ein:

- *Verantwortlich*: Ist ein Systembenutzer für die Durchführung einer Aktivität operativ verantwortlich, so benötigt er vollen Zugriff auf alle die Aktivität unterstützenden Managementfunktionen.
- *Informiert*: Ein Systembenutzer, der über das Ergebnis einer Aktivität informiert wird, erhält keinen funktionalen Zugriff auf die Aktivität. Der Kontext, aus dem heraus die Information stattgefunden hat, sollte allerdings für ihn ersichtlich sein.
- *Unterstützend, beratend hinzugezogen* oder *verifizierend*: Im Falle dieser drei Verantwortungsmodelle muss fallspezifisch festgelegt werden, ob Zugriffe auf bestimmte Managementfunktionen, die Teil der unterstützten oder verifizierten Aktivität sind, sinnvollerweise gewährt werden sollten. Dies hängt freilich auch davon ab, ob ein funktionaler Zugriff über die

gewählte Nutzerschnittstelle überhaupt möglich ist. Sichergestellt werden kann dies in jedem Fall, wenn ein eigenes Frontend zur Verfügung steht.

Systementwurfsrichtlinie 5: Um einen kontrollierten Zugriff auf Management-systemfunktionen sicherzustellen und unautorisierte Zugriffe und Aktionen zu vermeiden, sollte die Zugriffssteuerung die Entscheidung über die Erteilung einer Berechtigung auf Basis folgender Parameter treffen:

- Managementsystemfunktion, die ausgeführt werden soll
- Workflow-Kontext (welcher Workflow, welcher Fortschritt im Kontrollfluss?)
- Rolle des zugreifenden Aktors
- Verantwortungsmodell, aus dem heraus der Zugriff erfolgt

6.1.3.2 Kontrolle informationeller Systemzugriffe

Im Zusammenhang mit Informationen und Daten sind grundsätzlich folgende Zugriffsrechte bzw. -einschränkungen zu unterscheiden:

- Kein Zugriff (*no permission*)
- Nur lesender Zugriff (*read only*)
- Lesender und schreibender Zugriff (*read/write*)

Daneben müssen auch die Objekte, auf die sich die zu steuernden Zugriffe beziehen, festgelegt werden. Hierbei muss eine sinnvolle Balance zwischen möglichst feiner Granularität und vertretbarem Implementierungs-, Rechen- und Wartungsaufwand gefunden werden.

Systementwurfsrichtlinie 6: Die Granularität der informationellen Zugriffskontrolle sollte sich mindestens auf Ebene der Informationsartefakte bewegen. Abhängig von deren Ausprägung und Umfang kann auch eine feinere Granularität auf Ebene von Informationsklassen bzw. Artefaktkomponenten sinnvoll sein.

Die Entscheidung über die Autorisierung eines Zugriffs hängt auch hier wieder unter anderem davon ab, in welcher Art von Verantwortlichkeit der jeweilige Akteur in seiner aktuellen Rolle agiert:

- *Verantwortlich*: Ist ein Systembenutzer für die Durchführung einer Aktivität operativ verantwortlich, so benötigt er lesenden Zugriff auf alle Informationsartefakte, die als Teil eines Objektflusses in die betreffende Aktivität eingehen (*Inputs*), sowie lesenden und schreibenden Zugriff auf alle Informationsartefakte, die als Teil eines Objektflusses von der Aktivität ausgehen (*Outputs*).
- *Informiert*: Ein Systembenutzer, der über den Status eines Informationsartefakts informiert wird, erhält nur lesenden Zugriff auf dieses Objekt.
- *Unterstützend, beratend hinzugezogen* oder *verifizierend*: Unterstützt oder verifiziert ein Akteur des Managementsystems eine Aktivität, so benötigt er in aller Regel mindestens lesenden Zugriff auf alle Informationsartefakte, die in diese Aktivität einfließen sowie gegebenenfalls auf weitere Artefakte, die Informationen enthalten, die für die Unterstützung oder Verifizierung benötigt werden. Ein schreibender Zugriff kann fall-spezifisch sinnvoll sein, muss dann allerdings mit schreibenden Zugriffen der verantwortlichen Akteure synchronisiert werden.

Systementwurfsrichtlinie 7: Um einen kontrollierten Zugriff auf Informationsartefakte sicherzustellen und unautorisierte Zugriffe oder Veränderungen zu vermeiden, sollte die Zugriffssteuerung die Entscheidung über die Erteilung einer Berechtigung zum lesenden oder schreibenden Zugriff auf Basis folgender Parameter treffen:

- Objekt, auf das zugegriffen werden soll
- Zugriffsmodus (lesend oder lesend/schreibend)
- Workflow-Kontext (welcher Workflow, welcher Fortschritt im Kontrollfluss?)
- Rolle des zugreifenden Akteurs
- Verantwortungsmodell, aus dem heraus der Zugriff erfolgt

6.1.4 Zusammenfassung

Aufbauend auf dem Organisationsmodell der Prozess-Sicht wurden einige Erweiterungen und Entwurfsrichtlinien formuliert, die einen Anwender der Managementarchitektur für Service-Level-Management in die Lage versetzen sollen, einen konkreten System-Feinentwurf so zu entwickeln, dass er mit den Anforderungen und Modellen der Prozess-Sicht harmonisiert. Das gleiche Ziel wird auch für die noch fehlenden drei Teilmodelle verfolgt.

Konkret wurde das Rollenmodell der Prozess-Sicht als Grundlage für die Definition der Systembenutzerrollen verwendet, wobei zur Feststellung der Notwendigkeit einer Benutzerkontenverwaltung der Instanzierungsgrad jeder Rolle analysiert wurde. Zusätzliche systemrelevante Rollen wurden untersucht und schließlich sowohl Aspekte der Kontrolle funktionaler als auch informationeller Systemzugriffe ausgearbeitet.

6.2 Informationsmodell

Die Informationsmodellierung auf der Ebene der System-Sicht gehört zu den komplexesten Aufgaben im Zusammenhang mit dem Entwurf der Managementarchitektur. Gleichzeitig bilden Informationen aber den Kern jedes Managementsystems, und insofern gilt das Informationsmodell als Herzstück eines solchen.

Zur Erinnerung: Das Informationsmodell der Prozess-Sicht umfasst

- ein grundlegendes Konzeptmodell für Service-Level-Management und einige exemplarische Erweiterungen,
- Informationsartefakte zur Realisierung dieser Grundkonzepte sowie
- Klassen von Informationsanforderungen je Artefakt.

Während also auf der Ebene der Prozess-Sicht bereits die Fragen beantwortet wurden, welche Informationsartefakte im Service-Level-Management benötigt werden, wie sie zusammenhängen und welche Informationsanforderungen an jedes Artefakt bestehen, beschäftigt sich dieses Teilkapitel mit Aspekten der Datenmodellierung, also der Überführung der Elemente des Prozess-Informationsmodells in ein Computer-verarbeitbares Format unter Modellierung aller Informationsklassen.

Fragestellungen dieses Moduls: Das Informationsmodell der System-Sicht soll somit Antworten auf die folgenden Fragen geben:

- Wie lassen sich die Anforderungen an die Informationsartefakte zu formalisierten Datenmodellen verfeinern?
- Welche konkreten Datenmodelle ergeben sich dieser Methodik folgend für die Informationsartefakte im Service-Level-Management?
- Wie lassen sich diese Datenmodelle zu einem übergeordneten, konsistenten Modell eines Informationsmanagementsystems zusammenführen?

6.2.1 Auswahl einer Modellierungssprache

Um diese Punkte durch das Informationsmodell der System-Sicht zu adressieren, muss zunächst ein geeigneter Modellierungsansatz gewählt werden, der mächtig genug ist, um die Anforderungen an die Datenmodellierung der System-Sicht erfüllen zu können. Zudem sollen die erzeugten Datenmodelle in Computer-verarbeitbarer Form vorliegen, ohne dabei einschränkende Vorgaben zum konkret zu wählenden Implementierungsansatz im Rahmen des Plattformentwurfs zu machen.

Eine Modellierungssprache, die diesen Anforderungen gerecht wird, ist *XML Schema Definition* (XSD) – oder kurz: *XML Schema*. XSD ist ein W3C-Standard [91, 92] für die Definition von XML-Dokumenttypen. In dieser Eigenschaft eignet sich XSD zur Modellierung der Informationsartefakte im Rahmen der Architektur. Die so entstehenden Schemadefinitionen sind wohlgeformt, gegen die *XML Schema*-Grammatik validierbar, leicht erweiter- und modifizierbar und durch ihre formale Syntax Computer-verarbeitbar. Sie lassen sich außerdem in verschiedenartiger Weise visualisieren und in andere Modellierungsansätze (darunter zum Beispiel UML) transformieren. Damit bietet *XML Schema* als Datenmodellierungssprache ideale Eigenschaften, um der „Baukasten“-Idee der Architektur zu entsprechen.

Ziel ist es, unter Verwendung von *XML Schema* die Elemente des Informationsmodells der Prozess-Sicht, also alle Artefaktklassen und Klassen von Informationsanforderungen, in einer Weise zu modellieren, dass die resultierenden Datenmodelle als konzeptionelle Vorgabe und syntaktische Vorlage zur Realisierung eines Informationsmanagementsystems für Service-Level-Management fungieren können. Die beiden wesentlichen Aspekte hierbei sind:

- Adäquate und anforderungskonforme Definition und Modellierung der Artefakt-Typen
- Modellierung von Beziehungen und Abhängigkeiten zwischen Informationsartefakten und ihren Komponenten

Abschnitt 6.2.2 beschreibt zunächst das grundsätzliche methodische Vorgehen, Abschnitt 6.2.3 gibt anschließend einen Überblick über konkrete Datenmodelle ausgewählter Informationsartefakte und geht dabei auf spezifische Besonderheiten der Modellierung ein.

6.2.2 Methodisches Vorgehen

Dieser Abschnitt fasst die wichtigsten Schritte zusammen, die ausgehend vom Informationsmodell der Prozess-Sicht unternommen werden, um zu einem vollständigen und vollwertigen Datenmodell zu gelangen.

Modularisierungskonzept Das gesamte Datenmodell wird in Module unterteilt, die den identifizierten Informationsartefakten vom Typ Datenbank entsprechen. Jedes Modul kapselt somit den Teil des Datenmodells, der sich auf Informationsartefakte bezieht, die in der jeweiligen Datenbank gespeichert sind. Jedes Datenmodell-Modul besteht aus genau einer XSD-Datei.

Modellierung der Artefakt-Typen Mit Ausnahme der Datenbank-Artefakte müssen zu allen Informationsartefakten Typdefinitionen angegeben werden, die spezifizieren, gegen welche syntaktischen Vorgaben (formal: gegen welche Grammatik) eine gültige Instanz des jeweiligen Artefakt-Typs validierbar sein muss. Hierzu wird im entsprechenden XSD-Modul ein komplexer Datentyp (`xs:complexType`) definiert, der wiederum eine Sequenz von Unterelementen definiert, aus denen sich der komplexe Artefakt-Typ zusammensetzt. Dem Aufbau des Informationsmodells der Prozess-Sicht folgend entsprechen diese Unterelemente den Klassen an Informationsanforderungen je Artefaktklasse.

Modellierung der Informationsklassen Die Datenmodellierung der Unterelemente jedes Artefakt-Typs stellt den wichtigsten Verfeinerungsschritt im Rahmen des Aufbaus des Informationsmodells der System-Sicht dar. Grundsätzlich sind hier drei Varianten denkbar:

1. Die Informationsanforderungen an das Unterelement können durch einen vordefinierten XSD-Datentypen (wie zum Beispiel `xs:string`, `xs:integer`, `xs:date` oder `xs:time`) abgedeckt werden. In diesem Fall wird der entsprechende Typ gesetzt. *XML Schema* liefert eine Reihe vordefinierter Datentypen in den Kategorien Zeichenketten (*string types*), Numerik (*numeric types*) und Datum/Zeit (*date types*).
2. Die Informationsanforderungen an das Unterelement können basierend auf einem der vordefinierten XSD-Datentypen realisiert werden, wobei nicht der gesamte Wertebereich zur Anwendung kommt, sondern dieser eingeschränkt wird. In diesem Fall wird ein einfacher Typ (`xs:simpleType`) als Beschränkung des Wertebereichs eines Basistyps

(`xs:restriction base="..."`) definiert. Im Falle eines numerischen Basistyps besteht die Beschränkung oft aus der Angabe eines geschlossenen Intervalls zulässiger Werte, bei Basistypen aus der Kategorie der *string types* können zulässige *Patterns* oder auch dedizierte zulässige Werte als Teil einer Aufzählung (`xs:enumeration value="..."`) angegeben werden.

3. Wenn weder ein vordefinierter XSD-Datentyp noch ein einfacher Datentyp als Restriktion eines Basistyps in Frage kommen, wird für das jeweilige Unterelement ein neuer komplexer Datentyp (`xs:complexType`) definiert, der seinerseits Unterelemente verschiedener Typen enthalten kann.

Modellierung von Identifikationsinformationen Um Identifikationsinformationen im Rahmen des Datenmodells abzubilden, werden künstliche Schlüssel in Form von Objekt-IDs eingeführt. Dazu wird zu jedem Element, welches einen Primärschlüssel benötigt, ein Attribut mit dem Namen `id` definiert und als zwingend erforderlich (`use="required"`) deklariert.

Generisches XSD-Modul Die Einhaltung der gerade beschriebenen Schritte verfolgt vor allem das Ziel einer konsistenten und einheitlichen Modellierung aller Bestandteile des Datenmodells. Dies macht es auch möglich, ein *generisches XSD-Modul* anzugeben, das die Struktur aller konkret zu entwickelnden Module beschreibt und vorgibt. Listing 6.1 stellt dieses generische Modul dar. Es enthält `complexType`-Definitionen für das Datenbank-Artefakt, welches durch dieses Modul repräsentiert wird, sowie für die enthaltenen Informationsartefakte. Darüber hinaus sind `complexType`- und `simpleType`-Definitionen für alle weiteren benötigten Typen, insbesondere von Artefakt-Komponenten, vorgesehen.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:complexType name="type_...">
4     ...
5   </xs:complexType>
6   ...
7
8   <xs:simpleType name="type_...">
9     <xs:restriction base="xs:...">
10       ...
11     </xs:restriction>
12   </xs:simpleType>
```

```

13  ...
15  <xs:complexType name="type_ARTIFACT">
16    <xs:sequence>
17      <xs:element name="ARTIFACT_COMPONENT" type="..." />
18      ...
19    </xs:sequence>
20    <xs:attribute name="id" type="xs:string" use="required" />
21  </xs:complexType>
22  ...
23
24  <xs:complexType name="DATABASE_ARTIFACT">
25    <xs:sequence>
26      <xs:element name="ARTIFACT" type="type_ARTIFACT" />
27    </xs:sequence>
28  </xs:complexType>
29 </xs:schema>

```

Listing 6.1: Generisches XSD-Modul zur Datenmodellierung von Informationsartefakten

Umfassendes Informationssystem Jedes einzelne auf diese Weise erstellte XSD-Modul kapselt einen Teil des Informations- und Datenmodells für Service-Level-Management. Diese Einzelbestandteile zu einem zusammenhängenden Datenmodell zu integrieren, muss das Ziel eines *SLM-Informationssystem*s sein, das somit den gemeinsamen Überbau über die verschiedenen Module bildet.

Modellierung von Referenzen zwischen Artefakten Die Assoziationen zwischen verschiedenen Informationsartefakten, wie sie durch das Informationsmodell der Prozess-Sicht definiert werden, müssen innerhalb des Datenmodells umgesetzt werden. Dies geschieht durch Deklarationen von Primär- und Fremdschlüsseln, die ebenfalls Teil des Informationssystems sind.

Primärschlüsseldeklarationen Damit die Objekt-IDs auch als Primärschlüssel verwendet werden können, müssen sie unter Verwendung des `<key>`-Elements als solche deklariert werden. Mittels mehrfacher Angabe des Unterelements `<field>` könnten Schlüssel definiert werden, die aus verschiedenen Elementen und Attributen zusammengesetzt sind. Da mit der Einführung von Objekt-IDs allerdings künstliche Schlüsselkandidaten verfügbar sind, besteht hierfür keine Notwendigkeit.

Fremdschlüsseldeklarationen Um Fremdschlüssel zu deklarieren, wird mit dem `<keyref>`-Element gearbeitet, dessen Unterelemente `<selector>` und `<field>` einen XPath-Ausdruck formen, der Attribut(e) und Element(e) spezifiziert, die den Fremdschlüssel bilden. Über das `refer="..."`-Attribut des `<keyref>`-Elements kann genau festgelegt werden, auf welchen Primärschlüssel sich die `<keyref>`-Definition bezieht.

Modellierung von Referenzen zwischen Artefakt-Komponenten Neben den Assoziationen zwischen verschiedenen Informationsartefakten – wie durch das Informationsmodell der Prozess-Sicht vorgegeben – muss das Informations- und Datenmodell der System-Sicht auch Referenzierungen auf tieferen Ebenen, also ganz besonders zwischen Komponenten von Informationsartefakten, enthalten und modellieren. Diese Beziehungen müssen zuvor natürlich identifiziert werden, was neben der Datenmodellierung der Informationsklassen einen weiteren wichtigen konzeptionellen Verfeinerungsschritt zwischen Prozess- und System-Sicht darstellt. Hierzu zwei Beispiele:

- Jedes in einer Dienstvereinbarung enthaltene Service-Level-Ziel bezieht sich auf eine konkrete Service-Level-Metrik aus dem assoziierten Dienstartefakt. Daher muss jedes Service-Level-Ziel einer Dienstvereinbarung eine Service-Level-Metrik referenzieren.
- Dienstvereinbarungen werden durch operative Vereinbarungen (OLAs) und Verträge mit externen Lieferanten (UCs) abgestützt, um eine nahtlose Dienstleistung zu fördern und die Service-Level-Ziele aus den SLAs durch entsprechende *sub level targets* zu untermauern. Auch diese Zusammenhänge müssen mit Hilfe des Datenmodells darstellbar sein. Jedes Service-Level-Ziel muss also ein oder mehrere *sub level targets* referenzieren können.

Zur Realisierung wird ebenfalls wieder mit den Elementen `<key>` und `<keyref>` gearbeitet, sodass die Referenzen zwischen Artefakt-Komponenten syntaktisch identisch mit denen zwischen Artefakten sind.

Listing 6.2 zeigt den resultierenden generischen Aufbau der Schemadefinition für das SLM-Informationssystem .

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:include schemaLocation="DATABASE_ARTIFACT.xsd"/>
4   ...
5
```

```

7   <xs:element name="ServiceLevelManagementInformationSystem">
    <xs:complexType>
      <xs:all>
9        <xs:element name="DATABASE_ARTIFACT" type="
            type_DATABASE_ARTIFACT"/>
            ...
11       </xs:all>
    </xs:complexType>

13   <xs:key name="ARTIFACT_Identification">
15     <xs:selector xpath="./DATABASE_ARTIFACT/ARTIFACT"/>
17     <xs:field xpath="@id"/>
    </xs:key>
    ...

19   <xs:keyref name="referenceTo...FromARTIFACT" refer="
            ARTIFACT_Identification">
21     <xs:selector xpath="./.../.../referenceTo...
            FromARTIFACT"/>
23     <xs:field xpath="@idref"/>
    </xs:keyref>
    ...

25   </xs:element>
27 </xs:schema>

```

Listing 6.2: Generische XSD-Datei zur Definition eines XML-Schemas für ein SLM-Informationssystem

6.2.3 Datenmodellierung

Gemäß der im vorigen Abschnitt entwickelten Modellierungsschritte können nun die einzelnen XSD-Module sowie die Schemadefinition für das SLM-Informationssystem in einheitlicher Weise entwickelt werden. Abschnitt 6.2.3.1 gibt zunächst einen Überblick über alle Bestandteile des Informations- und Datenmodells der System-Sicht, Abschnitt 6.2.3.2 stellt danach einige ausgewählte Modellelemente vor.

6.2.3.1 Überblick

Der zentrale Kern des Informations- und Datenmodells für Service-Level Management auf der Ebene der System-Sicht der Architektur ist das Modul

`ServiceLevelManagementInformationSystem`, das wie auch alle anderen Module mit der Modellierungssprache *XML Schema Definition* spezifiziert wurde. Es enthält allerdings keinerlei Typdefinitionen, sondern definiert ein SLM-Informationssystem als eine Menge miteinander verknüpfter (logischer) Datenbanken. Zur Verknüpfung dieser Datenbanken werden Schnittstellen benötigt, über die mittels Fremdschlüsseln auf enthaltene Objekte zugegriffen werden kann.

Die vollständige Schemadefinition ist in Listing A.1 enthalten und definiert sowohl die Datenbankartefakte als auch alle Primärschlüssel-Fremdschlüssel-Beziehungen, die Teil dieses Informationssystems sind. Es inkludiert alle anderen XSD-Module, sodass die dort jeweils enthaltenen Typ-Definitionen für die Datenbankartefakte und alle anderen Informationsartefakte im Sichtbarkeitsbereich des Informationssystems liegen. Damit ist dieses Informationssystem vollständig spezifiziert und modelliert. Abbildung 6.2 visualisiert dieses Modul auszugswise. Die weißen Kästen mit schwarzer Umrahmung stellen hier nun nicht mehr – wie in den UML-Modellen des Informationsmodells der Prozess-Sicht – Artefaktklassen, sondern konkrete Instanzen dieser Klassen, also Objekte, dar. Multiplizitäten sind ebenfalls angetragen. Ein umrahmtes `+`-Symbol an einem Objekt deutet an, dass das Datenmodell an dieser Stelle weiter entfaltet werden kann, an dieser Stelle also aus weiteren Objekten besteht, die (aus Gründen des Umfangs und der Übersichtlichkeit) im Diagramm nicht dargestellt sind.

Eine gestrichelte Umrahmung indiziert, dass nicht zwangsläufig eine Instanz des jeweiligen Objekts existieren muss, damit das Informationssystem an sich existieren kann. Die Minimalconfiguration besteht somit aus dem Vorhandensein der neun Datenbanken in jeweils genau einer Instanz unter Realisierung der in der Visualisierung als **constraints** angezeigten Zusatzbedingungen. Diese Zusatzbedingungen enthalten die Definitionen aller Primär- und Fremdschlüssel des gesamten Informationssystems. Abbildung 6.3 zeigt ausschnittartig eine Visualisierung von zwei Primärschlüssel- und zwei Fremdschlüsseldefinitionen am Beispiel des referenziellen Zusammenhangs zwischen den Artefaktklassen `Service`, `ServiceCatalog` und `ServiceLevelAgreement`.

Systementwurfsrichtlinie 8: Zur Realisierung eines IT-gestützten Informationssystems für Service-Level-Management als Teil einer Systemarchitektur sollte ein System aus föderierten (logischen oder physischen) Datenbanken in Konformität zu den Spezifikationen aus dem Informationsmodell der System-Sicht der Architektur aufgebaut werden. Die wichtigsten zu beachtenden Punkte sind:

- Geeignete Modularisierung (vgl. Modularisierungskonzept des Informations-

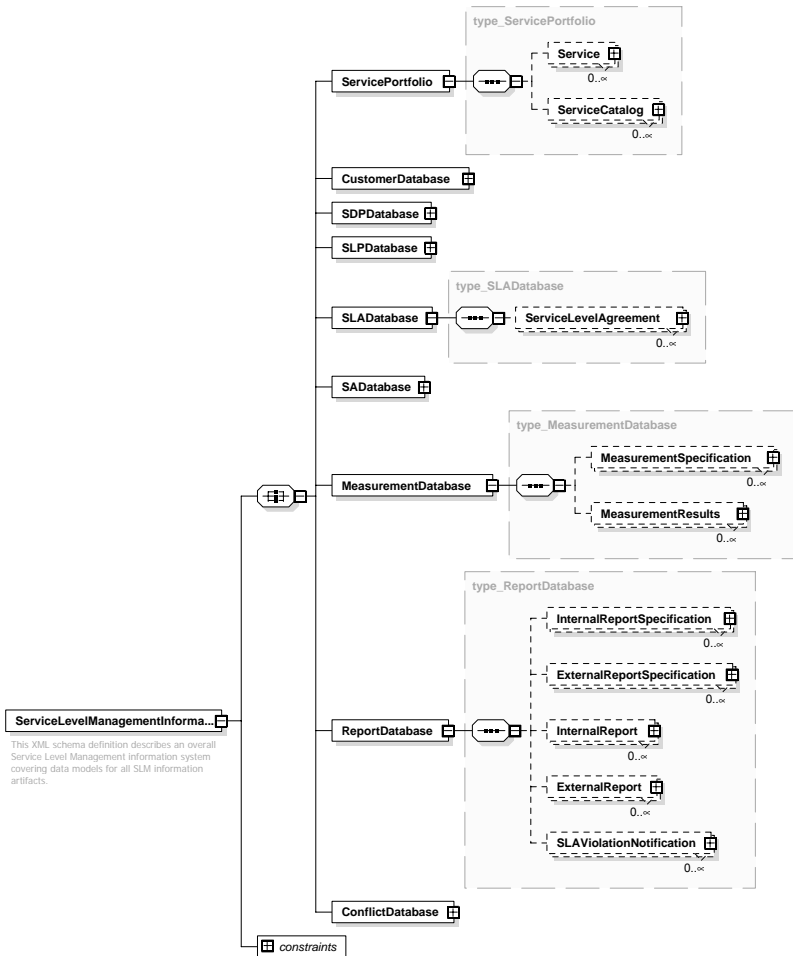


Abbildung 6.2: Visualisierung des Datenmodells zum ServiceLevelManagementInformationSystem (Auszug)

Comment	KEY DEFINITIONS												
Comment	ServicePortfolio												
▲ xs:key	<table border="1"> <tr> <td>name</td> <td>serviceIdentification</td> </tr> <tr> <td>▲ xs:selector</td> <td> <table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/Service</td> </tr> </table> </td> </tr> <tr> <td>▲ xs:field</td> <td> <table border="1"> <tr> <td>xpath</td> <td>@id</td> </tr> </table> </td> </tr> </table>	name	serviceIdentification	▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/Service</td> </tr> </table>	xpath	./ServicePortfolio/Service	▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@id</td> </tr> </table>	xpath	@id		
name	serviceIdentification												
▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/Service</td> </tr> </table>	xpath	./ServicePortfolio/Service										
xpath	./ServicePortfolio/Service												
▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@id</td> </tr> </table>	xpath	@id										
xpath	@id												
▲ xs:key	<table border="1"> <tr> <td>name</td> <td>serviceCatalogServiceIdentification</td> </tr> <tr> <td>▲ xs:selector</td> <td> <table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog</td> </tr> </table> </td> </tr> <tr> <td>▲ xs:field</td> <td> <table border="1"> <tr> <td>xpath</td> <td>@id</td> </tr> </table> </td> </tr> </table>	name	serviceCatalogServiceIdentification	▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog</td> </tr> </table>	xpath	./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog	▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@id</td> </tr> </table>	xpath	@id		
name	serviceCatalogServiceIdentification												
▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog</td> </tr> </table>	xpath	./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog										
xpath	./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog												
▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@id</td> </tr> </table>	xpath	@id										
xpath	@id												
Comment	KEYREF DEFINITIONS												
Comment	From ServicePortfolio												
▲ xs:keyref	<table border="1"> <tr> <td>name</td> <td>referenceToServiceFromServiceCatalog</td> </tr> <tr> <td>refer</td> <td>serviceIdentification</td> </tr> <tr> <td>▲ xs:selector</td> <td> <table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog</td> </tr> </table> </td> </tr> <tr> <td>▲ xs:field</td> <td> <table border="1"> <tr> <td>xpath</td> <td>@idref</td> </tr> </table> </td> </tr> </table>	name	referenceToServiceFromServiceCatalog	refer	serviceIdentification	▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog</td> </tr> </table>	xpath	./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog	▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@idref</td> </tr> </table>	xpath	@idref
name	referenceToServiceFromServiceCatalog												
refer	serviceIdentification												
▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog</td> </tr> </table>	xpath	./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog										
xpath	./ServicePortfolio/ServiceCatalog/referenceToServiceFromServiceCatalog												
▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@idref</td> </tr> </table>	xpath	@idref										
xpath	@idref												
▲ xs:keyref	<table border="1"> <tr> <td>name</td> <td>referenceToServiceFromSLA</td> </tr> <tr> <td>refer</td> <td>serviceCatalogServiceIdentification</td> </tr> <tr> <td>▲ xs:selector</td> <td> <table border="1"> <tr> <td>xpath</td> <td>./SLADatabase/ServiceLevelAgreement/referenceToServiceFromSLA</td> </tr> </table> </td> </tr> <tr> <td>▲ xs:field</td> <td> <table border="1"> <tr> <td>xpath</td> <td>@idref</td> </tr> </table> </td> </tr> </table>	name	referenceToServiceFromSLA	refer	serviceCatalogServiceIdentification	▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./SLADatabase/ServiceLevelAgreement/referenceToServiceFromSLA</td> </tr> </table>	xpath	./SLADatabase/ServiceLevelAgreement/referenceToServiceFromSLA	▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@idref</td> </tr> </table>	xpath	@idref
name	referenceToServiceFromSLA												
refer	serviceCatalogServiceIdentification												
▲ xs:selector	<table border="1"> <tr> <td>xpath</td> <td>./SLADatabase/ServiceLevelAgreement/referenceToServiceFromSLA</td> </tr> </table>	xpath	./SLADatabase/ServiceLevelAgreement/referenceToServiceFromSLA										
xpath	./SLADatabase/ServiceLevelAgreement/referenceToServiceFromSLA												
▲ xs:field	<table border="1"> <tr> <td>xpath</td> <td>@idref</td> </tr> </table>	xpath	@idref										
xpath	@idref												

Abbildung 6.3: Visualisierung des Datenmodells zum ServiceLevelManagementInformationSystem (Auszug)

modells)

- Sinnvolle und anforderungskonforme Datenmodellierung und Ableitung geeigneter Datenbankschemata (vgl. XSD-Module)
- Abbildung von Referenzen zwischen Objekten

Tabelle 6.2 gibt einen Überblick über alle auf der Ebene der Prozess-Sicht des Informationsmodells identifizierten Informationsartefakte, geordnet nach ihren Typen. Die dritte Tabellenspalte verweist für jede Artefaktklasse auf das jeweilige Teilkapitel aus dem Informationsmodell der Prozess-Sicht, in dem die Artefaktklasse eingeführt wurde, und außerdem auf den Teil des Anhangs, der die vollständige XML-Schemadefinition enthält, die das Datenmodell für das jeweilige Artefakt beinhaltet.

6.2.3.2 Ausgewählte Visualisierungen von Artefakt-Datenmodellen

Exemplarisch werden nun einige Informationsartefakte herausgegriffen und ihre Typdefinitionen (zum Teil ausschnittartig) visualisiert. Die Visualisierungen wurden mit Hilfe von *Altova XMLSpy*¹ aus den XSD-Modulen generiert und basieren auf einem an der UML angelehnten, aber für XML-Formate optimierten Metamodell. Die Typdefinitionen der folgenden Artefakte werden vorgestellt:

- Datenbank-Artefakt *ServicePortfolio*: Abbildung 6.4
- Datensatz-Artefakt *Customer*: Abbildung 6.5
- Datenbanksicht-Artefakt *ServiceCatalog*: Abbildung 6.6
- Dokumentations-Artefakt *ServiceLevelAgreement*: Abbildung 6.7
- Datensatz-Artefakt *MeasurementSpecification*: Abbildung 6.8
- Datensatz-Artefakt *ExternalReportSpecification*: Abbildung 6.9
- Berichts-Artefakt *ExternalReport*: Abbildung 6.10

¹<http://www.altova.com>

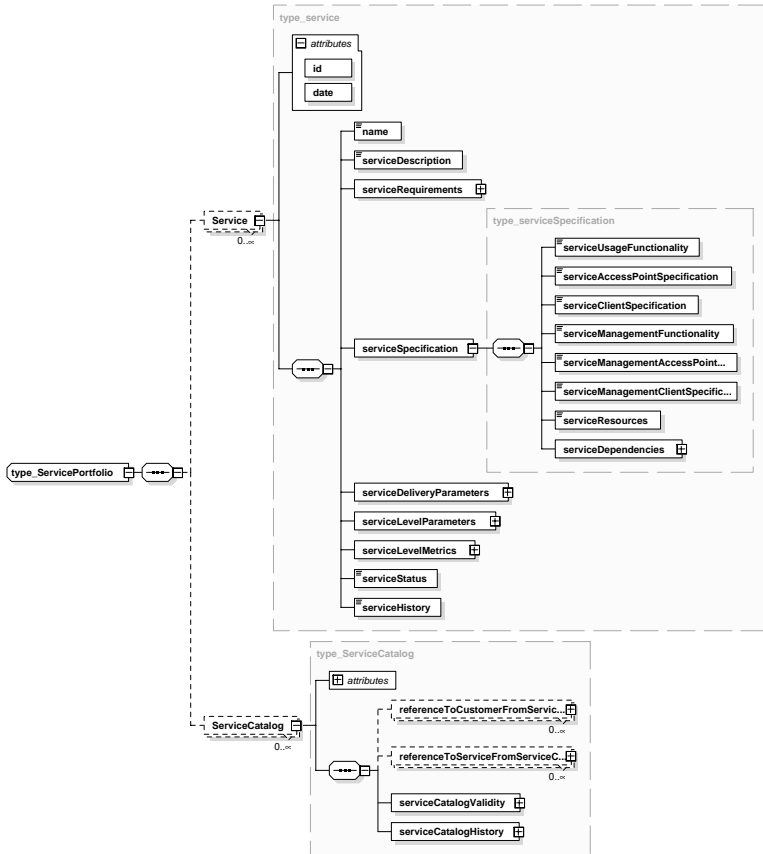


Abbildung 6.4: Visualisierung des Datenmodells zum Informationsartefakt ServicePortfolio (Auszug)

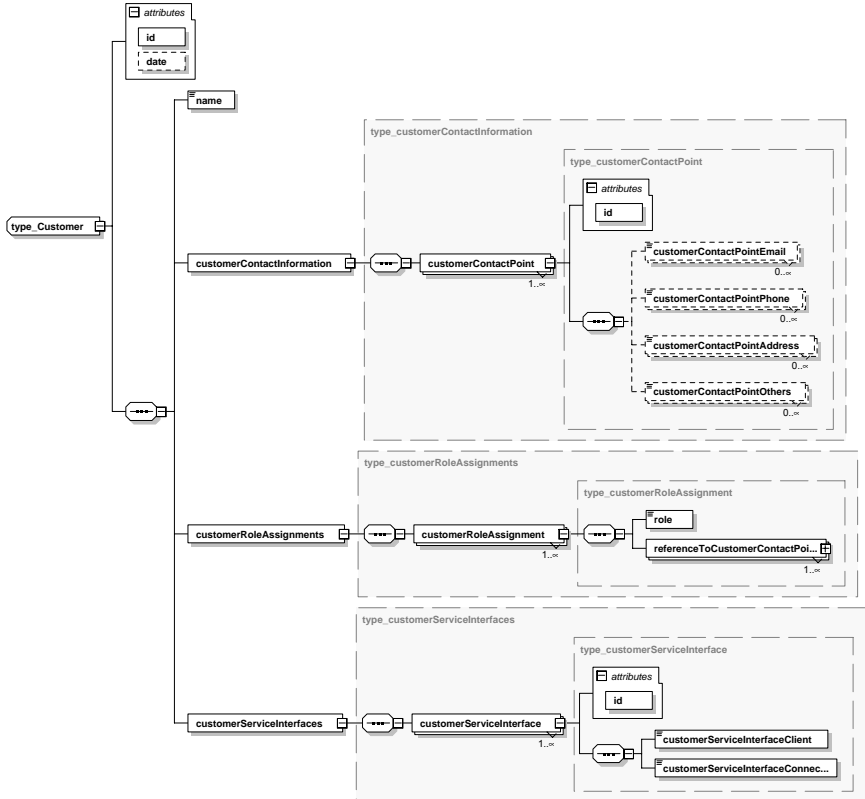
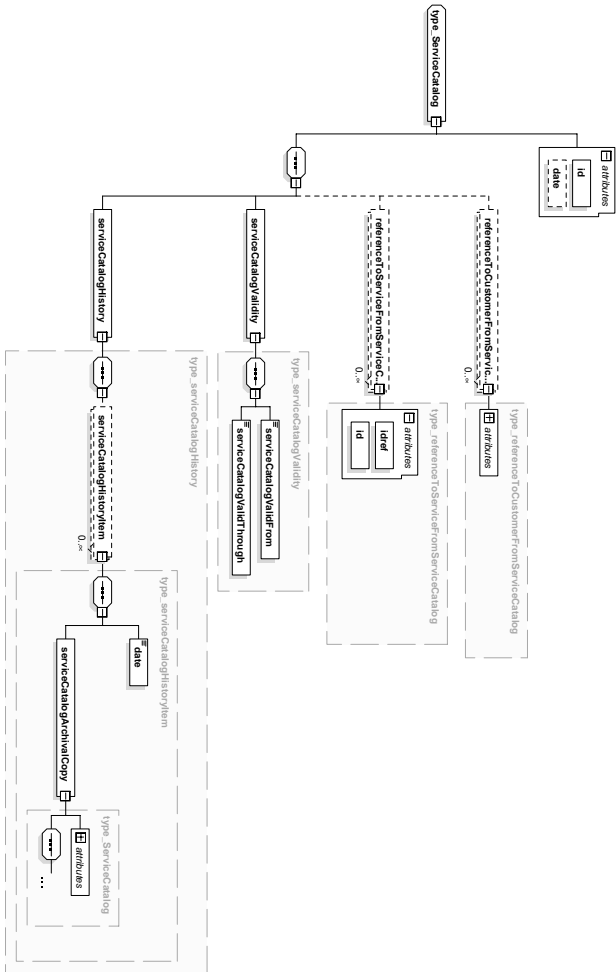


Abbildung 6.5: Visualisierung des Datenmodells zum Informationsartefakt Customer (Auszug)

Abbildung 6.6: Visualisierung der Typdefinition zum Informationsartefakt ServiceCatalog (Auszug)



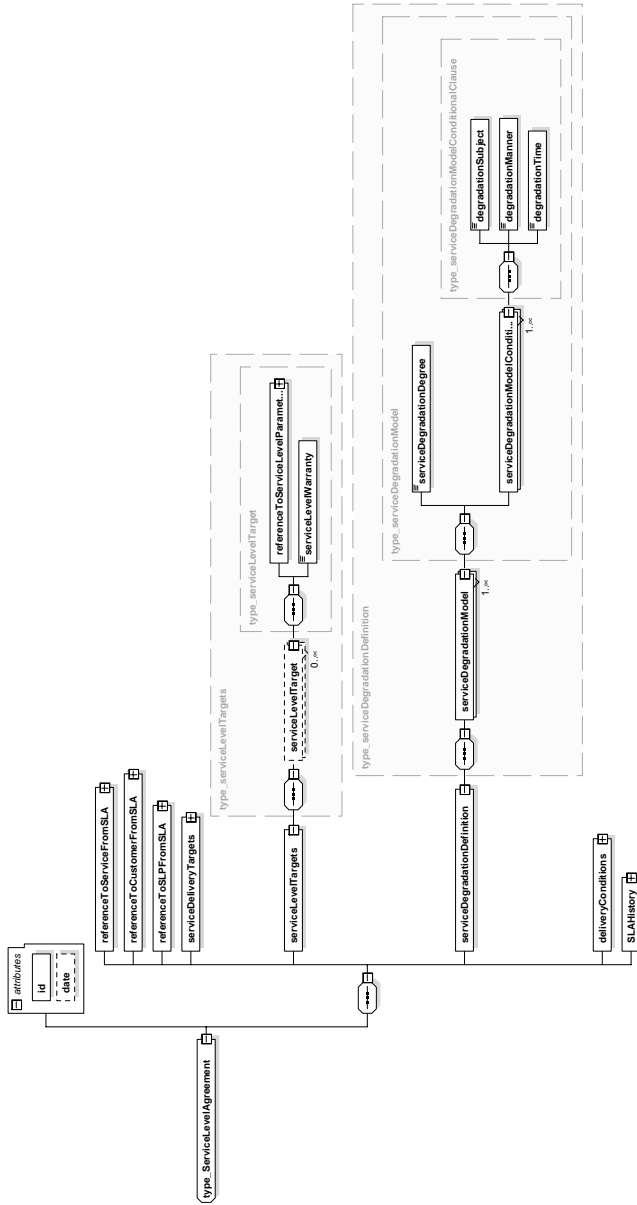
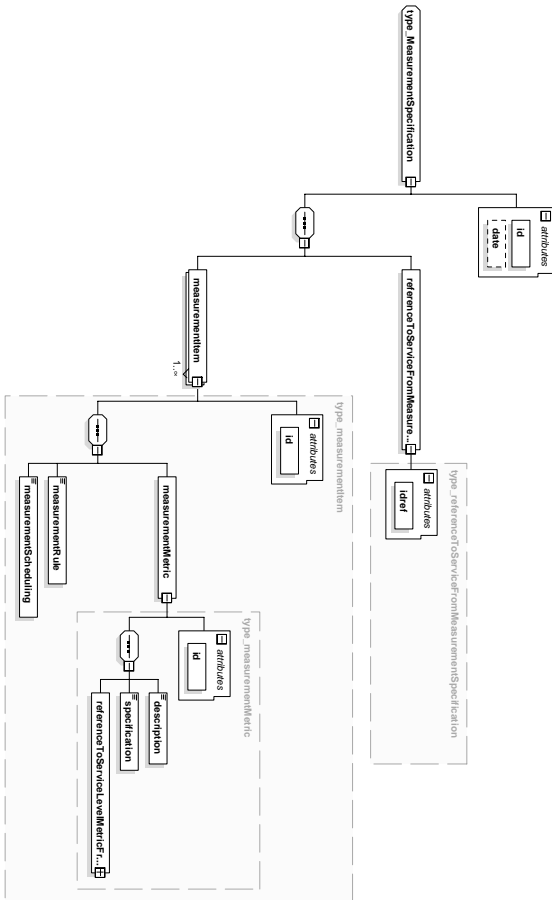


Abbildung 6.7: Visualisierung der Typdefinition zum Informationsartefakt ServiceLevelAgreement (Auszug)

Abbildung 6.8: Visualisierung der Typdefinition zum Informationsartefakt MeasurementSpecification (Auszug)



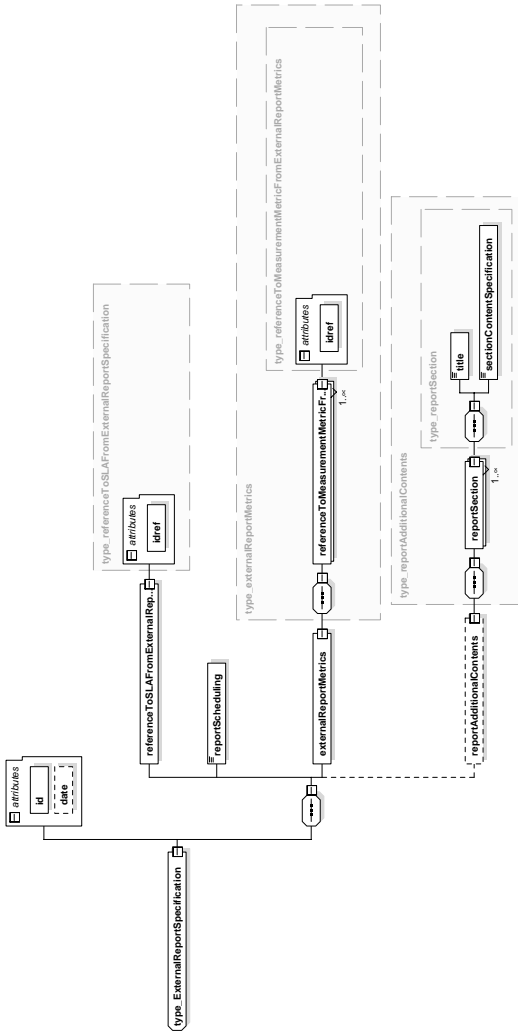


Abbildung 6.9: Visualisierung der Typdefinition zum Informationsartefakt ExternalReportSpecification (Auszug)

Artefakt-Typ	Artefaktklasse	Verweis
Datenbank	ServicePortfolio	5.2.4.1, A.3
	CustomerDatabase	5.2.4.2, A.4
	SDPDatabase	5.2.4.3, A.5
	SLPDatabase	5.2.4.4, A.6
	SLADatabase	5.2.4.5, A.7
	SADatabase	5.2.4.6, A.8
	MeasurementDatabase	5.2.4.7, A.9
	ReportDatabase	5.2.4.8, A.10
	ConflictDatabase	5.2.4.10, A.11
Datenbank-Sicht	ServiceCatalog	5.2.4.1, A.3
Datensatz	Service	5.2.4.1, A.3
	Customer	5.2.4.2, A.4
	ServiceDeliveryParty (SDP)	5.2.4.3, A.5
	SupportLevelProfile (SLP)	5.2.4.4, A.6
	ServiceMeasurementSpecification	5.2.4.7, A.9
	ServiceMeasurementResults	5.2.4.7, A.9
	InternalReportSpecification	5.2.4.8, A.10
	ExternalReportSpecification	5.2.4.8, A.10
	AgreementConflict	5.2.4.10, A.11
Dokumentation	ServiceLevelAgreement (SLA)	5.2.4.5, A.7
	OperationalLevelAgreement (OLA)	5.2.4.6, A.8
	UnderpinningContract (UC)	5.2.4.6, A.8
Bericht	InternalReport	5.2.4.8, A.10
	ExternalReport	5.2.4.8, A.10
	SLAViolationNotification	5.2.4.9, A.10

Tabelle 6.2: Überblick über alle SLM-spezifischen Informationsartefakte

6.2.4 Zusammenfassung

Das Informationsmodell der System-Sicht beschreibt einen Ansatz zur Realisierung eines umfassenden Informationsmanagementsystems für Service-Level-Management. Die in diesem Zusammenhang in der Modellierungssprache *XML Schema* entwickelten Module spezifizieren dabei verfeinerte Datenmodelle für alle Informationsartefakte und Informationsklassen, die im Rahmen des Informationsmodells der Prozess-Sicht identifiziert und beschrieben wurden. Daneben wurden alle Beziehungen zwischen Artefakten durch die Einführung

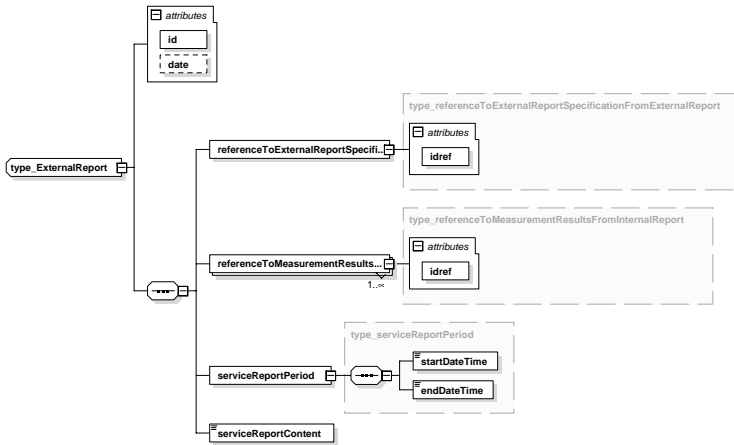


Abbildung 6.10: Visualisierung der Typdefinition zum Informationsartefakt ExternalReport (Auszug)

künstlicher Schlüsselattribute, die aus anderen Objekten referenziert werden können, berücksichtigt.

Das Ergebnis ist ein konsistentes Informations- und Datenmodell für Service-Level-Management, welches aufgrund der Modellierung in XML-Notation in Computer-verarbeitbarer Form vorliegt.

6.3 Funktionsmodell

Mit dem verfeinerten Informationsmodell der System-Sicht und dem Funktionsmodell der Prozess-Sicht im Hintergrund können nun Aspekte einer teilweisen funktionalen Automatisierung im Service-Level-Management adressiert werden.

Zur Erinnerung: Das Funktionsmodell der Prozess-Sicht umfasst

- Festlegungen von Funktionsbereichen und
- Definitionen von Referenz-Workflows für die Primäranwendungsfälle zur Darstellung der Verantwortungsverteilung sowie der Kontroll- und Informationsflüsse.

Darauf aufbauend behandelt dieses Teilkapitel schwerpunktmäßig Automatisierungsaspekte. Denn wie schon auf der Prozess-Sicht vereinzelt angesprochen wurde, wird von einem IT-gestützten Managementsystem für Service-Level-Management deutlich mehr erwartet als eine reine Workflow-Steuerung. Vielmehr soll auch die computergestützte Ausführung von Aktivitäten, wo immer dies möglich und sinnvoll erscheint, realisiert werden.

Fragestellungen dieses Moduls: Das Funktionsmodell der System-Sicht soll vor dem Hintergrund dieser Zielsetzung Antworten auf die folgenden Fragen geben:

- Welcher Grad an Automatisierung kann im Service-Level-Management potenziell erreicht werden, und welchen Beitrag liefert diese Managementarchitektur, um diesen Automatisierungsgrad tatsächlich zu erreichen?
- Welches Potenzial zur teilweisen oder vollständigen Automatisierung liefern welche (Teil-)Workflows und Aktivitäten?
- Welche Erweiterungen und Verfeinerungen ergeben sich für automatisierbare Anteile von Workflows als Beitrag zur Spezifikation eines IT-gestützten Managementsystems?

6.3.1 Methodischer Ansatz

Eine fundierte Analyse des funktionalen Automatisierungspotenzials im Service-Level-Management sowie des Potenzials zur Automatisierbarkeit einzelner Teil-Workflows der identifizierten Anwendungsfälle ist essenziell für einen konkreten Systementwurf. Der für diese Analyse gewählte Ansatz wird in diesem Abschnitt beschrieben. Dabei wird zunächst ein geeignetes Stufenschema beschrieben, das später verwendet werden kann, um den (potenziellen) Automatisierungsgrad bestimmter Funktionsbereiche zu ermitteln.

6.3.1.1 Stufenschema zur Festlegung des Automatisierungsgrads

Allen Automatisierungsbemühungen vorgelagert stellt sich die Frage, welcher *Grad* und welche *Art der Automatisierung* im Service-Level-Management überhaupt möglich sind. Brenner [9] hat in ähnlichem Zusammenhang die bereits in Kapitel 4 vorgestellte Taxonomie entwickelt, die versucht, Teildisziplinen bzw. Prozesse des Dienstmanagements anhand definierter Kriterien zu bewerten und schließlich zu kategorisieren. Das Ergebnis ist keine absolute, sondern eine relative Einordnung von Dienstmanagementprozessen, die vor allem ein Ziel erfüllt: die Bildung von Gruppen (*Clustern*) von Managementprozessen, die im Hinblick auf ihre Anforderungen an eine geeignete IT-Unterstützung Ähnlichkeiten aufweisen.

Das nachfolgend beschriebene Stufenschema dient hingegen nicht der Abgrenzung von Service-Level-Management gegen andere Dienstmanagementdisziplinen, sondern der Bewertung und Einordnung der verschiedenen Funktions(teil)bereiche innerhalb des Service-Level-Managements (vgl. Funktionsmodell der Prozess-Sicht) im Hinblick auf ihr Automatisierungspotenzial. Folgende Stufen werden unterschieden:

IT-gestützte Informationsverwaltung (*Information Management*)

Die schwächste Form der Automatisierung (*IT Support*) liegt vor, wenn lediglich die im Rahmen der Managementprozesse zu verarbeitenden Daten und Informationen rechnergestützt verwaltet werden. Der Einsatz von Versionierungswerkzeugen, Datenbanken oder speziellen Dokumentenmanagement-Werkzeugen realisiert diesen Automatisierungsgrad auch ohne das Vorhandensein eines integrierten, IT-gestützten Managementsystems mit entsprechender Managementplattform. Eine IT-gestützte Informationsverwaltung stellt allerdings gleichzeitig eine elementare Grundlage für ein IT-gestütztes Managementsystem dar (vgl. Kapitel 6.2.3.1 – Spezifikation eines SLM-Informationssystem im

Rahmen des Informationsmodells der System-Sicht, sowie Kapitel 7.1.2 – Schritt 1 des Aufbaus einer Managementplattform).

Kooperationsunterstützung (*Cooperation Support*) Wird zusätzlich zur Informationsverwaltung der gegenseitige Austausch von Informationen zwischen beteiligten Akteuren auf der Basis eines gemeinsamen Informationsmodells und über wohldefinierte Schnittstellen realisiert, kann bereits von Kooperationsunterstützung gesprochen werden. Brenner spricht in diesem Zusammenhang auch von *Workgroup-Computing-Werkzeugen*.

Workflow-Steuerung (*Workflow Control*) Eine weitere Erhöhung des Automatisierungsgrades wird erreicht, wenn nicht nur Kooperation über wohldefinierte Schnittstellen unterstützt wird, sondern der Gesamtprozess in geeignete Workflows zerlegt wurde, deren Einhaltung durch ein entsprechendes Plattform-Modul (*Workflow-Management-System*) forciert wird.

Entscheidungs-/Ausführungsunterstützung (*Decision/Execution Support*) Innerhalb von Workflows müssen Entscheidungen getroffen und Aktivitäten ausgeführt werden. In vielen Fällen müssen zu diesem Zweck Informationen ermittelt, ausgewertet und – im Falle einer Aktivität – zu einem neuen oder veränderten Informationsartefakt verarbeitet werden (*information processing*). Eine Workflow-Steuerung allein leistet hier noch keine Unterstützung. Wird ein Systembenutzer bei der Entscheidungsfindung und Aktivitätsausführung unterstützt, indem die erforderliche Informationsverarbeitung automatisch durchgeführt wird, dann ist der Automatisierungsgrad *Decision/Execution Support* erreicht.

(Voll-)Automatisierung (*Automation*) Schließlich wird der höchste Grad an IT-Unterstützung durch Vollautomatisierung von Entscheidungsfindung und Aktivitätsausführung erreicht.

Bemerkung: In seinem Buch *Humans and Automation* [93] definiert der Autor Thomas B. Sheridan drei grundsätzlich mögliche Ausprägungen von Automatisierung wie folgt:

1. Mechanisierung und Integration der Abtastung/Erkennung von Umweltparametern durch Sensoren
2. Datenverarbeitung und Entscheidungsfindung durch Computer

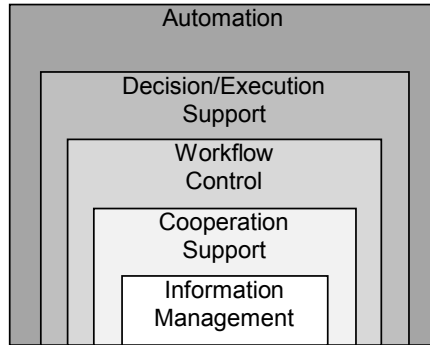


Abbildung 6.11: Stufenschema zur Festlegung des Automatisierungsgrads

3. Mechanische Aktivitäten durch motorisierte Geräte

Die zweite Ausprägung entspricht dem hier zugrunde gelegten Verständnis von Automatisierung (*Automation*).

Abbildung 6.11 stellt dieses Stufenschema grafisch dar und illustriert dabei einen weiteren wichtigen Aspekt: die Annahme, dass jede Stufe die darunter liegenden Stufen umschließt. Wie kann diese Annahme gerechtfertigt werden? Technisch gesehen wäre es ohne weiteres möglich, eine Workflow-Steuerung zu implementieren, ohne dass zuvor eine IT-gestützte Informationsverwaltung realisiert worden sein muss. Speziell im Hinblick auf die Disziplin Service-Level-Management macht das allerdings so gut wie keinen Sinn. Die Anforderungsanalyse hat deutlich gezeigt, welchen hohen Stellenwert Informationen in Form von Dokumenten, Datensätzen und Berichten im Service-Level-Management einnehmen. Workflows werden erst dann interessant, wenn diese Informationen definiert und verfügbar sind (*Information Management*), sich alle Akteure innerhalb des SLM-Prozesses auf ein gemeinsames Informationsmodell geeinigt haben (*Cooperation Support*) und nun Wege gesucht werden, strukturiert und effektiv mit den Managementaufgaben rund um diese Informationen umzugehen (*Workflow Control*). Insofern kann das Stufenmodell in gewisser Weise auch als ein Reifegradmodell im Hinblick auf Automatisierung von Service-Level-Management aufgefasst werden.

6.3.1.2 Methodik zur Identifikation des Automatisierungspotenzials

Legt man das gerade eingeführte Stufenschema zugrunde, so lässt sich mit den bisherigen Bestandteilen der Managementarchitektur der Automatisierungsgrad *Workflow Control* erreichen, sofern die Vorgaben der Prozess-Sicht sowie insbesondere des Informationsmodells der System-Sicht beim Plattformentwurf umgesetzt werden.

Daher beschäftigt sich die in diesem Abschnitt vorgestellte *Methodik zur Identifikation des Automatisierungspotenzials* speziell mit den Stufen 4 und 5 mit dem Ziel, Workflow- und Aktivitäts-spezifisch festzustellen, für welche Teile von Workflows, Aktivitäten und Entscheidungen eine automatisierte Entscheidungs- und Ausführungsunterstützung realisierbar ist, und welche sogar das Potenzial zur Vollautomatisierung bieten. Ausgangspunkt sind alle Referenz-Workflows aus dem Funktionsmodell der Prozess-Sicht. Diese liefern bereits eine ebenso umfassende wie detaillierte Sicht auf das Funktionsspektrum eines (IT-gestützten) Managementsystems für Service-Level-Management. Jede Aktivität und jede Entscheidungsraute eines Workflows muss untersucht und nach festgelegten Kriterien einer der folgenden Klassen zugeordnet werden:

- Nicht automatisierbar → nur Workflow-Steuerung: Die Aktivität bzw. Entscheidungsfindung ist nicht automatisierbar, da mindestens ein Kriterium für Automatisierbarkeit (siehe nächster Teilabschnitt) nicht erfüllbar ist. Sie ist zudem auch nicht weiter in automatisierbare und nicht automatisierbare Anteile zerlegbar.
- Teilweise automatisierbar → Workflow-Steuerung mit Entscheidungs-/Ausführungsunterstützung: Die Aktivität bzw. Entscheidungsfindung kann zerlegt werden in vollständig automatisierbare und nicht automatisierbare Anteile (Sub-Workflow, Sub-Aktivitäten).
- Vollständig automatisierbar → Vollautomatisierung: Die Durchführung der Aktivität bzw. die Entscheidungsfindung kann vollständig automatisiert abgearbeitet werden und liefert als Ergebnis alle für nachgelagerte Aktivitäten oder den Abschluss eines Workflows erforderlichen Ausgaben (Outputs) bzw. im Falle einer Entscheidungsfindung ein eindeutiges Ergebnis.

Wird eine einzelne Aktivität oder Entscheidungsraute nach genauer Analyse als teilweise automatisierbar bewertet, so ist an dieser Stelle also eine Verfeinerung des Referenz-Workflows erforderlich. Ein Sub-Workflow muss definiert werden, auf den die Analyse-Methodik rekursiv angewendet wird. Die

„Men Are Better At ...“	„Machines Are Better At ...“
Improvisation, Anwendung flexibler Verfahren, logisches Denken, induktive Beweisführung, Beurteilung/-Urteilsvermögen	Schnelle Reaktion auf Steueranweisungen/Signale, präzise Ausführung von Anweisungen, Speichern und Löschen von Informationen, Deduktion

Tabelle 6.3: Auszug aus der MABA-MABA-Liste nach Fitts [5]

Methodik geht davon aus, dass durch schrittweise Verfeinerung und rekursive Anwendung nach endlich vielen Iterationen ein Workflow entsteht, der ausschließlich aus nicht automatisierbaren und vollständig automatisierbaren Aktivitäten besteht.

6.3.1.3 Analysekriterien

Nach welchen Kriterien kann überhaupt entschieden werden, ob eine Aktivität automatisierbar oder nicht automatisierbar ist? Der Schlüssel zur Beantwortung dieser Frage liegt offenbar in einem grundlegenden Verständnis darüber, was Maschinen und Programme grundsätzlich in der Lage sind zu leisten und was nicht. Bereits in den frühen 1950er Jahren veröffentlichte Fitts [5] eine Gegenüberstellung der Eigenschaften und Stärken von Menschen im Vergleich zu denen von Maschinen. Aufgrund der englischen Bezeichnungen „Men Are Better At“ und „Machines Are Better At“ wird diese Gegenüberstellung auch als *MABA-MABA-Liste* bezeichnet. Tabelle 6.3 zeigt einen Auszug aus dieser Liste, die freilich aus einer Zeit stammt, in der sich Automatisierung im Wesentlichen auf industrielle Produktions- und Fertigungsprozesse bezog.

Fitts Erkenntnisse erscheinen aus heutiger Sicht weitgehend trivial, wenngleich sie bis heute Gültigkeit besitzen. So bilden sie einen guten Ausgangspunkt für die Definition der Analysekriterien zur Untersuchung der Referenz-Workflows auf Automatisierbarkeit.

Ausschlusskriterien gegen vollständige Automatisierbarkeit Ausgehend von der Fitts-Liste lassen sich speziell für die Automatisierbarkeitsanalyse von Aktivitäten und Entscheidungen in Workflows folgende Ausschlusskriterien für die Automatisierbarkeit festlegen. Muss mindestens eine der folgende Fragen bei der Untersuchung einer konkreten Aktivität mit *ja* beantwortet werden, so spricht dies gegen eine vollständige Automatisierung:

- Erfordert die Aktivität die *Anwendung flexibler Verfahren* wie zum Beispiel Verhandlungen, Umgang mit unbekanntem Situationen?
- Erfordert die Aktivität *logisches Denken*?
- Erfordert die Aktivität eine *Beurteilung* eines bestehenden oder geplanten Zustands oder einer Situation unter Berücksichtigung von Wissen, Erfahrung und Kontext?
- Erfordert die Aktivität eine *Entscheidung unter Informationsdefizit*?

Die Beantwortung aller dieser Fragen mit *nein* stellt eine notwendige, aber noch keine hinreichende Bedingung für die Automatisierbarkeit einer Aktivität dar.

Kriterien für vollständige Automatisierbarkeit Wurden die Ausschlusskriterien erfolgreich passiert, alle Fragen also mit *nein* beantwortet, können die folgenden Kriterien das grundsätzlich vorhandene Potenzial zur Automatisierung entweder untermauern bzw. die Nichterfüllung eine genauere Abwägung erfordern:

- Standardisierte Eingaben (*Inputs*)
- Deterministisches Ablauf- und Entscheidungsverhalten anhand vordefinierter Prozeduren und Entscheidungsbäume
- Keine menschliche Intervention/Steuerung erforderlich
- Standardisierte Ausgaben (*Outputs*)

6.3.2 Automatisierung von Managementfunktionen

Das vorgestellte Stufenschema, die Analysemethodik und die Analyse Kriterien sind nur Mittel zum Zweck. Nun geht es darum, diese Instrumente in einer Weise anzuwenden, dass die Managementarchitektur im Ergebnis um weitere Modellelemente, Verfeinerungen und Systementwurfsrichtlinien erweitert werden kann, die für die Entwicklung eines IT-gestützten Managementsystems für Service-Level-Management hilfreich sind.

Ausgehend vom Funktionsmodell der Prozess-Sicht werden im Abschnitt 6.3.2.1 alle funktionalen Teilbereiche und die enthaltenen Referenz-Workflows

einer Analyse nach der vorgestellten Methodik unterzogen, um automatisierbare und teil-automatisierbare Aktivitäten zu identifizieren. Im Anschluss daran wird im Abschnitt 6.3.2.2 für eine Auswahl dieser Aktivitäten exemplarisch gezeigt, wie das Automatisierungspotenzial im konkreten Fall ausgeschöpft werden kann. Hierzu wird das Funktionsmodell an den entsprechenden Stellen um neue Komponenten erweitert.

6.3.2.1 Workflow-bezogene Automatisierbarkeitsanalyse

Tabelle 6.4 zeigt das Ergebnis der Automatisierbarkeitsanalyse über alle Funktionsbereiche und die enthaltenen Anwendungsfälle, ihre Referenz-Workflows und Aktivitäten (UC x verweist auf *Use Case x*).

<i>Workflow</i>	<i>Aktivität/Entscheidung</i>	<i>nicht automatisierbar</i>	<i>teilweise automatisierbar</i>	<i>voll automatisierbar</i>
1: <i>Create new service</i>	Aktivität: <i>Identify key functional service requirements</i>	×		
	Aktivität: <i>Assess feasibility</i>	×		
	Entscheidung: <i>Feasible?</i>	×		
	Aktivität: <i>Formal requirements engineering</i>	×		
	Aktivität: <i>Insert service into portfolio</i>			✓
2: <i>Change service</i>	Aktivität: <i>Define change request</i>	×		
	Entscheidung: <i>SLAs impacted?</i>			✓
	Workflow: <i>Revise SLA</i>	s. UC 20		
	Aktivität: <i>Update service portfolio</i>			✓
3: <i>Delete service</i>	Entscheidung: <i>Service in service catalogs?</i>			✓

	Workflow: <i>Remove service from service catalog</i>	s. UC 12		
	Aktivität: <i>Remove service from portfolio</i>			✓
4: <i>Create new customer</i>	Aktivität: <i>Create customer</i>		(✓)	
5: <i>Change customer</i>	Aktivität: <i>Update customer</i>	×		
6: <i>Delete customer</i>	Entscheidung: <i>Service catalog(s) assigned to customer?</i>			✓
	Workflow: <i>Dissociate catalog from customer</i>	s. UC 9		
	Aktivität: <i>Delete customer</i>			✓
7: <i>Initialize service catalog</i>	Aktivität: <i>Create service catalog</i>		(✓)	
8: <i>Assign catalog to customer</i>	Aktivität: <i>Confirm assignment</i>	×		
9: <i>Dissociate catalog from customer</i>	Entscheidung: <i>SLAs referencing service catalog?</i>			✓
	Workflow: <i>Resign SLA</i>	s. UC 21		
	Aktivität: <i>Confirm dissociation</i>	×		
10: <i>Delete service catalog</i>	Entscheidung: <i>Service catalog assigned to customer?</i>			✓
	Workflow: <i>Dissociate catalog from customer</i>	s. UC 9		
	Aktivität: <i>Delete service catalog</i>			✓
11: <i>Add service to catalog</i>	Aktivität: <i>Link service with catalog</i>			✓
12: <i>Remove service from catalog</i>	Entscheidung: <i>Active SLAs referencing service in this catalog?</i>			✓
	Workflow: <i>Resign SLA</i>	s. UC 21		
	Aktivität: <i>Unlink service from catalog</i>			✓
13: <i>Create support level profile</i>	Aktivität: <i>Define service support requirements</i>	×		
	Aktivität: <i>Confirm support level profile</i>	×		

14: <i>Change support level profile</i>	Aktivität: <i>Define change request</i>	×		
	Aktivität: <i>Confirm support level profile</i>	×		
	Entscheidung: <i>All SLAs updated?</i>			✓
	Workflow: <i>Revise SLA</i>	s. UC 20		
15: <i>Delete support level profile</i>	Entscheidung: <i>Support level profile applied in SLAs?</i>			✓
	Workflow: <i>Resign SLA</i>	s. UC 21		
	Aktivität: <i>Delete support level profile</i>			✓
16: <i>Create delivery party</i>	Entscheidung: <i>Type of delivery party?</i>	×		
	Aktivität: <i>Create internal service delivery party</i>		(✓)	
	Aktivität: <i>Create supplier</i>		(✓)	
17: <i>Change delivery party data</i>	Aktivität: <i>Update delivery party</i>	×		
18: <i>Delete delivery party</i>	Entscheidung: <i>Type of delivery party?</i>			✓
	Entscheidung: <i>Active Operational Level Agreements with internal delivery party?</i>			✓
	Workflow: <i>Resign OLA</i>	s. UC 24		
	Aktivität: <i>Delete internal service delivery party</i>			✓
	Entscheidung: <i>Active Underpinning Contracts with external supplier?</i>			✓
	Workflow: <i>Resign UC</i>	s. UC 27		
	Aktivität: <i>Delete supplier</i>			✓
19: <i>Close SLA</i>	Aktivität: <i>Select service from catalog</i>	×		
	Aktivität: <i>Initialize new SLA</i>			✓
	Aktivität: <i>Negotiate contents</i>		(✓)	
	Entscheidung: <i>All targets substantiated?</i>		(✓)	
	Workflow: <i>Close OLA</i>	s. UC 22		

	Workflow: <i>Close UC</i>		s. UC 25	
	Aktivität: <i>Confirm SLA</i>	×		
20: <i>Revise SLA</i>	Aktivität: <i>Specify requested changes</i>	×		
	Aktivität: <i>Negotiate changes</i>		(√)	
	Entscheidung: <i>All targets substantiated?</i>		(√)	
	Workflow: <i>Revise OLA</i>		s. UC 23	
	Workflow: <i>Revise UC</i>		s. UC 26	
	Aktivität: <i>Confirm changes</i>	×		
21: <i>Resign SLA</i>	Aktivität: <i>Confirm cancellation</i>	×		
22: <i>Close OLA</i>	Aktivität: <i>Initialize new OLA</i>			√
	Aktivität: <i>Negotiate contents</i>	×		
	Aktivität: <i>Confirm OLA</i>	×		
23: <i>Revise OLA</i>	Aktivität: <i>Specify requested changes</i>	×		
	Aktivität: <i>Negotiate changes</i>	×		
	Aktivität: <i>Confirm changes</i>	×		
24: <i>Resign OLA</i>	Aktivität: <i>Confirm OLA cancellation</i>	×		
25: <i>Close UC</i>	Aktivität: <i>Initialize new UC</i>			√
	Aktivität: <i>Negotiate contents</i>	×		
	Aktivität: <i>Confirm UC</i>	×		
26: <i>Revise UC</i>	Aktivität: <i>Specify requested changes</i>	×		
	Aktivität: <i>Negotiate changes</i>	×		
	Aktivität: <i>Confirm changes</i>	×		
27: <i>Resign OLA</i>	Aktivität: <i>Confirm UC cancellation</i>	×		
28: <i>Analyze agreements</i>	Aktivität: <i>Select analysis mode</i>	×		
	Aktivität: <i>Perform conflict analysis</i>			√
29: <i>Measure and record</i>	Aktivität: <i>Define measurement specification</i>		(√)	
	Workflow: <i>Generate SLA violation notification</i>		s. UC 32	

30: <i>Create internal report</i>	Aktivität: <i>Define internal report specification</i>		(√)	
	Aktivität: <i>Generate internal report</i>		(√)	
31: <i>Create external report</i>	Aktivität: <i>Define external report specification</i>		(√)	
	Aktivität: <i>Generate external report</i>		(√)	
	Aktivität: <i>Confirm and deliver external report</i>	×		
32: <i>Generate SLA violation notification</i>	Aktivität: <i>Generate SLA violation notification</i>			√

Tabelle 6.4: Automatisierbarkeitsanalyse aller Funktionsbereiche

6.3.2.2 Realisierung

Ziel dieses Abschnitts ist es, für einige ausgewählte Aktivitäten aus der Menge derer, die als teilweise oder vollständig automatisierbar klassifiziert wurden, darzustellen, *wie* das Automatisierungspotenzial ausgeschöpft werden kann.

Kopplung des Dienstportfolios an das Change und Release Management Die Aktivitäten *Insert service into portfolio* und *Update service portfolio* kommen im Kontext der Anwendungsfälle 1 und 2 zur Ausführung, nachdem ein neuer oder geänderter Dienst durch das *Release Management* freigegeben wurde. Die Aktivität *Remove service from portfolio* des Anwendungsfalls 3 kommt zur Ausführung, wenn ein Dienst aus dem Dienstportfolio entfernt wurde und somit stillgelegt werden kann. Alle diese Aktivitäten haben also festgelegte Auslöser (*Trigger*) und führen dann Operationen auf dem SLM-Informationssystem aus. Eine Kopplung des Managementsystems für SLM an die Prozesse *Change Management* und *Release Management* bzw. ihre unterstützenden Plattformen ermöglicht eine Vollautomatisierung der genannten Aktivitäten. Daraus ergibt sich die nachfolgende Systementwurfsrichtlinie mit spezifischem Bezug zu den genannten drei Anwendungsfällen.

Systementwurfsrichtlinie 9: Ein IT-gestütztes Managementsystem für Service-Level-Management sollte ausgerollte Dienste automatisch im Dienstportfolio aktivieren und für Dienste, die endgültig aus dem Dienstportfolio entfernt wurden,

automatisch *Requests for Changes* zur Stilllegung der Dienste generieren. Hierzu sind gemäß Kommunikationsmodell der Architektur geeignete Schnittstellen zu schaffen.

Automatisierte Entscheidungsfindung Zahlreiche Entscheidungen, wie zum Beispiel *SLAs impacted?* und *Service in service catalogs?* aus den Workflows zu den Anwendungsfällen 2 und 3, lassen sich mit Hilfe einer entsprechenden Datenbankabfrage auf dem SLM-Informationssystem leicht realisieren. Im ersten Fall werden alle Dienstvereinbarungen abgefragt, die den zu ändernden Dienst referenzieren, im zweiten Fall alle Dienstkataloge, die einen Verweis auf den Dienst enthalten. Ist die Rückgabemenge nicht leer, so liefert die Entscheidungsraute *SLA(s) impacted* bzw. *Service contained in one or more service catalog(s)* zurück (vgl. Kapitel 5.3.2.1, Abbildung 5.23 bzw. 5.24), bei leerer Rückgabemenge entsprechend das negierte Entscheidungsergebnis.

Die nachfolgende allgemeine Systementwurfsrichtlinie gilt nicht nur für die Entscheidungsrauten in den Workflows 2 und 3, sondern lässt sich auf alle anderen Entscheidungsprozesse übertragen, die auf der Analyse der Zusammenhänge zwischen Instanzen bestimmter Artefaktklassen beruhen.

Systementwurfsrichtlinie 10: Entscheidungen, die auf Basis referenzieller Abhängigkeiten innerhalb des Informationssystems zu treffen sind, sollten durch automatische Auswertung entsprechender Datenbankabfragen durch das Managementsystem automatisiert werden.

Teilautomatisierte Instanziierung von Informationsartefakten Die Aktivität *Create service catalog* des Workflows zum Anwendungsfall 7 kann in automatisierbare und nicht-automatisierbare Subaktivitäten zerlegt werden, wie in Abbildung 6.12 dargestellt. Dunkelgrau eingefärbte Subaktivitäten stellen die automatisierbaren Anteile dar. Darunter fallen in diesem Beispiel die Generierung des eigentlichen Objekts – beispielsweise durch ein Skript, basierend auf einem SQL `CREATE VIEW ...`-Befehl (da es sich bei diesem Informationsartefakt um eine Datenbank-Sicht handelt), welcher auf der entsprechenden Datenbankkomponente des Informationssystems ausgeführt wird – sowie die Vergabe einer eindeutigen ID und das Setzen des aktuellen Datums-/Zeitstempels.

Nicht automatisierbar und daher hellgrau eingefärbt ist die Subaktivität *Configure service catalog validity*, im Rahmen derer die Einträge

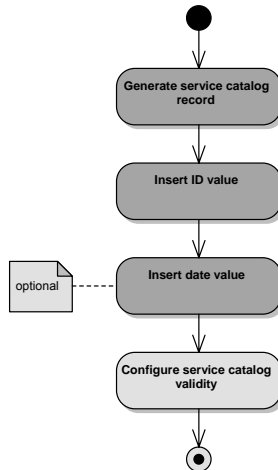


Abbildung 6.12: Sub-Workflow für die Aktivität *Create service catalog*

`serviceCatalogValidFrom` und `serviceCatalogValidThrough` gemäß Datenmodell manuell vorgenommen werden müssen.

Aus dieser fallbezogenen Betrachtung ergibt sich eine weitere allgemeine Systementwurfsrichtlinie:

Systementwurfsrichtlinie 11: Bei der Implementierung von Funktionen zur Instanziierung von Informationsartefakten sollten alle automatisch eindeutig ermittelbaren Felder des Datenmodells identifiziert werden und sichergestellt werden, dass sie systemseitig mit den korrekten Werten befüllt werden.

Automatische Initialisierung von Standard-Dienstvereinbarungen Die Aktivität *Initialize new SLA* des Anwendungsfalls 19 hat zum Ziel, in Vorbereitung auf die eigentliche Verhandlung (*Negotiate contents*) einen ersten Entwurf der Dienstvereinbarung zu generieren. Dies geschieht noch ohne Einbindung des Kundenvertreters. Je näher der Entwurf bereits an die spätere finale Dienstvereinbarung heranreicht und je besser die Kundenanforderungen im Vorfeld antizipiert werden, umso weniger Anpassungen sind in der Verhandlungsphase erforderlich und umso weniger Zeit wird für diese benötigt.

Wünschenswert wäre, wenn ein IT-gestütztes Managementsystem entsprechende Entwürfe von Dienstvereinbarungen automatisch generiert. Je häufiger neue Dienstvereinbarungen geschlossen werden, umso mehr gewinnt diese Option zur Automatisierung an Relevanz. Es ergibt sich folgende weitere Systementwurfsrichtlinie.

Systementwurfsrichtlinie 12: Auf Basis des ausgewählten Dienstes, für den eine Dienstvereinbarung geschlossen werden soll, sollte vom Managementsystem eine initiale *Standard-Dienstvereinbarung* für diesen Dienst generiert werden, die in der nachgelagerten Verhandlungsphase angepasst und detailliert werden kann. Sofern ein Multi-Level-SLA-Ansatz verfolgt wird, muss sich das Standard-SLA geeignet in die Hierarchie bestehender über- und untergeordneter Dienstvereinbarungen einfügen.

Eine mögliche systemtechnische Realisierung besteht darin, den Workflow zu Anwendungsfall 1 (*Create new service*) um die Erstellung einer Standard-Dienstvereinbarung für den neu erstellten Dienst zu erweitern. Die Standard-Dienstvereinbarung wird genau einmal instanziiert und dient dann als Muster/Vorlage (*Sample*) zur Verwendung im Rahmen von Anwendungsfall 19 (*Close SLA*). Das Anlegen eines *Dummy-Kunden*, mit dem alle Standard-Dienstvereinbarungen „abgeschlossen“ werden, ermöglicht die Realisierung ohne die Notwendigkeit einer Erweiterung des Informationsmodells.

IT-gestützte Verhandlung der SLA-Inhalte Die Verhandlung einer Dienstvereinbarung ist die zentrale Aktivität im Workflow zu Anwendungsfall 19. Da sie einen hohen Grad an Interaktion zwischen den Rollen Service-Level-Manager und Kundenvertreter erfordert (vgl. Kapitel 5.4.1.3 – Intraprozess-Kommunikationsmodell der Prozess-Sicht, Tabelle 5.15), bietet sie kaum echtes Automatisierungspotenzial. Lediglich eine eingeschränkte Entscheidungsunterstützung kann durch ein IT-gestütztes Managementsystem erreicht werden.

Systementwurfsrichtlinie 13: Ein Managementsystem sollte den verantwortlichen Service-Level-Manager während der Verhandlung einer Dienstvereinbarung durch relevante und aussagekräftige Informationen über den betreffenden Dienst und die Realisierbarkeit bestimmter Service-Levels in Abhängigkeit von Kosten und anderen Randbedingungen unterstützen.

6.3.3 Zusammenfassung

Das Funktionsmodell der System-Sicht greift die Referenz-Workflows der Prozess-Sicht auf und analysiert sie mit Blick auf das Ziel Automatisierung. Eine umfassende Automatisierbarkeitsanalyse liefert Anhaltspunkte dafür, in welchen funktionalen Teilbereichen des Service-Level-Managements ein Automatisierungspotenzial realisierbar ist, das über eine IT-gestützte Informationsverwaltung, Kooperationsunterstützung und Workflow-Steuerung hinausgeht.

Das Ergebnis dieses Moduls der Managementarchitektur besteht neben einem methodischen und somit wiederverwendbaren Ansatz für eine Automatisierbarkeitsanalyse und dessen Anwendung auf die Modelle der Prozess-Sicht aus einem weiteren Satz an Systementwurfsrichtlinien, die sich Anwendungsfall-spezifisch mit ausgewählten Automatisierungsaspekten beschäftigen. In diesem Kontext wurden Realisierungskonzepte für die Kopplung von Service-Level-Management an die Prozesse *Change Management* und *Release Management*, automatisierte Entscheidungsfindung, teilautomatisierte Instanziierung von Informationsartefakten, automatische Initialisierung von Standard-Dienstvereinbarungen und IT-gestützte Verhandlung von SLA-Inhalten entwickelt.

6.4 Kommunikationsmodell

Das Kommunikationsmodell der System-Sicht greift die differenzierte Betrachtung von Aspekten der Intraprozesskommunikation und der Interprozesskommunikation wieder auf und setzt sie im Hinblick auf systementwurfsrelevante und systemarchitekturelle Fragestellungen fort.

Zur Erinnerung: Das Kommunikationsmodell der Prozess-Sicht umfasst

- ein Intraprozess-Kommunikationsmodell zur Darstellung der Interaktionen zwischen SLM-spezifischen Rollen auf Basis der Referenz-Workflows und
- ein Interprozess-Kommunikationsmodell zur Einordnung von Service-Level-Management in den Kontext eines umfassenden Dienstmanagements (basierend auf dem Prozess-Rahmenwerk von ISO/IEC 20000) und zur Darstellung von Zusammenhängen und Informationsflüssen zwischen Service-Level-Management und anderen Dienstmanagementprozessen.

Während also die konkreten *Anforderungen an den Informationsaustausch* zwischen verschiedenen Rollen innerhalb des Service-Level-Managements bzw. zwischen einem Managementsystem für Service-Level-Management und anderen Dienstmanagementsystemen im Rahmen des Kommunikationsmodells der Prozess-Sicht identifiziert und spezifiziert wurden, beschäftigt sich dieses Unterkapitel schwerpunktmäßig mit Möglichkeiten zur *IT-gestützten Realisierung dieser Anforderungen*.

Fragestellungen dieses Moduls: Die Fragen, die durch das Kommunikationsmodell der System-Sicht demnach beantwortet werden müssen, lauten ausgehend vom Kommunikationsmodell der Prozess-Sicht:

- Wie können die Intraprozess-Kommunikationsbeziehungen, die im Rahmen des Kommunikationsmodells der Prozess-Sicht identifiziert und klassifiziert wurden, durch ein IT-gestütztes Managementsystem realisiert und abgestützt werden?
- Wie können die erforderlichen Informationsflüsse zwischen Service-Level-Management und anderen Dienstmanagementdisziplinen unter dem Dach eines integriert-kooperativen Ansatzes im Dienstmanagement realisiert werden?

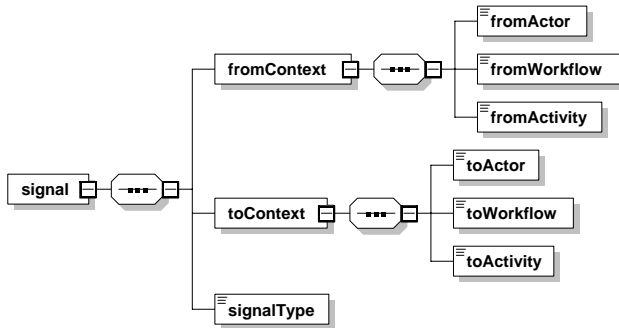


Abbildung 6.13: Schematische Darstellung für das Kommunikationsmuster *Signalisierung*

6.4.1 Realisierung der Intraprozesskommunikation

Maßgeblich für die Auseinandersetzung mit der Realisierung einer Kommunikationsbeziehung zwischen zwei Rollen durch ein IT-gestütztes Managementsystem ist die Charakteristik dieser Kommunikationsbeziehung, wie sie durch die Klassifizierung mit Hilfe der fünf in der Prozess-Sicht festgelegten Kommunikationsmuster beschrieben wird. Statt jede Kommunikationsbeziehung für sich genommen zu untersuchen, wird die Automatisierbarkeitsanalyse auf den vier relevanten Klassen von Kommunikationsmustern durchgeführt und liefert somit einen Satz generischer Ergebnisse, die sich schließlich auf die konkreten Kommunikationsbeziehungen übertragen lassen. Nachfolgend werden deshalb die Möglichkeiten und Grenzen der IT-Unterstützung der Kommunikationsmuster K_1 (Signalisierung), K_2 (Informationsübermittlung), K_3 (Anfrage-Antwort-Kommunikation) und K_4 (Standardisierter Dialog) evaluiert.

6.4.1.1 K_1 : Signalisierung

Signalisierung stellt die einfachste Form der Kommunikation zwischen Aktoren dar. Die technische Realisierung im Rahmen eines IT-gestützten Managementsystems ist simpel und in Abbildung 6.13 schematisch dargestellt. Wichtig bei der unidirektionalen Kommunikation über Signale ist der Kontext, aus dem heraus das Signal gesendet wird, der Kontext, in dem das Signal empfangen wird, und der Typ des Signals.

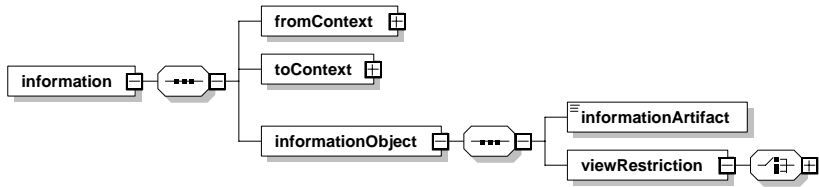


Abbildung 6.14: Schematische Darstellung für das Kommunikationsmuster *Informationsübermittlung*

6.4.1.2 K_2 : Informationsübermittlung

Dieses Kommunikationsmuster unterscheidet sich von der Signalisierung (K_1) im höheren Semantikgehalt der Nachrichten. Entsprechend muss der Fokus der technischen Realisierung aller Kommunikationsbeziehungen dieses Musters auf der Spezifikation der zu übermittelnden Information liegen. Das Funktionsmodell der Prozess-Sicht gibt diesbezüglich nur das Informationsartefakt, nicht aber die tatsächlich erforderlichen Informationskomponenten hieraus vor. Abbildung 6.14 illustriert, dass das Informationsobjekt, welches Gegenstand der Kommunikationsbeziehung ist, nicht zwangsläufig eine identische Kopie eines Informationsartefakts sein muss, sondern ein Teil oder eine Transformation desselben sein kann.

Als Beispiel sei das Informationsartefakt *Customer* genannt, welches mit Unterstützung des Kundenvertreters im Rahmen von Anwendungsfall 4 erstellt wird. Über die gespeicherten Daten wird der Kundenvertreter informiert, bevor der Workflow seinen Endzustand erreicht. Hierbei kann es sinnvoll sein, eine Auswahl an Informationen zu treffen und etwa die `customerServiceInterfaces` (vgl. Datenmodell) auszublenden.

6.4.1.3 K_3 : Anfrage-Antwort-Kommunikation

Während eine Aktorenkommunikation nach dem Muster K_2 immer auf einem Teil des Informationsmodells basiert, trifft diese einschränkende Annahme und Feststellung für das weitaus generischere Kommunikationsmuster K_3 nicht mehr zu, wenngleich sowohl Anfragen als auch Antworten natürlich Teile von Informationsartefakten oder Transformationen derselben enthalten können. Grundsätzlich aber muss für alle Vorkommen des Kommunikationsmusters K_3 (gemäß Kommunikationsmodell der Prozess-Sicht) genau spezifiziert werden,

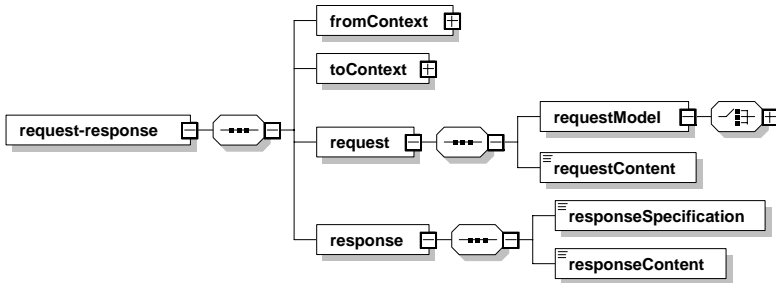


Abbildung 6.15: Schematische Darstellung für das Kommunikationsmuster *Anfrage-Antwort-Kommunikation*

- welche verschiedenen Anfragen aus dem Kontext des sendenden Aktors an den empfangenden Aktor gestellt werden können und
- welche Anforderungen die jeweils zugehörigen Antworten erfüllen müssen.

Nur so kann die Kommunikation durch ein IT-gestütztes Managementsystem effektiv unterstützt werden und sichergestellt werden, dass Anfragen zielgerichtet gestellt und vollständig beantwortet werden. Abbildung 6.15 illustriert, dass jede Anfrage (*request*) auf einem zuvor zu definierenden Anfragemodell (*requestModel*) basiert. In Abhängigkeit vom gewählten Anfragemodell werden Struktur und erforderliche Inhalte der Antwort spezifiziert (*responseSpecification*).

6.4.1.4 K_4 : Standardisierter Dialog

Abbildung 6.16 stellt schließlich das Schema für einen standardisierten Dialog dar. Wo immer eine Aktorenkommunikation nach diesem Muster stattfindet, beginnt diese mit einer initialen Anfrage, die verschiedene Antwortoptionen zulässt. In Abhängigkeit von der gewählten Option verläuft der weitere Dialog. Die Planung eines standardisierten Dialogs erfordert also eine sehr sorgfältige Ausarbeitung aller möglichen Abläufe. Daher sollte mit Blick auf die voraussichtliche Instanziierungshäufigkeit jeder Aktorenkommunikation nach diesem Muster eine Abwägung zwischen dem damit verbundenen Aufwand und dem resultierenden Nutzen getroffen werden.

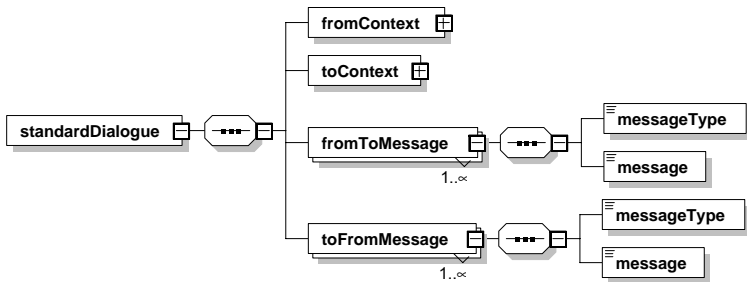


Abbildung 6.16: Schematische Darstellung für das Kommunikationsmuster *Standardisierter Dialog*

6.4.1.5 Zusammenfassung und Systementwurfsrichtlinien

Speziell im Hinblick auf die Kommunikationsmuster K_2 , K_3 und K_4 einer IT-gestützten Aktorenkommunikation muss im Rahmen des Systementwurfs sichergestellt werden, dass unter Berücksichtigung der Anfrage (im Falle von K_3) bzw. unter Berücksichtigung des Dialogfortschritts (im Falle von K_4) jeweils die korrekten und relevanten Informationen übermittelt werden.

Systementwurfsrichtlinie 14: Für jede Kommunikationsbeziehung des Typs Informationsübermittlung (Muster K_2) ist festzulegen, ob und welche Selektion oder Transformation des betreffenden Informationsartefakts Gegenstand der Aktorenkommunikation ist.

Systementwurfsrichtlinie 15: Es müssen Modelle für jedes Vorkommen einer Anfrage-Antwort-Kommunikation (K_3) sowie für jeden standardisierten Dialog (K_4) spezifiziert werden, die als Vorgabe für jede Kommunikationsinstanz dienen.

6.4.2 Realisierung der Interprozesskommunikation

Dieses Teilkapitel soll Ansatzpunkte für die Realisierung der Interprozess-Kommunikationsbeziehungen liefern, die im Kommunikationsmodell der Prozess-Sicht dargestellt wurden. Es sei an dieser Stelle daran erinnert, dass die Prozesse aus ISO/IEC 20000 als Referenz-Prozessrahmenwerk verwendet

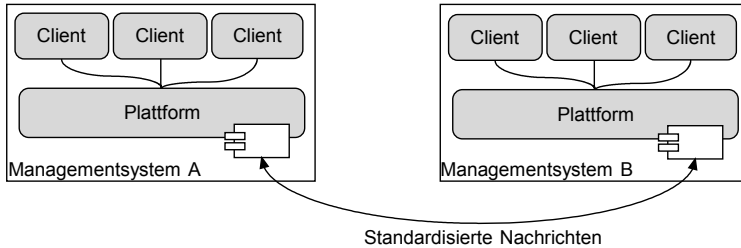


Abbildung 6.17: Ansatz für die Kopplung von Managementsystemen unter Kommunikationsgesichtspunkten

wurden. Auf diese Weise war es möglich, die Kommunikationsbeziehungen zwischen Service-Level-Management und anderen Dienstmanagementprozessen auf einer hinreichend konkreten Grundlage zu analysieren.

Wenn es darum geht, Informationszugriffe und -bereitstellungen im gemeinsamen Kontext eines Managementsystems für Service-Level-Management und anderer Managementsysteme im Dienstmanagement zu adressieren, so wird zu diesem Zweck zusätzlich ein *Referenz-Informationsmodell* benötigt, das den Bereich des Dienstmanagements und insbesondere natürlich die Prozesse aus ISO/IEC 20000 weitgehend gesamtheitlich abdeckt. Der nachfolgende Abschnitt erläutert, inwiefern das *Shared Information/Data Model (SID)* hierfür eine Basis darstellen kann.

6.4.2.1 Methodischer Ansatz

Abbildung 6.17 skizziert die bereits in der Einleitung zu dieser Arbeit präsentierte Idee eines dualen, integriert-kooperativen Ansatzes im Dienstmanagement. Die Grundidee ist einfach:

1. Innerhalb *einer* Disziplin des Dienstmanagements wird ein vollständig integriertes, IT-gestütztes Managementsystem gefordert bzw. vorausgesetzt. Für die Disziplin Service-Level-Management liefert diese Arbeit eine entsprechende Architektur. Alle sich aus der Anforderungsanalyse für eine Disziplin ergebenden Aufgaben und Aufgabenbereiche werden durch das jeweilige Managementsystem unterstützt, das den beteiligten Akteuren die dazu erforderliche Funktionalität über Nutzerschnittstellen verfügbar macht und die benötigten Daten in einem entsprechenden

Informationssystem vorhält. Eine gemeinsame Plattform dient als „Basiswerkzeug“, welches die funktionalen Eigenschaften des Managementsystems bündelt. (Mehr zu diesem Punkt im folgenden Kapitel 7.)

2. Zwischen den Managementsystemen *unterschiedlicher* Dienstmanagementdisziplinen wird durch diese Arbeit ein *kooperativer Ansatz* gefördert. Dabei erhält die Plattform jedes Managementsystems ein Kommunikationsmodul, das die Schnittstellen zu den Kommunikationsmodulen der Plattformen anderer Managementsysteme realisiert. Zwischen den Kommunikationsmodulen werden Management-relevante Informationen über wohldefinierte Schnittstellen und standardisierte Nachrichtenformate ausgetauscht.

Die Realisierung des kooperativen Ansatzes hängt von zwei kritischen Faktoren ab: Zum Einen wird ein rudimentäres, gemeinsames Verständnis über die Disziplinen und Prozesse des Dienstmanagements benötigt. Das allein reicht allerdings noch nicht aus. Darüber hinaus muss auch ein übergeordnetes, gemeinsames Informationsmodell existieren, das die *wesentlichen Informationsanforderungen und Artefakte* der einzelnen Disziplinen und Prozesse darstellen kann. In einer „idealen Welt“ würde dieses Informationsmodell aus einer widerspruchsfreien Synthese aller Informations- und Datenmodelle der einzelnen Managementsysteme bestehen (was wiederum den Weg hin zu einem vollständig integrierten Ansatz ebnen würde). Dies zu fordern oder anzunehmen, wäre mit dem Wissen um die Komplexität im Dienstmanagement und nach den Erfahrungen mit entsprechenden Projekten auf deutlich kleinerem Feld nicht zu rechtfertigen.

6.4.2.2 Realisierung eines einheitlichen Informationsmodells für Dienstmanagementprozesse

Die Auseinandersetzung mit der Thematik der Informationsmodellierung im IT-Dienstmanagement ist Gegenstand diverser Forschungsartikel des MNM-Teams. Ein Artikel aus dem Jahr 2006 [94] beschäftigt sich mit Entwurfsanforderungen und -kriterien für eine Konfigurationsmanagementdatenbank (*Configuration Management Database, CMDDB*), die gemäß *ITIL Service Support* [10] zwei Ziele verfolgt: die Bereitstellung eines logischen Abbilds der IT-Infrastruktur und aller IT-Dienste und die Funktion als zentrales Informationsverzeichnis (*information hub*) für alle Aktivitäten und Prozesse des Dienstmanagements. Es ist vor allem dieser zweite Aspekt, der für den Weg hin zu einem kooperativen Ansatz im Dienstmanagement, wie im vorherigen Abschnitt beschrieben, essenziell ist.

	IMM	CIM	SID	
Adaptability of Model	✗	✓	✓	Legend ✓ satisfied ✓ partially satisfied ✗ not satisfied - not applicable
Alignment to ITSM information needs	✗	✓	✓	
Comprehensive view	✗	✓	✓	
ITSM process artifacts	✗	✗	✓	
Integration with external databases	✗	✗	-	
Integration with management data stores	✓	✓	-	
Support for life cycle status accounting	✗	✓	✓	
Catalog of basic CI types	✓	✓	✓	

Abbildung 6.18: Analyse von IMM, CIM und SID auf ihre Eignung als Dienstmanagement-Informationsmodell

Speziell aufgrund der augenscheinlichen Ähnlichkeit zwischen Konfigurationselementen (*Configuration Items*) einer CMDB und gemanagten Objekten (*Managed Objects*) im Netz- und Systemmanagement wurden drei etablierte Managementmodelle aus dem Bereich des Netz- und Systemmanagements im Hinblick auf ihre Eignung als Informationsmodell für einen CMDB-Entwurf untersucht: das *Internet Management Model* (IMM) [95], das *Common Information Model* (CIM) [96] und das *Shared Information/Data Model* (SID) [97]. Das Ergebnis dieser Analyse im Hinblick auf acht zuvor abgeleitete Kriterien ist in Abbildung 6.18 zusammengefasst.

Insgesamt bietet SID zur Verwendung als Dienstmanagement-Informationsmodell das größte Potenzial. Das überrascht insofern nicht, als SID ja gerade als Informationsmodell für die *enhanced Telecom Operations Map* (eTOM) (vgl. Kapitel 4) entworfen wurde, die – ähnlich wie ISO/IEC 20000 – auch ein Prozessrahmenwerk speziell für Anbieter von Telekommunikationsdiensten beinhaltet. Betrachtet man die diversen Domänen-Addenda von SID, so erkennt man schnell, dass der Fokus dieses Modells auf der Darstellung von Diensten und Ressourcen, zahlreichen *Business entities* wie *Party* und *Location* sowie Produkten und Märkten liegt. Dienstmanagementprozess-spezifische Informationsartefakte werden nur sehr begrenzt abgedeckt.

Aus diesem Grund wird derzeit in einem weiteren Forschungsprojekt des MNM-Teams der Versuch unternommen, SID an dieser Stelle zu erweitern. Es wurde eine Methodik entwickelt [98], um auf der Basis von SID ein Informationsmodell für Dienstmanagementprozesse nach ITIL und ISO/IEC 20000 zu entwickeln. Ein solches Informationsmodell würde einen kooperativen Ansatz im Dienstmanagement ermöglichen.

6.4.2.3 Schlussfolgerungen

Solange im Dienstmanagement – speziell mit Blick auf den internationalen Standard ISO/IEC 20000 – kein einheitliches, übergeordnetes Informationsmodell existiert, ist nur eine *spezifische Kopplung* eines IT-gestützten Managementsystems für Service-Level-Management an andere Werkzeuge und Managementsysteme des Dienstmanagements möglich. Eine generische Managementarchitektur kann auf der Ebene der System-Sicht kaum Vorgaben machen oder Bausteine liefern, da hierfür (noch) das Fundament fehlt. Immerhin aber liefern die Empfehlungen des Interprozess-Kommunikationsmodells der Prozess-Sicht einen detaillierten Satz konkreter Kommunikationsanforderungen.

6.4.3 Zusammenfassung

Ausgehend vom Kommunikationsmodell der Prozess-Sicht wurden in diesem Teilkapitel Aspekte der Unterstützung der Aktorenkommunikation (Intraprozesskommunikation) durch ein IT-gestütztes Managementsystem adressiert. Speziell die Muster zur Klassifizierung von Kommunikationsbeziehungen wurden aufgegriffen und verfeinert.

Im Zusammenhang mit der Interprozesskommunikation wurde der Ansatz integriert-kooperativer Managementsysteme erläutert und begründet, warum dieser nur auf Basis eines gemeinsamen, übergeordneten Informationsmodells realisierbar ist. Das *Shared Information Data/Model* liefert hierfür zwar eine Basis, muss allerdings maßgeblich erweitert werden. Dies ist Gegenstand zukünftiger Arbeiten.

Kapitel 7

Plattformentwicklung und Implementierungsaspekte

Inhalt dieses Kapitels

7.1 Realisierung einer Managementplattform	358
7.1.1 Komponenten einer Managementplattform	359
7.1.2 Schritte zum Plattformentwurf	361
7.2 Exemplarische Instanziierung des Informationsmodells	364
7.2.1 Umsetzung	364
7.2.2 Fazit	368

Dieses Kapitel widmet sich der Anwendung der verschiedenen Module der Managementarchitektur im Rahmen eines konkreten Implementierungsprojekts mit dem Ziel der Erstellung einer Plattform für Service-Level-Management.

Fragestellung dieses Moduls: Folgende Frage ist im Zusammenhang mit dem Entwurf eines Implementierungsleitfadens zu beantworten:

Wie lässt sich mit Hilfe des in den Kapiteln 5 und 6 dieser Arbeit entwickelten „Baukastens“, den Teilmodellen der Prozess- und Systemsicht, ein integriertes, IT-gestütztes Managementsystem zur Unterstützung der Aufgaben im Service-Level-Management konzipieren?

Daneben wird im Abschnitt 7.2 eine exemplarische Instanziierung des Informationsmodells vorgestellt, die für eines der IT-Szenarien aus der Anforderung

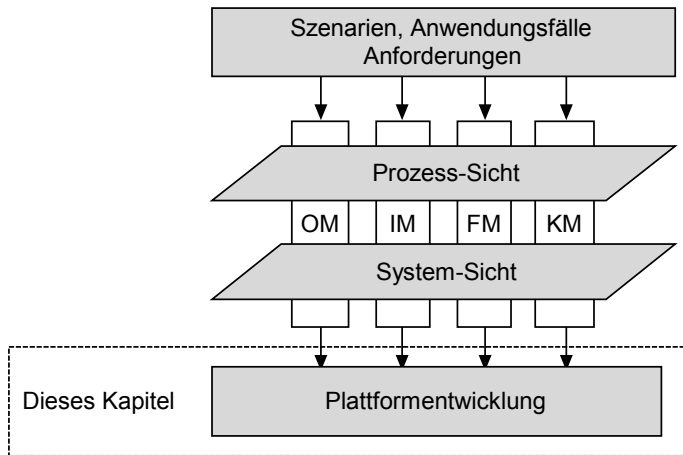


Abbildung 7.1: Einordnung dieses Kapitels in den Ordnungsrahmen

rungsanalyse zeigt, dass das Informationsmodell in der Lage ist, den Informationsbedarf des Service-Level-Managements in diesem Szenario abzudecken.

7.1 Realisierung einer Managementplattform

Bevor sich dieser Abschnitt mit der Realisierung einer Managementplattform beschäftigt, wird zunächst eine begriffliche Abgrenzung zwischen den Konzepten *Managementsystem* und *Managementplattform* vorgenommen und kurz erklärt, wie diese beiden zusammenhängen.

Wie in Kapitel 2.3 definiert, wird ein (IT-gestütztes) *Managementsystem* als die „Gesamtheit aller informationstechnischen Komponenten (einschließlich Hard- und Software) bezeichnet, die in koordinierter Weise eingesetzt werden, um die Planung, Ausführung und Dokumentation anfallender Managementaufgaben effektiv und effizient zu unterstützen“. Ein *integriertes Managementsystem* liegt vor, wenn „Informationen über wohldefinierte Schnittstellen zugänglich sind und eine Managementplattform zur Verfügung steht, die ein offenes Trägersystem für Managementanwendungen bereitstellt“ (vgl. Kapitel 1).

Somit ist eine *Managementplattform* das maßgebliche technische Fundament für ein integriertes, IT-gestütztes Managementsystem. Vor diesem Hintergrund

illustriert Abbildung 7.2 den Aufbau von Managementplattformen in Anlehnung an Hegering et al. [1]. Obwohl dieses Modell speziell für den Einsatzbereich im integrierten Management vernetzter Systeme entwickelt wurde, lassen sich seine Grundkonzepte auch auf das Dienstmanagement übertragen.

7.1.1 Komponenten einer Managementplattform

Mit der Plattform-*Infrastruktur* ist nicht etwa die Netzinfrastruktur der Umgebung gemeint, die die Plattform an das verteilte System anbindet, sondern die grundlegende Plattform-Logik. Der Begriff der Infrastruktur mag zwar auf den ersten Blick ungünstig gewählt erscheinen, bringt allerdings sehr gut zum Ausdruck, dass diese Komponente der Plattform ebenso wenig wegzudenken ist, um die Gesamtfunktionalität der Plattform zu erzielen, wie eine Netzinfrastruktur von einem verteilten System, das ohne Konnektivität nahezu nicht mehr als Gesamtsystem arbeitsfähig ist.

Die Plattform-Infrastruktur besteht aus dem Plattform-*Kernsystem*, welches auf den *Kommunikationsbausteine* sowie den Bausteine der *Informationsverwaltung* zugreift. Der Kommunikationsbaustein stellt Dienste zur Anbindung des mittels der Plattform realisierten Managementsystems an andere Managementsysteme und somit zum Zugriff auf entfernte *Managed Objects* bereit, während der Baustein der Informationsverwaltung den Zugriff auf die Plattform-„eigenen“ *Managed Objects* realisiert.

Über eine Programmierschnittstelle (*Application Programming Interface, API*) greifen *Basisanwendungen* auf das Kernsystem der Plattform-Infrastruktur zu. Zu diesen Basisanwendungen zählen im Falle von Netzmanagementplattformen unter anderem ein Konfigurationsmanager, ein Topologiemanager, ein Leistungsmonitor, ein MIB-Browser, ein Ereignismanager und ein Zustandsmonitor. Es handelt sich hierbei um vergleichsweise rudimentäre Anwendungen, die sich zumeist auf eine Visualisierung der Informationen zu den *Managed Objects* beschränken.

Auf den Basisanwendungen setzen komplexere *Managementapplikationen* auf. Aus Sicht der Softwaretechnik nehmen Basisanwendungen damit eine vergleichbare Rolle ein wie beispielsweise Bibliotheken in der C-Programmierung. Managementapplikationen machen also Gebrauch von den Grundfunktionen, die durch die Basisanwendungen bereit gestellt werden, und sind aus Sicht des Benutzers des Systems (zum Beispiel ein Netzwartungsbeauftragter) die eigentlich interessanten Elemente der Plattform. Denn in ihnen werden Funktionen realisiert und gebündelt, die sich aus den Anwendungsfällen des jeweiligen Managementbereichs ergeben.

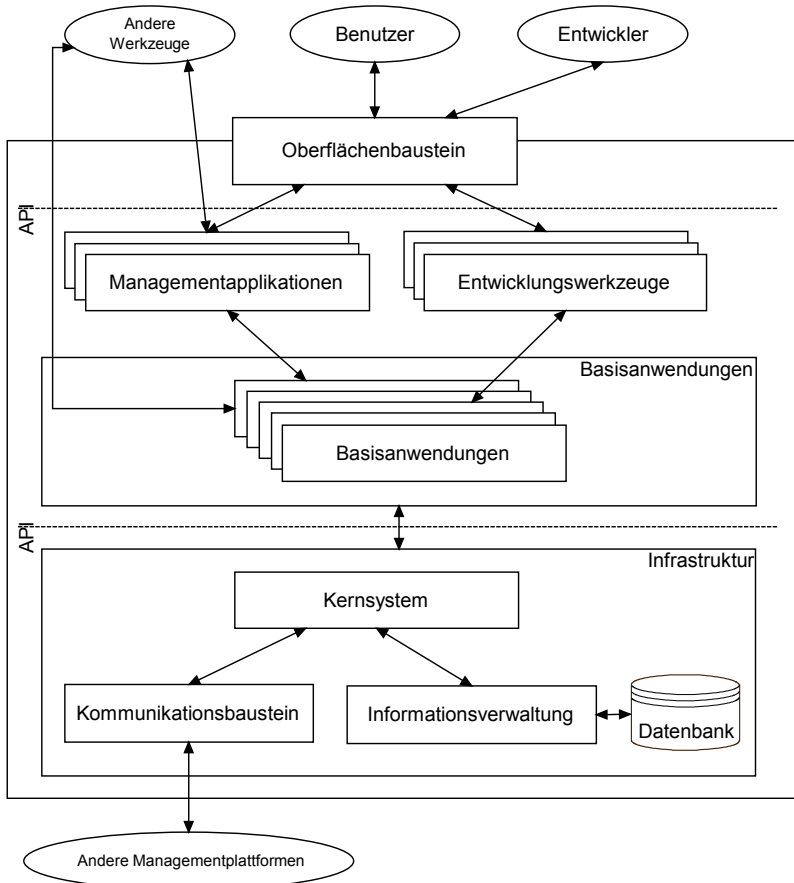


Abbildung 7.2: Aufbau von Managementplattformen nach Hegering et al. [1]

Eine grafische Benutzeroberfläche (*Graphical User Interface, GUI*) ermöglicht einem Benutzer interaktiven Zugang zu Funktionen und Anwendungen der Plattform und damit insbesondere zu den Managementapplikationen. Die Betrachtung der Entwickler-Rolle sowie von Entwicklungswerkzeugen (zum Beispiel bestimmte Programmiersprachen oder Muster der Softwareentwicklung) komplettieren diese Betrachtung.

Es soll nun also ein Ansatz zur Adoption des Plattformkonzepts aus dem integrierten Management vernetzter Systeme im Dienstmanagement vorgestellt werden. Dazu wird der prinzipielle Plattformaufbau beibehalten, die enthaltenen Bausteine aber aus Dienstmanagement-Sicht beschrieben und ggf. verfeinert bzw. ergänzt.

7.1.2 Schritte zum Plattformentwurf

In diesem Abschnitt wird aufgezeigt, wie die Modelle der Kapitel 5 und 6 verwendet werden können, um eine Managementplattform für Service-Level-Management schrittweise zu implementieren und somit ein IT-gestütztes Managementsystem zu realisieren. Die entsprechende Methodik orientiert sich an den beschriebenen Bausteinen einer Managementplattform. Sie besteht aus den in Abbildung 7.3 dargestellten fünf Schritten.

Schritt 1: Entwurf des Bausteins zur Informationsverwaltung Der Entwurf des Bausteins zur Informationsverwaltung stellt den ersten logischen Schritt auf dem Weg der Plattformimplementierung dar. Das Informationsmodell der Managementarchitektur liefert hierfür das Fundament. Zur Realisierung des Informationssystems für Service-Level-Management wird ein geeignetes Datenbanksystem benötigt. Sowohl relationale Datenbanken als auch Systeme, die auf anderen Datenmodellen aufbauen (zum Beispiel objektorientierte oder objektrelationale Datenbanken) können eingesetzt werden.

Die Anforderungen des Informationssystems an die zugrunde liegende Datenbank werden durch praktisch alle verbreiteten Standardlösungen aus diesem Bereich erfüllt, wie zum Beispiel *MySQL Server* oder *PostgreSQL* als populäre *Open Source*-Produkte. Eine interessante Alternative bietet *eXist*, ein Datenbanksystem, welches auf XML-Technologien basiert. Daten werden nicht in Tabellen (Relationen), sondern in einer Baumstruktur gespeichert, und die Anfragesprache ist nicht SQL, sondern XQuery.

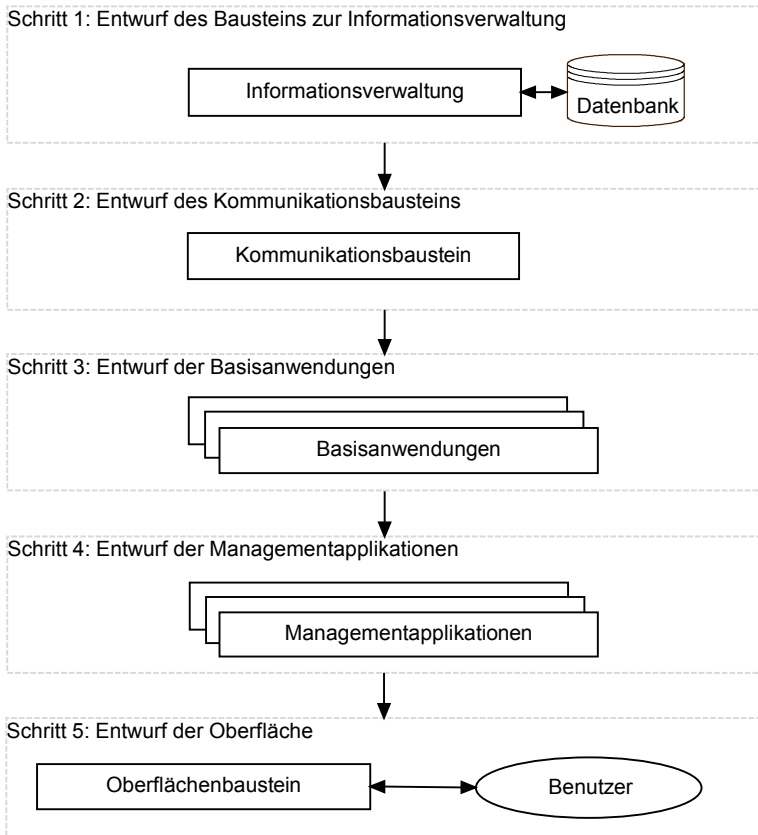


Abbildung 7.3: Schritte zum Entwurf einer Plattform für Service-Level-Management

Schritt 2: Entwurf des Kommunikationsbausteins Der Kommunikationsbaustein muss sowohl die Vorgaben aus dem Intra- als auch aus dem Interprozess-Kommunikationsmodell umsetzen. Somit läuft jegliche Aktorenkommunikation sowie die Kommunikation aus dem Managementsystem heraus mit anderen Managementsystemen über dieses Modul. Gemäß der System-Sicht der Architektur müssen speziell die Modelle für die Anfrage-Antwort-Kommunikation sowie für standardisierte Dialoge spezifiziert und an das Workflow-Management-System (siehe Schritt 3) gekoppelt werden.

Schritt 3: Entwurf der Basisanwendungen Steht das Kernsystem, bestehend aus Informations- und Kommunikationsbaustein, zur Verfügung, so können im nächsten Schritt Basisanwendungen entworfen werden, die darauf operieren. Folgende Basisanwendungen werden mindestens benötigt:

- Workflow-Management-System (WFMS): Diese Basisanwendung dient dem Zweck, die Workflows der Managementsystem-Anwendungsfälle einpflegen und administrieren zu können. Sie steuert die Einhaltung aller Kontrollflüsse.
- Artefakt-Browser: Der Artefakt-Browser steht in enger Analogie zum MIB-Browser einer Netzmanagementplattform. Er setzt direkt auf dem Baustein zur Informationsverwaltung auf und ermöglicht die Spezifikation eines Satzes an Standardanfragen auf dem Informationssystem sowie deren Abruf durch die Managementapplikationen.

Schritt 4: Entwurf der Managementapplikationen Die Managementapplikationen setzen auf den Basisanwendungen auf und realisieren die Service-Level-Management-relevanten Plattformfunktionen. Die maßgeblichen Vorgaben für den Entwurf der Managementapplikationen liefert daher das Funktionsmodell der Managementarchitektur. Angelehnt an die Hauptfunktionsbereiche ergeben sich vier zu realisierende Managementapplikationen:

- Dienstinformationsmanagementapplikation
- Stakeholder-Managementapplikation
- Vereinbarungsmanagementapplikation
- Messdaten- und Berichtsmanagementapplikation

Schritt 5: Entwurf der Oberfläche Schließlich müssen gemäß Organisationsmodell der System-Sicht einerseits Bediensysteme (*Frontends*) entworfen und implementiert werden, die es den Systembenutzern erlauben, auf die Managementapplikationen zuzugreifen. Andererseits müssen, wo dies sinnvoll ist, Anbindungen an bestehende andere Werkzeuge geschaffen werden.

7.2 Exemplarische Instanziierung des Informationsmodells

Das Herzstück jeder Managementarchitektur ist ihr Informationsmodell, wesentlicher Bestandteil des Kernsystems der Managementplattform ihr Informationsbaustein. Um zu zeigen, dass das im Rahmen der Prozess-Sicht entwickelte und auf der Ebene der System-Sicht verfeinerte Informations- und Datenmodell in der Lage ist, reale Informationsanforderungen abzubilden, greift dieses Teilkapitel eines der Szenarien aus der Anforderungsanalyse wieder auf und instanziiert das Informationsmodell mit einem (kleinen) Teil der Informationen aus diesem Szenario. Neben einem praxisorientierten *Proof of Concept* für das Informationsmodell wird damit auch die Semantik der einzelnen Komponenten am ganz konkreten Beispiel veranschaulicht.

7.2.1 Umsetzung

Es wird Bezug genommen auf das Szenario 2 (LRZ), das in Kapitel 3.1.3 beschrieben wurde. Modelliert wird folgende Situation: Das LRZ bietet einen *Wireless LAN*-Dienst (WLAN) als Teil des Dienstportfolios an. Dieser wird aus dem für den Kunden *Ludwig-Maximilians-Universität* (LMU) gültigen Dienstkatalog referenziert. Der Kunde LMU hat diesen Dienst beauftragt, und eine entsprechende Dienstvereinbarung wurde abgeschlossen. Die vereinbarte Dienstunterstützung entspricht dem zur Verfügung stehenden Standard-Support-Level.

Quelle für die Modellierung der Informationen dieses Ausschnitts aus dem LRZ-Szenario waren vorhandene technische und Nutzer-Dokumentationen über den WLAN-Dienst. Abbildung 7.4 visualisiert einen überblickartigen Auszug der resultierenden Instanz des SLM-Informationssystems, Abbildung 7.5 „zoomt“ in die Instanz des Dienst-Artefakts, Abbildung 7.6 visualisiert einen Teil der Dienstvereinbarung. Die vollständige XML-Datei ist im Code-Listing B.1 des Anhangs abgedruckt.

XML	
ServiceLevelManagementInformationSystem	
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespac...	H:\diss\neu\XML\ServiceLevelManagementInformationSystem.xsd
ServicePortfolio	
Service id=srv-1 date=2008-11-18	
ServiceCatalog	
id	sc-1
referenceToCustomerFromServiceCata...	
referenceToServiceFromServiceCatalog	
serviceCatalogValidity	
serviceCatalogHi...	
CustomerDatabase	
Customer	
id	cust-1
name	LMU
customerContactInformation	
customerRoleAssignments	
customerServiceInterfaces	
SDPDatabase	
SLPDatabase	
SupportLevelProfile	
id	slp-1
name	Standard Support
supportChannels	
supportTimes	
maxResponseTimes	
maxRecoveryTi...	
SLADatabase	
ServiceLevelAgreement	
id	sla-1
referenceToServiceFromSLA idref=sc-1...	
referenceToCustomerFromSLA idref=cu...	
referenceToSLPFromSLA idref=slp-1	
serviceDeliveryTargets	
serviceLevelTargets	
serviceDegradationDefinition	
deliveryConditio...	
SLAHistory	
SADatabase	
MeasurementDat...	
ReportDatabase	
ConflictDatabase	

Abbildung 7.4: Visualisierung der exemplarischen Instanz des Informationsmodells (Auszug, Teil 1)

Service									
id	snv-1								
date	2008-11-18								
name	Wireless LAN								
serviceDescription	In vielen Bereichen der Muenchener Universitaeten und Hochschulen ist ein Netzzugang ueber WLAN (Wireless LAN) moeglich. WLAN bietet insbesondere allen mobilen Nutzern mit Laptop, PDA oder aehnlichem die Gelegenheit, drahtlos die IT-Dienste im MMN und naeherlich auch im Internet zu nutzen.								
serviceRequirements	<table border="1"> <thead> <tr> <th>description</th> <th>priority</th> </tr> </thead> <tbody> <tr> <td>1 WLAN-Zugang in Hoersaalen</td> <td>1</td> </tr> <tr> <td>2 WLAN-Zugang in uebrigen Raerumen</td> <td>2</td> </tr> <tr> <td>3 WLAN-Zugang in Aussenbereichen</td> <td>3</td> </tr> </tbody> </table>	description	priority	1 WLAN-Zugang in Hoersaalen	1	2 WLAN-Zugang in uebrigen Raerumen	2	3 WLAN-Zugang in Aussenbereichen	3
description	priority								
1 WLAN-Zugang in Hoersaalen	1								
2 WLAN-Zugang in uebrigen Raerumen	2								
3 WLAN-Zugang in Aussenbereichen	3								
serviceSpecification	<ul style="list-style-type: none"> serviceUsageFu... Uneingeschaenktter Internetzugang, verschliesserte Ueberrtragung serviceAccessP... Standard Access Point serviceClientSpe... Cisco VPN-Client auf Endgeraet mit Adapter fuer Wireless LAN im IEEE 802.11g- oder IEEE 802.11b-Stand... serviceManagem... von WLAN Access Point Hardware serviceManagem... von WLAN Access Point Hardware serviceResources serviceDependencies <table border="1"> <thead> <tr> <th>dependency</th> </tr> </thead> <tbody> <tr> <td>1 Physische Netzanbindung MMN</td> </tr> <tr> <td>2 DHCP</td> </tr> <tr> <td>3 DNS</td> </tr> </tbody> </table>	dependency	1 Physische Netzanbindung MMN	2 DHCP	3 DNS				
dependency									
1 Physische Netzanbindung MMN									
2 DHCP									
3 DNS									
serviceDeliveryParameters									
serviceLevelMetrics									
serviceStatus	active-operating								
serviceHistory									

Abbildung 7.5: Visualisierung der exemplarischen Instanz des Informationsmodells (Auszug, Teil 2)

ServiceLevelAgreement	id	sla-1
referenceToServiceFromSLA	idref	sc-1-srv-1
referenceToCustomerFromSLA	idref	cust-1
referenceToSLPPFromSLA	idref	slp-1
serviceDeliveryTargets	referenceToServiceDeliveryParameterFromSLA	idref
	serviceDeliveryWarranty	120000
serviceLevelTargets	referenceToServiceLevelParameterFromSLA	idref
	serviceLevelWarranty	99.8
serviceDegradationDefinition	serviceDegradationDegree	10
	degradationSubject	Physische Access Point Hardware
	degradationManner	Ohne Funktion
	degradationTime	Laenger als 15 Minuten
deliveryCondition...		
SLAHistory		

Abbildung 7.6: Visualisierung der exemplarischen Instanz des Informationsmodells (Auszug, Teil 3)

7.2.2 Fazit

Alle in diesem Teilszenario managementrelevanten Informationen bezüglich Dienst (WLAN), Kunde (LMU), Dienstkatalog und Dienstvereinbarung ließen sich problemlos auf Basis des Informations- und Datenmodells der Architektur realisieren. Dies ist vor allem dem streng deduktiven *Top-Down-Ansatz* geschuldet, der mit dieser Arbeit verfolgt wurde. Die funktionalen Anforderungen an ein IT-gestütztes Managementsystem für Service-Level-Management wurden aus realen IT-Szenarien abgeleitet und zu Anwendungsfällen verfeinert. Diese wiederum bildeten die Basis für alle Architekturteilmodelle – so auch für das Informationsmodell.

Obwohl diese Vorgehensmethodik allein natürlich noch keine Garantie dafür liefert, dass die entwickelten Modelle und Modellelemente alle denkbaren Facetten einer Szenarioausprägung auch tatsächlich berücksichtigen, so stellt sie doch zumindest sicher, dass jede funktionale Anforderung, die zu Beginn identifiziert wurde, auch in späteren Phasen der Modellentwicklung noch ihren Niederschlag findet.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

Als eine der Kerndisziplinen des dienstorientierten IT-Managements soll ein effektiv betriebenes Service-Level-Management (SLM) sicherstellen, dass alle IT-Dienste eines Providers auf einem Qualitätsniveau erbracht werden, welches die kundenspezifischen Anforderungen an den Nutzen dieser Dienste erfüllt. Dabei sind zwei wesentliche Nutzendimensionen zu berücksichtigen: die funktionalen Eigenschaften des Dienstes sowie sein Leistungsverhalten. Zudem können verschiedene Kunden sehr unterschiedliche Anforderungen an Funktionalität und Leistungsverhalten des prinzipiell gleichen IT-Dienstes stellen, die ganz maßgeblich davon abhängen, welche *Geschäftsprozesse* wie stark durch den Dienst unterstützt werden.

8.1.1 Ausgangspunkt der Arbeit

Die genannte Zielsetzung des Service-Level-Managements gewinnt in einem zunehmend geschäftsorientierten IT-Umfeld zusätzlich an Bedeutung. Der Leitgedanke des *Business IT Alignment* hat in vielen IT-Organisationen längst zu einem Paradigmenwechsel geführt – weg von einer rein technologiegeprägten Perspektive und hin zu einem stärkeren Bewusstsein für die Zusammenhänge zwischen qualitativ hochstehenden IT-Diensten und der Fähigkeit eines Unternehmens, Produkte und Dienstleistungen zu produzieren, die am Markt bestehen können. Es hat sich also die Erkenntnis etabliert, dass IT-Dienste zwar indirekt, aber in nicht unmaßgeblichem Umfang das Gewinnpotenzial eines Unternehmens beeinflussen.

Service-Level-Management ist einer der Schlüssel für ein geschäftsorientiertes IT-Management. Das wichtigste Instrument in diesem Kontext sind Dienstvereinbarungen (*Service Level Agreements*, SLAs), die zwischen einem IT-Provider und seinen Kunden geschlossen werden und die zu liefernden Dienste spezifizieren, Service-Level-Ziele dokumentieren und Verantwortlichkeiten festlegen. Zahlreiche Veröffentlichungen, *Best Practice*-Rahmenwerke, Forschungsarbeiten und -projekte und auch der Dienstmanagement-Standard ISO/IEC 20000 liefern Konzepte, Modelle, Empfehlungen und (Mindest-)Anforderungen an die Umsetzung von Service-Level-Management. Gerade die verbreiteten Prozessrahmenwerke konzentrieren sich dabei allerdings eher auf einen organisationstheoretischen Betrachtungswinkel. Ein für ein effektives Service-Level-Management ganz wesentlicher Aspekt wird dabei meistens ausgeblendet oder nur sehr oberflächlich adressiert: *die Unterstützung SLM-spezifischer Aufgaben und Aktivitäten durch geeignete automatisierte Managementwerkzeuge*.

Hier setzt die vorliegende Arbeit an, deren übergeordnete Zielsetzung darin besteht, diese Lücke zu füllen und den Weg zu einem IT-gestützten Managementsystem für Service-Level-Management zu ebnen. Die größten initialen Herausforderungen stellen in diesem Zusammenhang die zum Teil sehr unklaren Detailanforderungen an das Service-Level-Management und die unscharfe Abgrenzung gegen andere Dienstmanagementdisziplinen dar.

Der aktuelle Stand von Wissenschaft und Praxis zum Thema Service-Level-Management weist im Hinblick auf die Realisierung eines IT-gestützten Managementsystems vor allem die folgenden Defizite auf:

- Es fehlt ein formales Prozessmodell für Service-Level-Management, das alle Teilaufgaben, Aktivitäten, Rollen, Schnittstellen und Informationsartefakte umfasst und ihre Zusammenhänge vollständig und konsistent abbildet.
- In Folge dessen existieren derzeit keine eindeutigen und klar formulierten Erwartungen, Anforderungen und Vorgaben hinsichtlich des zu realisierenden Funktionsspektrums eines IT-gestützten Managementsystems für SLM.

Diese Defizite führen dazu, dass heute im Bereich Service-Level-Management eine unzureichende Unterstützung durch automatisierte Managementlösungen existiert und dass die wenigen existierenden Werkzeuge (*Tools*) lose gekoppelte „Insellösungen“ darstellen, die keine gemeinsame Informationsbasis nutzen und keine wohldefinierten Schnittstellen zueinander realisieren.

8.1.2 Methodisches Vorgehen und wesentliche Ergebnisse

Um die genannten Defizite zu adressieren, wurde ein modellbasierter Ansatz gewählt, dessen Ziel darin bestand, eine *Managementarchitektur* zu entwerfen, die sowohl eine fundierte Anforderungsanalyse und ein formales Prozessmodell liefert als auch systementwurfsrelevante Aspekte der Implementierung einer Managementplattform umfasst.

Szenariobasierte Anforderungsanalyse Im Rahmen einer fundierten und auf realen IT- und Unternehmensszenarien aufbauenden Anforderungsanalyse (Kapitel 3) wurde zunächst das erforderliche Funktionsspektrum eines Managementsystems für SLM ermittelt und skizziert, wobei zwischen *Kernfunktionalitäten*, zu deren Realisierung benötigten *Basisfunktionalitäten* sowie *erweiterten Funktionalitäten* differenziert wurde. In diesen Kategorien wurden in einem zweiten Schritt insgesamt 38 Anwendungsfälle definiert und initial spezifiziert, die das Fundament für die weitere Entwicklung der Managementarchitektur darstellten. Die Vollständigkeit und Plausibilität von Funktionsspektrum und Anwendungsfällen wurden durch eine morphologische Analyse der zugrunde liegenden Szenarien nachgewiesen.

Ordnungsrahmen mit horizontaler und vertikaler Schichtenbildung Um die Komplexität der Aufgabenstellung und somit den Problemraum beherrschbar zu machen, wurde die Gesamtaufgabe *Entwicklung einer Managementarchitektur für SLM* gemäß dem Prinzip *Divide and Conquer* auf zwei horizontale und vier vertikale Schichten verteilt. So entstand zugleich ein statischer Ordnungsrahmen aus acht Modulen, die den Kern der Managementarchitektur darstellen. Auf der ersten horizontalen Ebene, der *Prozess-Sicht* der Architektur (Kapitel 5), wurde ein detailliertes, formales Prozessmodell für Service-Level-Management entwickelt. Auf der zweiten Ebene, der *System-Sicht* der Architektur (Kapitel 6), wurden ausgehend von den Modellen der Prozess-Sicht Erweiterungen, Verfeinerungen und Entwurfsrichtlinien im Hinblick auf das Ziel *Automatisierung* vorgenommen. Auf beiden Ebenen wurde die gleiche vertikale Partitionierung in ein *Organisationsmodell*, ein *Informationsmodell*, ein *Funktionsmodell* und ein *Kommunikationsmodell* vorgenommen.

Entwicklungsrahmen zur Darstellung von Modellzusammenhängen Neben dem statischen Ordnungsrahmen wurde ein Entwicklungsrahmen spezifiziert, der darstellt, wie die einzelnen Teilmodelle beider Ebenen zusammenhängen. Dieser Entwicklungsrahmen stellt eine Art *Meta-Architektur* dar.

Differenziertes Organisationsmodell Im Rahmen des Organisationsmodells wurde ein generisches Domänen- und Rollenmodell für Service-Level-Management spezifiziert, auf dem nicht nur das Funktions- und Kommunikationsmodell maßgeblich aufbauen, sondern das sich auf der System-Sicht auch mit Aspekten von Nutzerschnittstellen und Zugriffskontrolle beschäftigt. Eine Interaktivitätsanalyse auf Basis von Rollen, Verantwortlichkeiten und Aktivitäten lieferte wichtige Entwurfskriterien für die Realisierung der Nutzer-System-Interaktionen.

Integriertes Informations- und Datenmodell Als Herzstück der Managementarchitektur für SLM wurde ein Informationsmodell entwickelt, das sämtliche aus den Anwendungsfällen abgeleiteten Informationsanforderungen erfüllt. Dieser Teil der Managementarchitektur wurde in drei wesentlichen Schritten entwickelt: Zuerst wurden die SLM-spezifischen *Grundkonzepte* und ihre groben Zusammenhänge definiert. Im zweiten Schritt wurde ein Klassenmodell entwickelt, welches die realisierenden *Informationsartefakte* darstellt und zu den Grundkonzepten in Beziehung setzt. Der letzte Schritt bestand aus einer Feinmodellierung im Rahmen eines XML-basierten *Datenmodells*.

Workflow-orientiertes Funktionsmodell Ein Schwerpunkt des SLM-Prozessmodells wurde auf die Definition von Workflows für die Anwendungsfälle der Basis- und Kernfunktionalitäten gelegt. Diese Workflows stellen neben den *Kontrollflüssen* auch die *Informationsflüsse* im Service-Level-Management dar und greifen dazu das Artefakt-Klassenmodell des Informationsmodells auf.

Duales Kommunikationsmodell zur Abbildung der Aktoren- und Prozesskommunikation Ziel des Kommunikationsmodells war es, alle Kommunikationsbeziehungen zwischen den Aktoren des Service-Level-Managements einerseits sowie zwischen einem IT-gestützten Managementsystem für SLM und anderen Dienstmanagementwerkzeugen und -systemen andererseits zu identifizieren, zu kategorisieren und im Hinblick auf die Realisierung eines Kommunikationsbausteins einer Managementplattform zu formalisieren.

Methodik zum Plattformentwurf Abschließend wurde, ausgehend von einem generischen Managementplattform-Modell, eine Methodik vorgestellt, die aufzeigt, wie die einzelnen Module der Managementarchitektur in fünf Syntheseschritten für die Implementierung einer IT-gestützten Managementplattform für Service-Level-Management instrumentiert werden können (Kapitel 7).

8.1.3 Offene Punkte

Mit den beschriebenen Ergebnissen liefert die Arbeit eine vollständige und generische Managementarchitektur für ein IT-gestütztes Service-Level-Management. Sie bietet damit zugleich Anknüpfungspunkte für einige mögliche Erweiterungen, die nicht mehr in den Lösungsraum dieser Arbeit fallen. Als offene Punkte seien daher in diesem Sinne die folgenden genannt:

- Das Funktionsmodell der Systemsicht beinhaltet eine umfassende Automatisierbarkeitsanalyse aller Workflow-Aktivitäten des Prozessmodells. Hier konnte aus Gründen des Umfangs nur für eine Auswahl der erforderliche nächste Verfeinerungsschritt durchgeführt und somit konkret gezeigt werden, wie das volle Automatisierungspotenzial durch einen geeigneten Systementwurf ausgeschöpft werden kann.
- Das Interprozesskommunikationsmodell liefert zwar einen Ansatz zur Realisierung des Kooperationsparadigmas zwischen einem SLM-unterstützenden Managementsystem und den (potenziellen) Managementsystemen anderer Dienstmanagementdisziplinen, musste allerdings auf der Ebene der System-Sicht in Ermangelung eines umfassenden Dienstmanagement-Informationsmodells auf eine Feinmodellierung der Kommunikation verzichten.
- Der Aspekt einer vertikalen Integration eines IT-gestützten Managementsystems für Service-Level-Management mit Managementwerkzeugen des Netz- und Systemmanagements wurde nicht im Detail adressiert. Dies stellt aber insbesondere für den Funktionsbereich *Messdaten- und Berichtsmanagement* eine Betrachtungsebene dar, die nicht nur beim konkreten Systementwurf relevant, sondern auch für eine generische Managementarchitektur sinnvoll sein kann. Hier geht es zum Beispiel um die Frage der Schnittstellenbildung zwischen technischen Dienst- und Infrastruktur-Überwachungswerkzeugen (*Monitoring Tools*) und den SLM-spezifischen Anwendungsfällen der Messdatenspezifikation einerseits und Berichterstellung andererseits. Debusmann, dessen Arbeit [77] in Kapitel 4 vorgestellt wurde, liefert mit seiner modellbasierten Lösung einen Ansatz, der bereits einen ähnlichen Weg einschlägt – allerdings in isolierter Betrachtung und ohne Einbettung in ein übergeordnetes Werkzeugkonzept für Service-Level-Management.

8.2 Ausblick

Mit ihrer methodischen Herangehensweise und den erzielten Ergebnissen bietet diese Arbeit einen soliden Grundstock für weiterführende Projekt- und Forschungsfragestellungen. Zwei der wichtigsten und vielversprechendsten werden im Folgenden überblicksartig dargestellt.

Entwurf eines übergeordneten Dienstmanagement-Informationsmodells Eine mögliche Strategie, um zu gewährleisten, dass IT-gestützte Werkzeuglösungen einzelner Dienstmanagementdisziplinen miteinander kommunizieren können, besteht in der Definition eines übergeordneten Dienstmanagement-Informationsmodells. Der Detaillierungsgrad eines solchen Informationsmodells sollte so gewählt werden, dass alle Informationsobjekte abgedeckt werden, die die Grenzen einzelner Managementprozesse überschreiten. Informationen, die nur innerhalb eines Prozesses relevant sind, müssen in einem übergeordneten Informationsmodell nicht abgebildet werden. Dadurch bleibt die Komplexität auf einem vertretbaren Niveau, und gleichzeitig werden durch das resultierende Informationsmodell keine unnötig einschränkenden und normativen Vorgaben bezüglich der Informationsmodellierung innerhalb einzelner, prozessspezifischer Managementsysteme gemacht.

Das *Shared Information/Data Model* (SID) [97] verfolgt in der Tendenz einen solchen Ansatz für die *enhanced Telecom Operations Map* (eTOM) [20]. Eine Adaptierung von SID für ein Dienstmanagement-Umfeld, das sich eher am Prozessrahmenwerk von ISO/IEC 20000 (oder ITIL) orientiert, stellt eine sinnvolle Alternative zu einer vollständigen Neuentwicklung dar. Wichtige Teilfragestellungen sind in diesem Zusammenhang:

- Welche Informationsartefakte müssen im Modell enthalten sein?
- Welche dieser Artefakte sind bereits (in welchem Umfang) Teil von SID? (Eine rein syntaktische Analyse reicht in diesem Punkt nicht aus. Vielmehr muss die Semantik der unterschiedlichen Konzepte berücksichtigt werden, was diesen Schritt schnell sehr aufwändig werden lässt.)
- An welchen Stellen muss SID um neue Klassen, zusätzliche Attribute und Beziehungen erweitert werden?

Die Entwicklung eines Dienstmanagement-Informationsmodells wird vom MNM-Team angestrebt und war bereits Gegenstand einiger Veröffentlichungen [94, 89, 98].

Anwendung der Methodik zur Architekturentwicklung auf andere Dienstmanagementdisziplinen Der methodische Ansatz, der dieser Arbeit zugrunde liegt (vgl. Kapitel 2), ist nicht in allen Teilen spezifisch auf Aspekte des Service-Level-Managements ausgerichtet. Insbesondere der Ordnungsrahmen, die Systematik der szenariobasierten Anforderungsanalyse sowie eine geeignete Adaption des Entwicklungsrahmens der Architektur lassen sich auch auf andere Dienstmanagementdisziplinen anwenden. Auf diese Weise könnten weitere Managementarchitekturen entwickelt werden, die bei konsistentem und einheitlichem Einsatz der Modellierungswerkzeuge zu einer Gesamt-Managementarchitektur für den Bereich des Dienstmanagements gebündelt werden könnten.

Anhang A

SLM-Informationssystem und Datenmodelle

Dieser Anhang enthält die XML-Dokumentenschemata, die als Datenmodelle für die Informationsartefakte fungieren, die im Rahmen des Informationsmodells der System-Sicht der Managementarchitektur entwickelt wurden.

A.1 ServiceLevelManagementInformationSystem

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
   qualified" attributeFormDefault="unqualified">
3   <xs:include schemaLocation="ServicePortfolio.xsd"/>
4   <xs:include schemaLocation="CustomerDatabase.xsd"/>
5   <xs:include schemaLocation="SDPDatabase.xsd"/>
6   <xs:include schemaLocation="SLPDatabase.xsd"/>
7   <xs:include schemaLocation="SLADatabase.xsd"/>
8   <xs:include schemaLocation="SADatabase.xsd"/>
9   <xs:include schemaLocation="MeasurementDatabase.xsd"/>
10  <xs:include schemaLocation="ReportDatabase.xsd"/>
11  <xs:include schemaLocation="ConflictDatabase.xsd"/>
12  <xs:element name="ServiceLevelManagementInformationSystem">
13    <xs:annotation>
14      <xs:documentation>This XML schema definition describes an overall Service
        Level Management information system covering data models for all SLM
        information artifacts.</xs:documentation>
15    </xs:annotation>
16    <xs:complexType>
17      <xs:all>
18        <xs:element name="ServicePortfolio" type="type_ServicePortfolio"/>
19        <xs:element name="CustomerDatabase" type="type_CustomerDatabase"/>
20        <xs:element name="SDPDatabase" type="type_SDPDatabase"/>
21        <xs:element name="SLPDatabase" type="type_SLPDatabase"/>
22        <xs:element name="SLADatabase" type="type_SLADatabase"/>
23        <xs:element name="SADatabase" type="type_SADatabase"/>
24        <xs:element name="MeasurementDatabase" type="type_MeasurementDatabase"/>
25        <xs:element name="ReportDatabase" type="type_ReportDatabase"/>
26        <xs:element name="ConflictDatabase" type="type_ConflictDatabase"/>
27      </xs:all>
28    </xs:complexType>
```

```

29 <!-- KEY DEFINITIONS -->
30 <!-- ServicePortfolio -->
31 <xs:key name="serviceIdentification">
32   <xs:selector xpath="./ServicePortfolio/Service"/>
33   <xs:field xpath="@id"/>
34 </xs:key>
35 <xs:key name="serviceDeliveryParameterIdentification">
36   <xs:selector xpath="./ServicePortfolio/Service/serviceDeliveryParameters/
37     serviceDeliveryParameter"/>
38   <xs:field xpath="@id"/>
39 </xs:key>
40 <xs:key name="serviceLevelParameterIdentification">
41   <xs:selector xpath="./ServicePortfolio/Service/serviceLevelParameters/
42     serviceLevelParameter"/>
43   <xs:field xpath="@id"/>
44 </xs:key>
45 <xs:key name="serviceLevelMetricIdentification">
46   <xs:selector xpath="./ServicePortfolio/Service/serviceLevelMetrics/
47     serviceLevelMetric"/>
48   <xs:field xpath="@id"/>
49 </xs:key>
50 <xs:key name="ServiceCatalogIdentification">
51   <xs:selector xpath="./ServicePortfolio/ServiceCatalog"/>
52   <xs:field xpath="@id"/>
53 </xs:key>
54 <xs:key name="serviceCatalogServiceIdentification">
55   <xs:selector xpath="./ServicePortfolio/ServiceCatalog/
56     referenceToServiceFromServiceCatalog"/>
57   <xs:field xpath="@id"/>
58 </xs:key>
59 <!-- CustomerDatabase -->
60 <xs:key name="customerIdentification">
61   <xs:selector xpath="./CustomerDatabase/Customer"/>
62   <xs:field xpath="@id"/>
63 </xs:key>
64 <xs:key name="customerContactPointIdentification">
65   <xs:selector xpath="./Customer/customerContactInformation/
66     customerContactPoint"/>
67   <xs:field xpath="@id"/>
68 </xs:key>
69 <!-- SDPDatabase -->
70 <xs:key name="SDPIdentification">
71   <xs:selector xpath="./SDPDatabase/ServiceDeliveryParty"/>
72   <xs:field xpath="@id"/>
73 </xs:key>
74 <!-- SLPDatabase -->
75 <xs:key name="SLPIdentification">
76   <xs:selector xpath="./SLPDatabase/SupportLevelProfile"/>
77   <xs:field xpath="@id"/>
78 </xs:key>
79 <!-- SLADatabase -->
80 <xs:key name="SLAIdentification">
81   <xs:selector xpath="./SLADatabase/ServiceLevelAgreement"/>
82   <xs:field xpath="@id"/>
83 </xs:key>
84 <!-- SADatabase -->
85 <xs:key name="SAIdentification">
86   <xs:selector xpath="./SADatabase/SupportiveAgreement"/>
87   <xs:field xpath="@id"/>
88 </xs:key>
89 <!-- MeasurementDatabase -->
90 <xs:key name="measurementSpecificationIdentification">
91   <xs:selector xpath="./MeasurementDatabase/MeasurementSpecification"/>

```



```

87     <xs:field xpath="@id"/>
</xs:key>
89 <xs:key name="measurementItemIdentification">
  <xs:selector xpath="./MeasurementDatabase/MeasurementSpecification/
    measurementItem"/>
91   <xs:field xpath="@id"/>
</xs:key>
93 <xs:key name="measurementMetricIdentification">
  <xs:selector xpath="./MeasurementDatabase/MeasurementSpecification/
    measurementItem/measurementMetric"/>
95   <xs:field xpath="@id"/>
</xs:key>
97 <xs:key name="measurementResultsIdentification">
  <xs:selector xpath="./MeasurementDatabase/MeasurementResults"/>
99   <xs:field xpath="@id"/>
</xs:key>
101 <!-- ReportDatabase -->
<xs:key name="internalReportSpecificationIdentification">
103   <xs:selector xpath="./ReportDatabase/InternalReportSpecification"/>
  <xs:field xpath="@id"/>
105 </xs:key>
<xs:key name="externalReportSpecificationIdentification">
107   <xs:selector xpath="./ReportDatabase/ExternalReportSpecification"/>
  <xs:field xpath="@id"/>
109 </xs:key>
<xs:key name="internalReportIdentification">
111   <xs:selector xpath="./ReportDatabase/InternalReport"/>
  <xs:field xpath="@id"/>
113 </xs:key>
<xs:key name="externalReportIdentification">
115   <xs:selector xpath="./ReportDatabase/ExternalReport"/>
  <xs:field xpath="@id"/>
117 </xs:key>
<xs:key name="SLAViolationNotificationIdentification">
119   <xs:selector xpath="./ReportDatabase/SLAViolationNotificationIdentification
    "/>
  <xs:field xpath="@id"/>
121 </xs:key>
<!-- ConflictDatabase -->
123 <xs:key name="agreementConflictIdentification">
  <xs:selector xpath="./ConflictDatabase/AgreementConflict"/>
125   <xs:field xpath="@id"/>
</xs:key>
127 <!-- KEYREF DEFINITIONS -->
<!-- From ServicePortfolio -->
129 <xs:keyref name="referenceToCustomerFromServiceCatalog" refer="
  customerIdentification">
  <xs:selector xpath="./ServicePortfolio/ServiceCatalog/
    referenceToCustomerFromServiceCatalog"/>
131   <xs:field xpath="@idref"/>
</xs:keyref>
133 <xs:keyref name="referenceToServiceFromServiceCatalog" refer="
  serviceIdentification">
  <xs:selector xpath="./ServicePortfolio/ServiceCatalog/
    referenceToServiceFromServiceCatalog"/>
135   <xs:field xpath="@idref"/>
</xs:keyref>
137 <!-- From CustomerDatabase -->
<xs:keyref name="referenceToCustomerContactPointFromCustomerRoleAssignment"
  refer="customerContactPointIdentification">
139   <xs:selector xpath="./Customer/customerRoleAssignments/
    customerRoleAssignment/referenceToCustomerContactPoint"/>
  <xs:field xpath="@idref"/>

```

```

141     </xs:keyref>
142     <!-- From SLADatabase -->
143     <xs:keyref name="referenceToServiceFromSLA" refer="
        serviceCatalogServiceIdentification">
        <xs:selector xpath="./SLADatabase/ServiceLevelAgreement/
            referenceToServiceFromSLA"/>
144     <xs:field xpath="@idref"/>
145     </xs:keyref>
146     <xs:keyref name="referenceToCustomerFromSLA" refer="customerIdentification">
        <xs:selector xpath="./SLADatabase/ServiceLevelAgreement/
            referenceToCustomerFromSLA"/>
147     <xs:field xpath="@idref"/>
148     </xs:keyref>
149     <xs:keyref name="referenceToSLPFromSLA" refer="SLPIdentification">
        <xs:selector xpath="./SLADatabase/ServiceLevelAgreement/
            referenceToSLPFromSLA"/>
150     <xs:field xpath="@idref"/>
151     </xs:keyref>
152     <xs:keyref name="referenceToServiceDeliveryParameterFromSLA" refer="
        serviceDeliveryParameterIdentification">
        <xs:selector xpath="./SLADatabase/ServiceLevelAgreement/
            serviceDeliveryTargets/serviceDeliveryTarget/
            referenceToServiceDeliveryParameterFromSLA"/>
153     <xs:field xpath="@idref"/>
154     </xs:keyref>
155     <xs:keyref name="referenceToServiceLevelParameterFromSLA" refer="
        serviceLevelParameterIdentification">
        <xs:selector xpath="./SLADatabase/ServiceLevelAgreement/serviceLevelTargets
            /serviceLevelTarget/referenceToServiceLevelParameterFromSLA"/>
156     <xs:field xpath="@idref"/>
157     </xs:keyref>
158     <!-- From SADatabase -->
159     <xs:keyref name="referenceToServiceDeliveryPartyFromSupportiveAgreement"
        refer="SDPIdentification">
        <xs:selector xpath="./SADatabase/SupportiveAgreement/
            referenceToServiceDeliveryPartyFromSupportiveAgreement"/>
160     <xs:field xpath="@idref"/>
161     </xs:keyref>
162     <!-- From MeasurementDatabase -->
163     <xs:keyref name="referenceToServiceFromMeasurementSpecification" refer="
        serviceIdentification">
        <xs:selector xpath="./MeasurementDatabase/MeasurementSpecification/
            referenceToServiceFromMeasurementSpecification"/>
164     <xs:field xpath="@idref"/>
165     </xs:keyref>
166     <xs:keyref name="referenceToServiceLevelMetricFromMeasurementMetric" refer="
        serviceLevelMetricIdentification">
        <xs:selector xpath="./MeasurementDatabase/MeasurementSpecification/
            measurementItem/measurementMetric/
            eferenceToServiceLevelMetricFromMeasurementMetric"/>
167     <xs:field xpath="@idref"/>
168     </xs:keyref>
169     <xs:keyref name="referenceToMeasurementSpecificationFromMeasurementResults"
        refer="measurementSpecificationIdentification">
        <xs:selector xpath="./MeasurementDatabase/MeasurementResults/
            referenceToMeasurementSpecificationFromMeasurementResults"/>
170     <xs:field xpath="@idref"/>
171     </xs:keyref>
172     <!-- From ReportDatabase -->
173     <xs:keyref name="referenceToServiceFromInternalReportSpecification" refer="
        serviceIdentification">
        <xs:selector xpath="./ReportDatabase/InternalReportSpecification/
            referenceToServiceFromInternalReportSpecification"/>
174     </xs:keyref>

```

```

185     <xs:field xpath="@idref"/>
186 </xs:keyref>
187 <xs:keyref name="referenceToSLAFromExternalReportSpecification" refer="
    SLAIdentification">
188     <xs:selector xpath="./ReportDatabase/ExternalReportSpecification/
    referenceToSLAFromExternalReportSpecification"/>
189     <xs:field xpath="@idref"/>
190 </xs:keyref>
191 <xs:keyref name="referenceToMeasurementMetricFromInternalReportMetrics" refer
    ="measurementMetricIdentification">
192     <xs:selector xpath="./ReportDatabase/InternalReportSpecification/
    internalReportMetrics/
    referenceToMeasurementMetricFromInternalReportMetrics"/>
193     <xs:field xpath="@idref"/>
194 </xs:keyref>
195 <xs:keyref name="referenceToMeasurementMetricFromExternalReportMetrics" refer
    ="measurementMetricIdentification">
196     <xs:selector xpath="./ReportDatabase/ExternalReportSpecification/
    externalReportMetrics/
    referenceToMeasurementMetricFromExternalReportMetrics"/>
197     <xs:field xpath="@idref"/>
198 </xs:keyref>
199 <xs:keyref name="referenceToInternalReportSpecificationFromInternalReport"
    refer="internalReportSpecificationIdentification">
200     <xs:selector xpath="./ReportDatabase/InternalReport/
    referenceToInternalReportSpecificationFromInternalReport"/>
201     <xs:field xpath="@idref"/>
202 </xs:keyref>
203 <xs:keyref name="referenceToExternalReportSpecificationFromExternalReport"
    refer="externalReportSpecificationIdentification">
204     <xs:selector xpath="./ReportDatabase/ExternalReport/
    referenceToExternalReportSpecificationFromExternalReport"/>
205     <xs:field xpath="@idref"/>
206 </xs:keyref>
207 <xs:keyref name="referenceToMeasurementResultsFromInternalReport" refer="
    measurementResultsIdentification">
208     <xs:selector xpath="./ReportDatabase/InternalReport/
    referenceToMeasurementResultsFromInternalReport"/>
209     <xs:field xpath="@idref"/>
210 </xs:keyref>
211 <xs:keyref name="referenceToMeasurementResultsFromExternalReport" refer="
    measurementResultsIdentification">
212     <xs:selector xpath="./ReportDatabase/ExternalReport/
    referenceToMeasurementResultsFromExternalReport"/>
213     <xs:field xpath="@idref"/>
214 </xs:keyref>
215 <xs:keyref name="referenceToMeasurementResultsFromSLAViolationNotification"
    refer="measurementResultsIdentification">
216     <xs:selector xpath="./ReportDatabase/SLAViolationNotification/
    referenceToMeasurementResultsFromSLAViolationNotification"/>
217     <xs:field xpath="@idref"/>
218 </xs:keyref>
219 <xs:keyref name="referenceToSLAFromSLAViolationNotification" refer="
    SLAIdentification">
220     <xs:selector xpath="./ReportDatabase/SLAViolationNotification/
    referenceToSLAFromSLAViolationNotification"/>
221     <xs:field xpath="@idref"/>
222 </xs:keyref>
223 <!-- From ConflictDatabase -->
224 <xs:keyref name="referenceToSLAFromAgreementConflict" refer="
    SLAIdentification">
225     <xs:selector xpath="./ConflictDatabaseAgreementConflict/
    referenceToAgreement/referenceToSLAFromAgreementConflict"/>

```

```
225     <xs:field xpath="@idref"/>
226   </xs:keyref>
227   <xs:keyref name="referenceToSAFromAgreementConflict" refer="SAIdentification"
228     >
229     <xs:selector xpath="./ConflictDatabaseAgreementConflict/
230       referenceToAgreement/referenceToSAFromAgreementConflict"/>
231   </xs:keyref>
232 </xs:element>
233 </xs:schema>
```

Listing A.1: ServiceLevelManagementInformationSystem.xsd

A.2 CommonTypes

```
<?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
3   <xs:annotation>
4     <xs:documentation>Global type definitions</xs:documentation>
5   </xs:annotation>
6   <xs:complexType name="type_selectFromList">
7     <xs:sequence>
8       <xs:element name="listElement" type="xs:string" minOccurs="2" maxOccurs="
9         unbounded"/>
10    </xs:sequence>
11  </xs:complexType>
12  <xs:complexType name="type_selectFromRange">
13    <xs:sequence>
14      <xs:element name="minimumInclusiveValue" type="xs:float" minOccurs="0"/>
15      <xs:element name="maximumInclusiveValue" type="xs:float" minOccurs="0"/>
16    </xs:sequence>
17  </xs:complexType>
18  <xs:complexType name="type_parameterDomain">
19    <xs:choice>
20      <xs:element name="selectFromList" type="type_selectFromList"/>
21      <xs:element name="selectFromRange" type="type_selectFromRange"/>
22      <xs:element name="description" type="xs:string"/>
23    </xs:choice>
24  </xs:complexType>
</xs:schema>
```

Listing A.2: CommonTypes.xsd

A.3 ServicePortfolioDatabase

```
<?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
3   <xs:annotation>
4     <xs:documentation>An XML schema definition for the ITSM/SLM information
5     artifact Service of type record</xs:documentation>
```

```

6      </xs:annotation>
7      <xs:include schemaLocation="CommonTypes.xsd"/>
8      <xs:complexType name="type_serviceDeliveryParameter">
9          <xs:sequence>
10             <xs:element name="name" type="xs:string"/>
11             <xs:element name="parameterDomain" type="type_parameterDomain"/>
12         </xs:sequence>
13         <xs:attribute name="id" type="xs:string" use="required"/>
14     </xs:complexType>
15     <xs:complexType name="type_serviceLevelParameter">
16         <xs:sequence>
17             <xs:element name="name" type="xs:string"/>
18             <xs:element name="parameterDomain" type="type_parameterDomain"/>
19         </xs:sequence>
20         <xs:attribute name="id" type="xs:string" use="required"/>
21     </xs:complexType>
22     <xs:complexType name="type_serviceRequirements">
23         <xs:sequence>
24             <xs:element name="serviceRequirement" type="type_serviceRequirement"
25                 maxOccurs="unbounded"/>
26         </xs:sequence>
27     </xs:complexType>
28     <xs:complexType name="type_serviceRequirement">
29         <xs:sequence>
30             <xs:element name="description" type="xs:string"/>
31             <xs:element name="priority" type="xs:decimal"/>
32         </xs:sequence>
33     </xs:complexType>
34     <xs:complexType name="type_serviceSpecification">
35         <xs:sequence>
36             <xs:element name="serviceUsageFunctionality" type="xs:string"/>
37             <xs:element name="serviceAccessPointSpecification" type="xs:string"/>
38             <xs:element name="serviceClientSpecification" type="xs:string"/>
39             <xs:element name="serviceManagementFunctionality" type="xs:string"/>
40             <xs:element name="serviceManagementAccessPointSpecification" type="
41                 xs:string"/>
42             <xs:element name="serviceManagementClientSpecification" type="xs:string"/>
43             <xs:element name="serviceResources" type="xs:string"/>
44             <xs:element name="serviceDependencies" type="type_serviceDependencies"/>
45         </xs:sequence>
46     </xs:complexType>
47     <xs:complexType name="type_serviceDependencies">
48         <xs:sequence>
49             <xs:element name="dependency" type="xs:string" minOccurs="0" maxOccurs="
50                 unbounded"/>
51         </xs:sequence>
52     </xs:complexType>
53     <xs:complexType name="type_serviceDeliveryParameters">
54         <xs:sequence>
55             <xs:element name="serviceDeliveryParameter" type="
56                 type_serviceDeliveryParameter" minOccurs="0" maxOccurs="unbounded"/>
57         </xs:sequence>
58     </xs:complexType>
59     <xs:complexType name="type_serviceLevelParameters">
60         <xs:sequence>
61             <xs:element name="serviceLevelParameter" type="type_serviceLevelParameter"
62                 minOccurs="0" maxOccurs="unbounded"/>
63         </xs:sequence>
64     </xs:complexType>
65     <xs:complexType name="type_serviceLevelMetrics">
66         <xs:sequence>
67             <xs:element name="serviceLevelMetric" type="type_serviceLevelMetric"
68                 minOccurs="0" maxOccurs="unbounded"/>

```

```

62     </xs:sequence>
</xs:complexType>
64 <xs:complexType name="type_serviceLevelMetric">
  <xs:sequence>
66     <xs:element name="name" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
68     <xs:element name="measurementRule" type="xs:string"/>
  </xs:sequence>
70   <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
72 <xs:simpleType name="type_serviceStatus">
  <xs:restriction base="xs:string">
74     <xs:enumeration value="planned-requirements"/>
    <xs:enumeration value="planned-design"/>
76     <xs:enumeration value="active-ready"/>
    <xs:enumeration value="active-operating"/>
78     <xs:enumeration value="inactive-maintenance"/>
    <xs:enumeration value="inactive-withdrawn"/>
80   </xs:restriction>
</xs:simpleType>
82 <xs:complexType name="type_Service">
  <xs:sequence>
84     <xs:element name="name" type="xs:string"/>
    <xs:element name="serviceDescription" type="xs:string"/>
86     <xs:element name="serviceRequirements" type="type_serviceRequirements"/>
    <xs:element name="serviceSpecification" type="type_serviceSpecification"/>
88     <xs:element name="serviceDeliveryParameters" type="
      type_serviceDeliveryParameters"/>
    <xs:element name="serviceLevelParameters" type="type_serviceLevelParameters
      "/>
90     <xs:element name="serviceLevelMetrics" type="type_serviceLevelMetrics"/>
    <xs:element name="serviceStatus" type="type_serviceStatus"/>
92     <xs:element name="serviceHistory" type="xs:string"/>
  </xs:sequence>
94   <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="date" type="xs:date" use="required"/>
96 </xs:complexType>
<xs:complexType name="type_referenceToCustomerFromServiceCatalog">
98   <xs:attribute name="idref" type="xs:string" use="required"/>
</xs:complexType>
100 <xs:complexType name="type_referenceToServiceFromServiceCatalog">
  <xs:attribute name="idref" type="xs:string" use="required"/>
102   <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
104 <xs:complexType name="type_serviceCatalogValidity">
  <xs:sequence>
106     <xs:element name="serviceCatalogValidFrom" type="xs:dateTime"/>
    <xs:element name="serviceCatalogValidThrough" type="xs:dateTime"/>
108   </xs:sequence>
</xs:complexType>
110 <xs:complexType name="type_serviceCatalogHistoryItem">
  <xs:sequence>
112     <xs:element name="date" type="xs:dateTime"/>
    <xs:element name="serviceCatalogArchivalCopy" type="type_ServiceCatalog"/>
114   </xs:sequence>
</xs:complexType>
116 <xs:complexType name="type_serviceCatalogHistory">
  <xs:sequence>
118     <xs:element name="serviceCatalogHistoryItem" type="
      type_serviceCatalogHistoryItem" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
120 </xs:complexType>
<xs:complexType name="type_ServiceCatalog">

```

```

122 <xs:sequence>
    <xs:element name="referenceToCustomerFromServiceCatalog" type="
        type_referenceToCustomerFromServiceCatalog" minOccurs="0" maxOccurs="
124 <xs:element name="referenceToServiceFromServiceCatalog" type="
        type_referenceToServiceFromServiceCatalog" minOccurs="0" maxOccurs="
        unbounded"/>
    <xs:element name="serviceCatalogValidity" type="type_serviceCatalogValidity
        "/>
126 <xs:element name="serviceCatalogHistory" type="type_serviceCatalogHistory"/
        >
    </xs:sequence>
128 <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="date" type="xs:date"/>
130 </xs:complexType>
<xs:complexType name="type_ServicePortfolio">
132 <xs:sequence>
    <xs:element name="Service" type="type_Service" minOccurs="0" maxOccurs="
        unbounded"/>
134 <xs:element name="ServiceCatalog" type="type_ServiceCatalog" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
136 </xs:complexType>
</xs:schema>

```

Listing A.3: ServicePortfolio.xsd

A.4 CustomerDatabase

```

1 <?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
    qualified" attributeFormDefault="unqualified">
3 <xs:annotation>
    <xs:documentation>An XML schema definition for the ITSM/SLM information
        artifact Customer of type record</xs:documentation>
5 </xs:annotation>
<xs:complexType name="type_customerContactPoint">
7 <xs:sequence>
    <xs:element name="customerContactPointEmail" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
9 <xs:element name="customerContactPointPhone" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
    <xs:element name="customerContactPointAddress" type="xs:string" minOccurs="
        0" maxOccurs="unbounded"/>
11 <xs:element name="customerContactPointOthers" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
13 <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
15 <xs:complexType name="
    type_referenceToCustomerContactPointFromCustomerRoleAssignment">
    <xs:attribute name="idref" type="xs:string" use="required"/>
17 </xs:complexType>
<xs:complexType name="type_customerContactInformation">
19 <xs:sequence>
    <xs:element name="customerContactPoint" type="type_customerContactPoint"
        maxOccurs="unbounded"/>
21 </xs:sequence>

```

```

23 </xs:complexType>
24 <xs:complexType name="type_customerRoleAssignment">
25   <xs:sequence>
26     <xs:element name="role" type="xs:string"/>
27     <xs:element name="referenceToCustomerContactPointFromCustomerRoleAssignment"
28       type="type_referenceToCustomerContactPointFromCustomerRoleAssignment"
29       maxOccurs="unbounded"/>
30   </xs:sequence>
31 </xs:complexType>
32 <xs:complexType name="type_customerRoleAssignments">
33   <xs:sequence>
34     <xs:element name="customerRoleAssignment" type="type_customerRoleAssignment"
35       maxOccurs="unbounded"/>
36   </xs:sequence>
37 </xs:complexType>
38 <xs:complexType name="type_customerServiceInterface">
39   <xs:sequence>
40     <xs:element name="customerServiceInterfaceClient" type="xs:string"/>
41     <xs:element name="customerServiceInterfaceConnectivity" type="xs:string"/>
42   </xs:sequence>
43   <xs:attribute name="id" type="xs:string" use="required"/>
44 </xs:complexType>
45 <xs:complexType name="type_customerServiceInterfaces">
46   <xs:sequence>
47     <xs:element name="customerServiceInterface" type="
48       type_customerServiceInterface" maxOccurs="unbounded"/>
49   </xs:sequence>
50 </xs:complexType>
51 <xs:complexType name="type_Customer">
52   <xs:sequence>
53     <xs:element name="name" type="xs:string"/>
54     <xs:element name="customerContactInformation" type="
55       type_customerContactInformation"/>
56     <xs:element name="customerRoleAssignments" type="
57       type_customerRoleAssignments"/>
58     <xs:element name="customerServiceInterfaces" type="
59       type_customerServiceInterfaces"/>
60   </xs:sequence>
61   <xs:attribute name="id" type="xs:string" use="required"/>
62   <xs:attribute name="date" type="xs:date"/>
63 </xs:complexType>
64 <xs:complexType name="type_CustomerDatabase">
65   <xs:sequence>
66     <xs:element name="Customer" type="type_Customer" minOccurs="0" maxOccurs="
67       unbounded"/>
68   </xs:sequence>
69 </xs:complexType>
70 </xs:schema>

```

Listing A.4: CustomerDatabase.xsd

A.5 ServiceDeliveryPartyDatabase

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
3   qualified" attributeFormDefault="unqualified">
4   <xs:annotation>

```



```

    <xs:documentation>An XML schema definition for the ITSM/SLM information
      artifact ServiceDeliveryPartyDatabase of type database</xs:documentation
    >
5  </xs:annotation>
    <xs:simpleType name="type_SDPType">
7    <xs:restriction base="xs:string">
      <xs:enumeration value="internal delivery party"/>
9      <xs:enumeration value="external supplier"/>
    </xs:restriction>
11  </xs:simpleType>
    <xs:complexType name="type_SDPContactPoint">
13    <xs:sequence>
      <xs:element name="SDPContactPointEmail" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
15      <xs:element name="SDPContactPointPhone" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:element name="SDPContactPointAddress" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
17      <xs:element name="SDPContactPointOthers" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
19    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
21  <xs:complexType name="type_referenceToSDPContactPointFromSDPRoleAssignment">
    <xs:attribute name="idref" type="xs:string" use="required"/>
23  </xs:complexType>
    <xs:complexType name="type_SDPContactInformation">
25    <xs:sequence>
      <xs:element name="SDPContactPoint" type="type_SDPContactPoint" maxOccurs="
        unbounded"/>
27    </xs:sequence>
  </xs:complexType>
29  <xs:complexType name="type_SDPRoleAssignment">
    <xs:sequence>
31      <xs:element name="role" type="xs:string"/>
      <xs:element name="referenceToSDPContactPointFromSDPRoleAssignment" type="
        type_referenceToSDPContactPointFromSDPRoleAssignment" maxOccurs="
        unbounded"/>
33    </xs:sequence>
  </xs:complexType>
35  <xs:complexType name="type_SDPRoleAssignments">
    <xs:sequence>
37      <xs:element name="SDPRoleAssignment" type="type_SDPRoleAssignment"
        maxOccurs="unbounded"/>
    </xs:sequence>
39  </xs:complexType>
    <xs:complexType name="type_SDPsSkillSet">
41    <xs:sequence>
      <xs:element name="SDPSkill" type="xs:string" minOccurs="0" maxOccurs="
        unbounded"/>
43    </xs:sequence>
  </xs:complexType>
45  <xs:complexType name="type_SDPServiceCatalog">
    <xs:sequence>
47      <xs:element name="SDPServiceCatalogLocation" type="xs:string"/>
    </xs:sequence>
49  </xs:complexType>
    <xs:complexType name="type_ServiceDeliveryParty">
51    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
53      <xs:element name="SDPType" type="type_SDPType"/>
      <xs:element name="SDPContactInformation" type="type_SDPContactInformation"/
    >
  >

```

```

55     <xs:element name="SDPRoleAssignments" type="type_SDPRoleAssignments"/>
56     <xs:element name="SDPSkillSet" type="type_SDPSkillSet"/>
57     <xs:element name="SDPServiceCatalog" type="type_SDPServiceCatalog"/>
58   </xs:sequence>
59   <xs:attribute name="id" type="xs:string" use="required"/>
60   <xs:attribute name="date" type="xs:date"/>
61 </xs:complexType>
62 <xs:complexType name="type_SDPDatabase">
63   <xs:sequence>
64     <xs:element name="ServiceDeliveryParty" type="type_ServiceDeliveryParty"
65       minOccurs="0" maxOccurs="unbounded">
66       <xs:key name="SDPContactPointIdentification">
67         <xs:selector xpath="./ServiceDeliveryParty/SDPContactInformation/
68           SDPContactPoint"/>
69         <xs:field xpath="@id"/>
70       </xs:key>
71       <xs:keyref name="SDPContactPointReference" refer="
72         SDPContactPointIdentification">
73         <xs:selector xpath="./ServiceDeliveryParty/SDPRoleAssignments/
74           SDPRoleAssignment/referenceToSDPContactPointFromSDPRoleAssignment"
75         />
76         <xs:field xpath="@idref"/>
77       </xs:keyref>
78     </xs:element>
79   </xs:sequence>
80 </xs:complexType>
81 </xs:schema>

```

Listing A.5: SDPDatabase.xsd

A.6 SupportLevelProfileDatabase

```

<?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
3   <xs:annotation>
4     <xs:documentation>An XML schema definition for the SLM information artifact
      Support Level Profile of type record</xs:documentation>
5   </xs:annotation>
6   <xs:simpleType name="type_channelOption">
7     <xs:restriction base="xs:string">
8       <xs:enumeration value="email"/>
9       <xs:enumeration value="webform"/>
10      <xs:enumeration value="phone"/>
11      <xs:enumeration value="physical"/>
12    </xs:restriction>
13  </xs:simpleType>
14  <xs:simpleType name="type_weekday">
15    <xs:restriction base="xs:string">
16      <xs:enumeration value="mon"/>
17      <xs:enumeration value="tue"/>
18      <xs:enumeration value="wed"/>
19      <xs:enumeration value="thu"/>
20      <xs:enumeration value="fri"/>
21      <xs:enumeration value="sat"/>
22      <xs:enumeration value="sun"/>
23    </xs:restriction>
24  </xs:simpleType>

```

```

26 <xs:complexType name="type_supportTimeException">
    <xs:sequence>
28       <xs:element name="startDateTime" type="xs:dateTime"/>
       <xs:element name="endDateTime" type="xs:dateTime"/>
30     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_supportChannel">
32     <xs:sequence>
34       <xs:element name="channelOption" type="type_channelOption"/>
       <xs:element name="contactInfo" type="xs:string"/>
36     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_supportTime">
38     <xs:sequence>
40       <xs:element name="weekday" type="type_weekday"/>
       <xs:element name="startTime" type="xs:time"/>
42       <xs:element name="endTime" type="xs:time"/>
     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_supportChannels">
44     <xs:sequence>
46       <xs:element name="channel" type="type_supportChannel" maxOccurs="unbounded"
         />
     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_supportTimes">
50     <xs:sequence>
52       <xs:element name="supportTime" type="type_supportTime" maxOccurs="unbounded"
         />
       <xs:element name="supportTimeExceptionalInclusion" type="
         type_supportTimeException" minOccurs="0" maxOccurs="unbounded"/>
       <xs:element name="supportTimeExceptionalExclusion" type="
         type_supportTimeException" minOccurs="0" maxOccurs="unbounded"/>
54     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_maxResponseTime">
56     <xs:sequence>
58       <xs:element name="timeFrame" type="xs:duration"/>
       <xs:element name="conditions" type="xs:string"/>
60     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_maxRecoveryTime">
62     <xs:sequence>
64       <xs:element name="timeFrame" type="xs:duration"/>
       <xs:element name="conditions" type="xs:string"/>
66     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_maxResponseTimes">
68     <xs:sequence>
70       <xs:element name="maxResponseTime" type="type_maxResponseTime" minOccurs="0"
         " maxOccurs="unbounded"/>
     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_maxRecoveryTimes">
72     <xs:sequence>
74       <xs:element name="maxRecoveryTime" type="type_maxRecoveryTime" minOccurs="0"
         " maxOccurs="unbounded"/>
76     </xs:sequence>
  </xs:complexType>
  <xs:complexType name="type_SupportLevelProfile">
78     <xs:sequence>
80       <xs:element name="name" type="xs:string"/>
       <xs:element name="supportChannels" type="type_supportChannels"/>

```

```

82     <xs:element name="supportTimes" type="type_supportTimes"/>
      <xs:element name="maxResponseTimes" type="type_maxResponseTimes"/>
84     <xs:element name="maxRecoveryTimes" type="type_maxRecoveryTimes"/>
      </xs:sequence>
86     <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:attribute name="date" type="xs:date"/>
88   </xs:complexType>
   <xs:complexType name="type_SLPDatabase">
90     <xs:sequence>
       <xs:element name="SupportLevelProfile" type="type_SupportLevelProfile"
          minOccurs="0" maxOccurs="unbounded"/>
92     </xs:sequence>
   </xs:complexType>
94 </xs:schema>

```

Listing A.6: SLPDatabase.xsd

A.7 ServiceLevelAgreementDatabase

```

<?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
  <xs:annotation>
4    <xs:documentation>An XML schema definition for the SLM information artifact
      Service Level Agreement of type documentation</xs:documentation>
  </xs:annotation>
6   <xs:complexType name="type_referenceToServiceFromSLA">
     <xs:attribute name="idref" type="xs:string" use="required"/>
8   </xs:complexType>
   <xs:complexType name="type_referenceToCustomerFromSLA">
10    <xs:attribute name="idref" type="xs:string" use="required"/>
   </xs:complexType>
12   <xs:complexType name="type_referenceToSLPFromSLA">
     <xs:attribute name="idref" type="xs:string" use="required"/>
   </xs:complexType>
14   <xs:complexType name="type_referenceToServiceDeliveryParameterFromSLA">
     <xs:attribute name="idref" type="xs:string" use="required"/>
16   </xs:complexType>
   <xs:complexType name="type_serviceDeliveryTarget">
18     <xs:sequence>
20       <xs:element name="referenceToServiceDeliveryParameterFromSLA" type="
          type_referenceToServiceDeliveryParameterFromSLA"/>
       <xs:element name="serviceDeliveryWarranty" type="xs:string"/>
22     </xs:sequence>
   </xs:complexType>
24   <xs:complexType name="type_serviceDeliveryTargets">
     <xs:sequence>
26       <xs:element name="serviceDeliveryTarget" type="type_serviceDeliveryTarget"
          minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
   </xs:complexType>
28   <xs:complexType name="type_referenceToServiceLevelParameterFromSLA">
     <xs:attribute name="idref" type="xs:string" use="required"/>
30   </xs:complexType>
   <xs:complexType name="type_serviceLevelTarget">
32     <xs:sequence>
34       <xs:element name="referenceToServiceLevelParameterFromSLA" type="
          type_referenceToServiceLevelParameterFromSLA"/>

```

```

36     <xs:element name="serviceLevelWarranty" type="xs:string"/>
37   </xs:sequence>
38 </xs:complexType>
39 <xs:complexType name="type_serviceLevelTargets">
40   <xs:sequence>
41     <xs:element name="serviceLevelTarget" type="type_serviceLevelTarget"
42       minOccurs="0" maxOccurs="unbounded"/>
43   </xs:sequence>
44 </xs:complexType>
45 <xs:complexType name="type_serviceDegradationModelConditionalClause">
46   <xs:sequence>
47     <xs:element name="degradationSubject" type="xs:string"/>
48     <xs:element name="degradationManner" type="xs:string"/>
49     <xs:element name="degradationTime" type="xs:string"/>
50   </xs:sequence>
51 </xs:complexType>
52 <xs:simpleType name="type_degradationDegree">
53   <xs:restriction base="xs:decimal">
54     <xs:minInclusive value="1"/>
55     <xs:maxInclusive value="100"/>
56   </xs:restriction>
57 </xs:simpleType>
58 <xs:complexType name="type_serviceDegradationModel">
59   <xs:sequence>
60     <xs:element name="serviceDegradationDegree" type="type_degradationDegree"/>
61     <xs:element name="serviceDegradationModelConditionalClause" type="
62       type_serviceDegradationModelConditionalClause" maxOccurs="unbounded"/>
63     <!-- models: AND -->
64   </xs:sequence>
65 </xs:complexType>
66 <xs:complexType name="type_serviceDegradationDefinition">
67   <xs:sequence>
68     <xs:element name="serviceDegradationModel" type="
69       type_serviceDegradationModel" maxOccurs="unbounded"/>
70     <!-- models: OR -->
71   </xs:sequence>
72 </xs:complexType>
73 <xs:simpleType name="type_deliveryConditionType">
74   <xs:restriction base="xs:string">
75     <xs:enumeration value="delivery"/>
76     <xs:enumeration value="transition"/>
77     <xs:enumeration value="resolution"/>
78     <xs:enumeration value="charging"/>
79   </xs:restriction>
80 </xs:simpleType>
81 <xs:complexType name="type_deliveryCondition">
82   <xs:sequence>
83     <xs:element name="deliveryConditionType" type="type_deliveryConditionType"/>
84     <xs:element name="deliveryConditionSpecification" type="xs:string"/>
85   </xs:sequence>
86 </xs:complexType>
87 <xs:complexType name="type_deliveryConditions">
88   <xs:sequence>
89     <xs:element name="deliveryCondition" type="type_deliveryCondition"
90       minOccurs="0" maxOccurs="unbounded"/>
91   </xs:sequence>
92 </xs:complexType>
93 <xs:complexType name="type_SLAHistoryItem">
94   <xs:sequence>
95     <xs:element name="date" type="xs:dateTime"/>
96     <xs:element name="SLAArchivalCopy" type="type_ServiceLevelAgreement"/>
97   </xs:sequence>

```

```

94 </xs:complexType>
<xs:complexType name="type_SLAHistory">
96   <xs:sequence>
     <xs:element name="SLAHistoryItem" type="type_SLAHistoryItem" minOccurs="0"
       maxOccurs="unbounded"/>
   </xs:sequence>
98 </xs:complexType>
<xs:complexType name="type_ServiceLevelAgreement">
100   <xs:sequence>
     <xs:element name="referenceToServiceFromSLA" type="
       type_referenceToServiceFromSLA"/>
102     <xs:element name="referenceToCustomerFromSLA" type="
       type_referenceToCustomerFromSLA"/>
     <xs:element name="referenceToSLPFromSLA" type="type_referenceToSLPFromSLA"/>
104     <xs:element name="serviceDeliveryTargets" type="type_serviceDeliveryTargets"
       />
     <xs:element name="serviceLevelTargets" type="type_serviceLevelTargets"/>
106     <xs:element name="serviceDegradationDefinition" type="
       type_serviceDegradationDefinition"/>
     <xs:element name="deliveryConditions" type="type_deliveryConditions"/>
108     <xs:element name="SLAHistory" type="type_SLAHistory"/>
   </xs:sequence>
110   <xs:attribute name="id" type="xs:string" use="required"/>
   <xs:attribute name="date" type="xs:date"/>
112 </xs:complexType>
<xs:complexType name="type_SLADatabase">
114   <xs:sequence>
     <xs:element name="ServiceLevelAgreement" type="type_ServiceLevelAgreement"
       minOccurs="0" maxOccurs="unbounded"/>
116   </xs:sequence>
</xs:complexType>
118 </xs:schema>

```

Listing A.7: SLADatabase.xsd

A.8 SupportiveAgreementDatabase

```

<?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
   <xs:annotation>
4     <xs:documentation>An XML schema definition for the SLM information artifact
       OLADatabase of type database</xs:documentation>
   </xs:annotation>
6   <xs:include schemaLocation="CommonTypes.xsd"/>
   <xs:complexType name="
     type_referenceToServiceDeliveryPartyFromSupportiveAgreement">
8     <xs:attribute name="idref" type="xs:string" use="required"/>
   </xs:complexType>
10   <xs:simpleType name="type_SAType">
     <xs:restriction base="xs:string">
12       <xs:enumeration value="Operational Level Agreement (OLA)"/>
       <xs:enumeration value="Underpinning Contract (UC)"/>
14     </xs:restriction>
   </xs:simpleType>
16   <xs:complexType name="type_subDeliveryParameter">
     <xs:sequence>

```

```

18     <xs:element name="name" type="xs:string"/>
19     <xs:element name="parameterDomain" type="type_parameterDomain"/>
20   </xs:sequence>
21 </xs:complexType>
22 <xs:complexType name="type_subDeliveryTarget">
23   <xs:sequence>
24     <xs:element name="subDeliveryParameter" type="type_subDeliveryParameter"/>
25     <xs:element name="subDeliveryWarranty" type="xs:string"/>
26   </xs:sequence>
27 </xs:complexType>
28 <xs:complexType name="type_subDeliveryTargets">
29   <xs:sequence>
30     <xs:element name="subDeliveryTarget" type="type_subDeliveryTarget"
31       minOccurs="0" maxOccurs="unbounded"/>
32   </xs:sequence>
33 </xs:complexType>
34 <xs:complexType name="type_subLevelParameter">
35   <xs:sequence>
36     <xs:element name="name" type="xs:string"/>
37     <xs:element name="parameterDomain" type="type_parameterDomain"/>
38   </xs:sequence>
39 </xs:complexType>
40 <xs:complexType name="type_subLevelTarget">
41   <xs:sequence>
42     <xs:element name="subLevelParameter" type="type_subLevelParameter"/>
43     <xs:element name="serviceLevelWarranty" type="xs:string"/>
44   </xs:sequence>
45 </xs:complexType>
46 <xs:complexType name="type_subLevelTargets">
47   <xs:sequence>
48     <xs:element name="subLevelTarget" type="type_subLevelTarget" minOccurs="0"
49       maxOccurs="unbounded"/>
50   </xs:sequence>
51 </xs:complexType>
52 <xs:complexType name="type_SAHistoryItem">
53   <xs:sequence>
54     <xs:element name="date" type="xs:dateTime"/>
55     <xs:element name="SAArchivalCopy" type="type_SupportiveAgreement"/>
56   </xs:sequence>
57 </xs:complexType>
58 <xs:complexType name="type_SAHistory">
59   <xs:sequence>
60     <xs:element name="SAHistoryItem" type="type_SAHistoryItem" minOccurs="0"
61       maxOccurs="unbounded"/>
62   </xs:sequence>
63 </xs:complexType>
64 <xs:complexType name="type_SupportiveAgreement">
65   <xs:sequence>
66     <xs:element name="referenceToServiceDeliveryPartyFromSupportiveAgreement"
67       type="type_referenceToServiceDeliveryPartyFromSupportiveAgreement"/>
68     <xs:element name="SAType" type="type_SAType"/>
69     <xs:element name="subDeliveryTargets" type="type_subDeliveryTargets"/>
70     <xs:element name="subLevelTargets" type="type_subLevelTargets"/>
71     <xs:element name="SAHistory" type="type_SAHistory"/>
72   </xs:sequence>
73   <xs:attribute name="id" type="xs:string" use="required"/>
74   <xs:attribute name="date" type="xs:date"/>
75 </xs:complexType>
76 <xs:complexType name="type_SADatabase">
77   <xs:sequence>
78     <xs:element name="SupportiveAgreement" type="type_SupportiveAgreement"
79       minOccurs="0" maxOccurs="unbounded"/>
80   </xs:sequence>

```

```
76 </xs:complexType>
</xs:schema>
```

Listing A.8: SADatabase.xsd

A.9 MeasurementDatabase

```
1 <?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
3 <xs:annotation>
  <xs:documentation>An XML schema definition for the SLM information artifact
    OLADatabase of type database</xs:documentation>
5 </xs:annotation>
<xs:include schemaLocation="CommonTypes.xsd"/>
7 <xs:complexType name="type_referenceToServiceFromMeasurementSpecification">
  <xs:attribute name="idref" type="xs:string" use="required"/>
9 </xs:complexType>
<xs:complexType name="
  type_referenceToMeasurementSpecificationFromMeasurementResults">
11 <xs:attribute name="idref" type="xs:string" use="required"/>
</xs:complexType>
13 <xs:complexType name="type_referenceToServiceLevelMetricFromMeasurementMetric">
  <xs:attribute name="idref" type="xs:string" use="required"/>
15 </xs:complexType>
<xs:complexType name="type_measurementMetric">
17 <xs:sequence>
  <xs:element name="description" type="xs:string"/>
19 <xs:element name="specification" type="xs:string"/>
  <xs:element name="referenceToServiceLevelMetricFromMeasurementMetric" type="
    type_referenceToServiceLevelMetricFromMeasurementMetric"/>
21 </xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
23 </xs:complexType>
<xs:complexType name="type_measurementItem">
25 <xs:sequence>
  <xs:element name="measurementMetric" type="type_measurementMetric"/>
27 <xs:element name="measurementRule" type="xs:string"/>
  <xs:element name="measurementScheduling" type="xs:string"/>
29 </xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
31 </xs:complexType>
<xs:complexType name="type_MeasurementSpecification">
33 <xs:sequence>
  <xs:element name="referenceToServiceFromMeasurementSpecification" type="
    type_referenceToServiceFromMeasurementSpecification"/>
35 <xs:element name="measurementItem" type="type_measurementItem" minOccurs="1
  " maxOccurs="unbounded"/>
</xs:sequence>
37 <xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="date" type="xs:date"/>
39 </xs:complexType>
41 <xs:complexType name="type_measurementPeriod">
  <xs:sequence>
43 <xs:element name="startDateTime" type="xs:dateTime"/>
  <xs:element name="endDateTime" type="xs:dateTime"/>
45 </xs:sequence>
```



```

47 </xs:complexType>
48 <xs:complexType name="type_MeasurementResults">
49   <xs:sequence>
50     <xs:element name="referenceToMeasurementSpecificationFromMeasurementResults"
51       type="type_referenceToMeasurementSpecificationFromMeasurementResults"
52       "/>
53     <xs:element name="measurementPeriod" type="type_measurementPeriod"/>
54     <xs:element name="measurementOutcomes" type="xs:string"/>
55   </xs:sequence>
56   <xs:attribute name="id" type="xs:string" use="required"/>
57   <xs:attribute name="date" type="xs:date"/>
58 </xs:complexType>
59 <xs:complexType name="type_MeasurementDatabase">
60   <xs:sequence>
61     <xs:element name="MeasurementSpecification" type="
62       type_MeasurementSpecification" minOccurs="0" maxOccurs="unbounded"/>
63     <xs:element name="MeasurementResults" type="type_MeasurementResults"
64       minOccurs="0" maxOccurs="unbounded"/>
65   </xs:sequence>
66 </xs:complexType>
67 </xs:schema>

```

Listing A.9: MeasurementDatabase.xsd

A.10 ReportDatabase

```

<?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified" attributeFormDefault="unqualified">
3   <xs:annotation>
4     <xs:documentation>An XML schema definition for the SLM information artifact
      ReportDatabase of type database</xs:documentation>
5   </xs:annotation>
6   <xs:include schemaLocation="CommonTypes.xsd"/>
7   <xs:complexType name="type_referenceToSLAFromExternalReportSpecification">
8     <xs:attribute name="idref" type="xs:string" use="required"/>
9   </xs:complexType>
10  <xs:complexType name="type_referenceToServiceFromInternalReportSpecification">
11    <xs:attribute name="idref" type="xs:string" use="required"/>
12  </xs:complexType>
13  <xs:complexType name="
14    type_referenceToMeasurementMetricFromInternalReportMetrics">
15    <xs:attribute name="idref" type="xs:string" use="required"/>
16  </xs:complexType>
17  <xs:complexType name="
18    type_referenceToMeasurementMetricFromExternalReportMetrics">
19    <xs:attribute name="idref" type="xs:string" use="required"/>
20  </xs:complexType>
21  <xs:complexType name="type_internalReportMetrics">
22    <xs:sequence>
23      <xs:element name="referenceToMeasurementMetricFromInternalReportMetrics"
24        type="type_referenceToMeasurementMetricFromInternalReportMetrics"
25        maxOccurs="unbounded"/>
26    </xs:sequence>
27  </xs:complexType>
28  <xs:complexType name="type_externalReportMetrics">
29    <xs:sequence>

```

```

26     <xs:element name="referenceToMeasurementMetricFromExternalReportMetrics"
           type="type_referenceToMeasurementMetricFromExternalReportMetrics"
           maxOccurs="unbounded"/>
27     </xs:sequence>
28 </xs:complexType>
29 <xs:complexType name="type_reportSection">
30   <xs:sequence>
31     <xs:element name="title" type="xs:string"/>
32     <xs:element name="sectionContentSpecification" type="xs:string"/>
33   </xs:sequence>
34 </xs:complexType>
35 <xs:complexType name="type_reportAdditionalContents">
36   <xs:sequence>
37     <xs:element name="reportSection" type="type_reportSection" maxOccurs="
           unbounded"/>
38   </xs:sequence>
39 </xs:complexType>
40 <xs:complexType name="type_ExternalReportSpecification">
41   <xs:sequence>
42     <xs:element name="referenceToSLAFromExternalReportSpecification" type="
           type_referenceToSLAFromExternalReportSpecification"/>
43     <xs:element name="reportScheduling" type="xs:string"/>
44     <xs:element name="externalReportMetrics" type="type_externalReportMetrics"/
           >
45     <xs:element name="reportAdditionalContents" type="
           type_reportAdditionalContents" minOccurs="0"/>
46   </xs:sequence>
47   <xs:attribute name="id" type="xs:string" use="required"/>
48   <xs:attribute name="date" type="xs:date"/>
49 </xs:complexType>
50 <xs:complexType name="type_InternalReportSpecification">
51   <xs:sequence>
52     <xs:element name="referenceToServiceFromInternalReportSpecification" type="
           type_referenceToServiceFromInternalReportSpecification"/>
53     <xs:element name="reportScheduling" type="xs:string"/>
54     <xs:element name="internalReportMetrics" type="type_internalReportMetrics"/
           >
55     <xs:element name="reportAdditionalContents" type="
           type_reportAdditionalContents" minOccurs="0"/>
56   </xs:sequence>
57   <xs:attribute name="id" type="xs:string" use="required"/>
58   <xs:attribute name="date" type="xs:date"/>
59 </xs:complexType>
60 <xs:complexType name="
           type_referenceToInternalReportSpecificationFromInternalReport">
61   <xs:attribute name="idref" type="xs:string" use="required"/>
62 </xs:complexType>
63 <xs:complexType name="
           type_referenceToExternalReportSpecificationFromExternalReport">
64   <xs:attribute name="idref" type="xs:string" use="required"/>
65 </xs:complexType>
66 <xs:complexType name="type_referenceToMeasurementResultsFromInternalReport">
67   <xs:attribute name="idref" type="xs:string" use="required"/>
68 </xs:complexType>
69 <xs:complexType name="type_referenceToMeasurementResultsFromExternalReport">
70   <xs:attribute name="idref" type="xs:string" use="required"/>
71 </xs:complexType>
72 <xs:complexType name="type_serviceReportPeriod">
73   <xs:sequence>
74     <xs:element name="startDateTime" type="xs:dateTime"/>
75     <xs:element name="endDateTime" type="xs:dateTime"/>
76   </xs:sequence>
77 </xs:complexType>

```

```

78 <xs:complexType name="type_InternalReport">
  <xs:sequence>
80 <xs:element name="referenceToInternalReportSpecificationFromInternalReport"
  type="type_referenceToInternalReportSpecificationFromInternalReport"/>
  <xs:element name="referenceToMeasurementResultsFromInternalReport" type="
    type_referenceToMeasurementResultsFromInternalReport" maxOccurs="
    unbounded"/>
82 <xs:element name="serviceReportPeriod" type="type_serviceReportPeriod"/>
  <xs:element name="serviceReportContent" type="xs:string"/>
84 </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="date" type="xs:date"/>
86 </xs:complexType>
<xs:complexType name="type_ExternalReport">
  <xs:sequence>
88 <xs:element name="referenceToExternalReportSpecificationFromExternalReport"
  type="type_referenceToExternalReportSpecificationFromExternalReport"/>
  <xs:element name="referenceToMeasurementResultsFromExternalReport" type="
    type_referenceToMeasurementResultsFromInternalReport" maxOccurs="
    unbounded"/>
92 <xs:element name="serviceReportPeriod" type="type_serviceReportPeriod"/>
  <xs:element name="serviceReportContent" type="xs:string"/>
94 </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="date" type="xs:date"/>
96 </xs:complexType>
98 <xs:complexType name="
  type_referenceToMeasurementResultsFromSLAViolationNotification">
  <xs:attribute name="idref" type="xs:string" use="required"/>
100 </xs:complexType>
<xs:complexType name="type_referenceToSLAFromSLAViolationNotification">
  <xs:attribute name="idref" type="xs:string" use="required"/>
102 </xs:complexType>
104 <xs:complexType name="type_SLAViolationNotification">
  <xs:sequence>
106 <xs:element name="referenceToMeasurementResultsFromSLAViolationNotification"
  type="type_referenceToMeasurementResultsFromSLAViolationNotification"
  maxOccurs="unbounded"/>
  <xs:element name="referenceToSLAFromSLAViolationNotification" type="
    type_referenceToSLAFromSLAViolationNotification"/>
108 <xs:element name="violationSpecification" type="xs:string"/>
  <xs:element name="notificationAdditionalRemarks" type="xs:string"/>
110 </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="date" type="xs:date"/>
112 </xs:complexType>
<xs:complexType name="type_ReportDatabase">
  <xs:sequence>
114 <xs:element name="InternalReportSpecification" type="
  type_InternalReportSpecification" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="ExternalReportSpecification" type="
  type_ExternalReportSpecification" minOccurs="0" maxOccurs="unbounded"/>
118 <xs:element name="InternalReport" type="type_InternalReport" minOccurs="0"
  maxOccurs="unbounded"/>
  <xs:element name="ExternalReport" type="type_ExternalReport" minOccurs="0"
  maxOccurs="unbounded"/>
120 <xs:element name="SLAViolationNotification" type="
  type_SLAViolationNotification" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>

```

122 </xs:complexType>
</xs:schema>

Listing A.10: ReportDatabase.xsd

A.11 ConflictDatabase

```

1 <?xml version="1.0" encoding="UTF-8"?>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
    qualified" attributeFormDefault="unqualified">
3    <xs:annotation>
      <xs:documentation>An XML schema definition for the SLM information artifact
        ConflictDatabase of type database</xs:documentation>
5    </xs:annotation>
    <xs:include schemaLocation="CommonTypes.xsd"/>
7    <xs:complexType name="type_referenceToSLAFromAgreementConflict">
      <xs:attribute name="idref" type="xs:string" use="required"/>
9    </xs:complexType>
    <xs:complexType name="type_referenceToSAFromAgreementConflict">
11     <xs:attribute name="idref" type="xs:string" use="required"/>
    </xs:complexType>
13    <xs:complexType name="type_referenceToAgreement">
      <xs:choice>
15        <xs:element name="referenceToSLAFromAgreementConflict" type="
          type_referenceToSLAFromAgreementConflict"/>
          <xs:element name="referenceToSAFromAgreementConflict" type="
            type_referenceToSAFromAgreementConflict"/>
17        </xs:choice>
      </xs:complexType>
19    <xs:complexType name="type_AgreementConflict">
      <xs:sequence>
21        <xs:element name="referenceToAgreement" type="type_referenceToAgreement"
          minOccurs="2" maxOccurs="2"/>
          <xs:element name="conflictDescription" type="xs:string"/>
23        <xs:element name="conflictResolutionOptions" type="xs:string"/>
      </xs:sequence>
25      <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:attribute name="date" type="xs:date"/>
27    </xs:complexType>
    <xs:complexType name="type_ConflictDatabase">
29      <xs:sequence>
        <xs:element name="AgreementConflict" type="type_AgreementConflict"
          minOccurs="0" maxOccurs="unbounded"/>
31      </xs:sequence>
    </xs:complexType>
33 </xs:schema>

```

Listing A.11: ConflictDatabase.xsd

Anhang B

Exemplarische Instanziierung des SLM-Informationssystems

Das folgende Listing stellt eine gültige (d.h. gegen die entsprechende *XML Schema Definition* validierte) Instanz des SLM-Informationssystems dar, wie in Kapitel 7.2 beschrieben.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ServiceLevelManagementInformationSystem xmlns:xsi="http://www.w3.org/2001/
  XMLSchema-instance" xsi:noNamespaceSchemaLocation="H:\diss\neu\XML\
  ServiceLevelManagementInformationSystem.xsd">
3   <ServicePortfolio>
4     <Service id="srv-1" date="2008-11-18">
5       <name>Wireless LAN</name>
6       <serviceDescription>In vielen Bereichen der Muenchner Universitaeten und
          Hochschulen ist ein Netzzugang ueber WLAN (Wireless LAN) moeglich.
          WLAN bietet insbesondere allen mobilen Nutzern mit Laptop, PDA oder
          aehnlichem die Gelegenheit, drahtlos die IT-Dienste im MWN und
          natuerlich auch im Internet zu nutzen.</serviceDescription>
7       <serviceRequirements>
8         <serviceRequirement>
9           <description>WLAN-Zugang in Hoersaelen</description>
10          <priority>1</priority>
11         </serviceRequirement>
12         <serviceRequirement>
13           <description>WLAN-Zugang in uebrigen Raeumen</description>
14           <priority>2</priority>
15         </serviceRequirement>
16         <serviceRequirement>
17           <description>WLAN-Zugang in Aussenbereichen</description>
18           <priority>3</priority>
19         </serviceRequirement>
20       </serviceRequirements>
21       <serviceSpecification>
22         <serviceUsageFunctionality>Uneingeschraenkter Internetzugang,
          verschluesselte Uebertragung</serviceUsageFunctionality>
23         <serviceAccessPointSpecification>Standard Access Point</
          serviceAccessPointSpecification>
          <serviceClientSpecification>Cisco VPN-Client auf Endgeraet mit Adapter
          fuer Wireless LAN im IEEE 802.11g- oder IEEE 802.11b-Standard</
          serviceClientSpecification>
```

```

25     <serviceManagementFunctionality>von WLAN Access Point Hardware</
      serviceManagementFunctionality>
26     <serviceManagementAccessPointSpecification>von WLAN Access Point Hardware
      </serviceManagementAccessPointSpecification>
27     <serviceManagementClientSpecification>Browser (beliebig)</
      serviceManagementClientSpecification>
28     </serviceManagementClientSpecification>
29     <serviceResources/>
30     <serviceDependencies>
31       <dependency>Physische Netzanbindung MWN</dependency>
32       <dependency>DHCP</dependency>
33       <dependency>DNS</dependency>
34     </serviceDependencies>
35   </serviceSpecification>
36   <serviceDeliveryParameters>
37     <serviceDeliveryParameter id="srv-1-sdp-1">
38       <name>Anzahl Nutzerkonten</name>
39       <parameterDomain>
40         <selectFromRange>
41           <minimumInclusiveValue>10</minimumInclusiveValue>
42           <maximumInclusiveValue>200000</maximumInclusiveValue>
43         </selectFromRange>
44       </parameterDomain>
45     </serviceDeliveryParameter>
46     <serviceDeliveryParameter id="srv-1-sdp-2">
47       <name>Authentifizierungsmethode</name>
48       <parameterDomain>
49         <selectFromList>
50           <listElement>Offen (keine Authentifizierung erforderlich)</
51             listElement>
52           <listElement>VPN-Client</listElement>
53           <listElement>Browser/Web-basiert</listElement>
54         </selectFromList>
55       </parameterDomain>
56     </serviceDeliveryParameter>
57   </serviceDeliveryParameters>
58   <serviceLevelParameters>
59     <serviceLevelParameter id="srv-1-slp-1">
60       <name>Garantierte Dienstbetriebszeiten</name>
61       <parameterDomain>
62         <selectFromList>
63           <listElement>24 Stunden (24/7)</listElement>
64           <listElement>Montag bis Sonntag, 06:00 bis 22:00 Uhr</listElement>
65           <listElement>Montag bis Samstag, 06:00 bis 22:00 Uhr</listElement>
66           <listElement>Montag bis Freitag, 06:00 bis 20:00 Uhr</listElement>
67         </selectFromList>
68       </parameterDomain>
69     </serviceLevelParameter>
70     <serviceLevelParameter id="srv-1-slp-2">
71       <name/>
72       <parameterDomain>
73         <selectFromRange>
74           <minimumInclusiveValue>90</minimumInclusiveValue>
75           <maximumInclusiveValue>99e-3</maximumInclusiveValue>
76         </selectFromRange>
77       </parameterDomain>
78     </serviceLevelParameter>
79   </serviceLevelParameters>
80   <serviceLevelMetrics>
81     <serviceLevelMetric id="srv-1-slm-1">
82       <name>Durchschnittliche Verfuegbarkeit</name>
83       <description>Durchschnittliche Verfuegbarkeit auf Basis der
84         garantierten Dienstbetriebszeiten</description>
85       <measurementRule>Uptime/(Uptime+Downtime)</measurementRule>

```

```

83     </serviceLevelMetric>
      </serviceLevelMetrics>
85     <serviceStatus>active-operating</serviceStatus>
      <serviceHistory/>
87   </Service>
  <ServiceCatalog id="sc-1">
89     <referenceToCustomerFromServiceCatalog idref="cust-1"/>
      <referenceToServiceFromServiceCatalog id="sc-1-srv-1" idref="srv-1"/>
91     <serviceCatalogValidity>
      <serviceCatalogValidFrom>2008-11-01T00:00:00</serviceCatalogValidFrom>
93     <serviceCatalogValidThrough>2009-11-01T00:00:00</
      serviceCatalogValidThrough>
      </serviceCatalogValidity>
95     <serviceCatalogHistory/>
  </ServiceCatalog>
</ServicePortfolio>
<CustomerDatabase>
99   <Customer id="cust-1">
      <name>LMU</name>
101     <customerContactInformation>
      <customerContactPoint id="cust-1-ccp-1">
103       <customerContactPointEmail>verwaltung@lmu.de</customerContactPointEmail
      >
      <customerContactPointAddress>Geschwister-Scholl-Platz</
      customerContactPointAddress>
105     </customerContactPoint>
      </customerContactInformation>
107     <customerRoleAssignments>
      <customerRoleAssignment>
109       <role/>
      <referenceToCustomerContactPointFromCustomerRoleAssignment idref=""/>
111     </customerRoleAssignment>
      </customerRoleAssignments>
113     <customerServiceInterfaces>
      <customerServiceInterface id="">
115       <customerServiceInterfaceClient/>
      <customerServiceInterfaceConnectivity/>
117     </customerServiceInterface>
      </customerServiceInterfaces>
119   </Customer>
</CustomerDatabase>
<SDPDatabase/>
<SLPDatabase>
123   <SupportLevelProfile id="slp-1">
      <name>Standard Support</name>
125     <supportChannels>
      <channel>
127       <channelOption>phone</channelOption>
      <contactInfo>(089) 35831 8800</contactInfo>
129     </channel>
      <channel>
131       <channelOption>email</channelOption>
      <contactInfo>hotline@lrz.de</contactInfo>
133     </channel>
      <channel>
135       <channelOption>webform</channelOption>
      <contactInfo>http://hotline.lrz-muenchen.de</contactInfo>
137     </channel>
      <channel>
139       <channelOption>physical</channelOption>
      <contactInfo>Boltzmannstr. 1, Garching</contactInfo>
141     </channel>
  </supportChannels>

```

```

143     <supportTimes>
144         <supportTime>
145             <weekday>mon</weekday>
146             <startTime>09:00:00</startTime>
147             <endTime>17:00:00</endTime>
148         </supportTime>
149         <supportTime>
150             <weekday>tue</weekday>
151             <startTime>09:00:00</startTime>
152             <endTime>17:00:00</endTime>
153         </supportTime>
154         <supportTime>
155             <weekday>wed</weekday>
156             <startTime>09:00:00</startTime>
157             <endTime>17:00:00</endTime>
158         </supportTime>
159         <supportTime>
160             <weekday>thu</weekday>
161             <startTime>09:00:00</startTime>
162             <endTime>17:00:00</endTime>
163         </supportTime>
164         <supportTime>
165             <weekday>fri</weekday>
166             <startTime>09:00:00</startTime>
167             <endTime>17:00:00</endTime>
168         </supportTime>
169     </supportTimes>
170     <maxResponseTimes>
171         <maxResponseTime>
172             <timeFrame>POYOMODTOH5MOS</timeFrame>
173             <conditions>Maximale Wartezeit bis Anrufannahme (channelOption phone)</
174                 conditions>
175         </maxResponseTime>
176         <maxResponseTime>
177             <timeFrame>POYOMODTOH10MOS</timeFrame>
178             <conditions>Maximale Wartezeit bis Ticketerstellung (channelOptions
179                 email und webform)</conditions>
180         </maxResponseTime>
181     </maxResponseTimes>
182     <maxRecoveryTimes/>
183 </SupportLevelProfile>
184 </SLPDatabase>
185 <SLADatabase>
186     <ServiceLevelAgreement id="sla-1">
187         <referenceToServiceFromSLA idref="sc-1-srv-1"/>
188         <referenceToCustomerFromSLA idref="cust-1"/>
189         <referenceToSLPFromSLA idref="slp-1"/>
190         <serviceDeliveryTargets>
191             <serviceDeliveryTarget>
192                 <referenceToServiceDeliveryParameterFromSLA idref="srv-1-sdp-1"/>
193                 <serviceDeliveryWarranty>12000</serviceDeliveryWarranty>
194             </serviceDeliveryTarget>
195         </serviceDeliveryTargets>
196         <serviceLevelTargets>
197             <serviceLevelTarget>
198                 <referenceToServiceLevelParameterFromSLA idref="srv-1-slp-1"/>
199                 <serviceLevelWarranty>99.8</serviceLevelWarranty>
200             </serviceLevelTarget>
201         </serviceLevelTargets>
202         <serviceDegradationDefinition>
203             <serviceDegradationModel>
204                 <serviceDegradationDegree>10</serviceDegradationDegree>
205                 <serviceDegradationModelConditionalClause>

```



```
205         <degradationSubject>Physische Access Point Hardware</  
           degradationSubject>  
207         <degradationManner>Ohne Funktion</degradationManner>  
           <degradationTime>Laenger als 15 Minuten</degradationTime>  
           </serviceDegradationModelConditionalClause>  
           </serviceDegradationModel>  
209     </serviceDegradationDefinition>  
           <deliveryConditions/>  
211     <SLAHistory/>  
           </ServiceLevelAgreement>  
213 </SLADatabase>  
           <SADatabase/>  
215 <MeasurementDatabase/>  
           <ReportDatabase/>  
217 <ConflictDatabase/>  
</ServiceLevelManagementInformationSystem>
```

Listing B.1: ServiceLevelManagementInformationSystemExemplaryInstance.xml

Anhang C

Literaturverzeichnis

- [1] HEGERING, H.-G. ; ABECK, S. ; NEUMAIR, B.: *Integriertes Management vernetzter Systeme – Konzepte, Architekturen und deren betrieblicher Einsatz*. Dpunkt Verlag, 1999. – ISBN 978-3932588167
- [2] IQBAL, M. ; NIEVES, M. ; OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Service Strategy*. The Stationary Office (TSO), 2007 (IT Infrastructure Library (ITIL)). – ISBN 978-0113310456
- [3] EXAMINATION INSTITUTE FOR INFORMATION SCIENCE (EXIN): *Fallbeschreibung ITIL Service Manager – Internationale Transportgesellschaft*. 2003
- [4] DREO RODOSEK, G.: *A Framework for IT Service Management*, Habilitation, Juni 2002
- [5] FITTS, P. M. ; VITELES, M. S. ; BARR, N. L. ; BRIMHALL, D. R. ; FINCH, G. ; GARDNER, E. ; GREYER, W. F. ; KELLUM, W. E. ; STEVENS, S. S.: *Human engineering for an effective air navigation and traffic control system*. Washington, D.C., USA, März 1951. – Forschungsbericht
- [6] LEWIS, L.: *Service Level Management for Enterprise Networks*. Artech House, 1999. – ISBN 978-1580530163
- [7] LEWIS, L. ; RAY, P.: *Service Level Management – Definition, Architecture and Research Challenges*. In: *Proceedings of Global Communications Conference (Globecom 1999)*. Rio de Janeiro, Brasilien, Dezember 1999
- [8] BRENNER, M.: *Classifying ITIL Processes – A Taxonomy under Tool Support Aspects*. In: *Proceedings of First IEEE/IFIP International Workshop*

- on Business-Driven IT Management (BDIM 2006)*. Vancouver, Kanada, April 2006
- [9] BRENNER, M.: *Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement – Ein modellbasierter Ansatz*, Ludwig-Maximilians-Universität München, Diss., Juli 2007
- [10] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Service Support*. The Stationary Office (TSO), 2000 (IT Infrastructure Library (ITIL)). – ISBN 978-0113300150
- [11] SAILER, M.: *Konzeption einer Service-MIB – Analyse und Spezifikation dienstorientierter Managementinformation*, Ludwig-Maximilians-Universität München, Diss., Juli 2007
- [12] SCHMITZ, D.: *Automated Service-Oriented Impact Analysis and Recovery Alternative Selection*, Ludwig-Maximilians-Universität München, Diss., Juli 2008
- [13] GARSCHHAMMER, M.: *Dienstgütebehandlung im Lebenszyklus – von der formalen Spezifikation zur rechnergestützten Umsetzung*, Ludwig-Maximilians-Universität München, Diss., Juli 2004
- [14] SCHMIDT, H.: *Entwurf von Service Level Agreements auf der Basis von Dienstprozessen*, Ludwig-Maximilians-Universität München, Diss., Juli 2001
- [15] OBJECT MANAGEMENT GROUP (OMG) (Hrsg.): *Model Driven Architecture (MDA)*. Object Management Group (OMG), Juli 2001
- [16] OBJECT MANAGEMENT GROUP (OMG) (Hrsg.): *MDA Guide*. Object Management Group (OMG), Juni 2003
- [17] BRENNER, M. ; SCHAAF, T.: On Tool Support for Service Level Management: From Requirements to System Specifications. In: *Proceedings of 3rd IEEE/IFIP International Workshop on Business-driven IT Management (BDIM 2008)*. Salvador, Brasilien : IEEE Publishing, April 2008
- [18] ITSMF INTERNATIONAL (Hrsg.): *ISO/IEC 20000 – An Introduction*. Van Haren Publishing, 2008 (ITSM Library). – ISBN 978-9087530815
- [19] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *The Official Introduction to the ITIL Service Lifecycle*. The Stationary Office (TSO), 2007 (IT Infrastructure Library (ITIL)). – ISBN 978-0113310616

- [20] TELEMANAGEMENT FORUM: Business Process Framework (eTOM) For The Information and Communications Services Industry / TMF. 2008 (GB921). – Forschungsbericht
- [21] Deutsches Institut für Normung (DIN) & International Organization for Standardization (ISO): *ISO 9000:2005 – Qualitätsmanagementsysteme – Grundlagen und Begriffe*. 2005
- [22] Deutsches Institut für Normung (DIN) & International Organization for Standardization (ISO): *ISO 9001:2000 – Qualitätsmanagementsysteme – Anforderungen*. 2000
- [23] MATTHIES, M.: *Einführung in die Systemwissenschaft*. 2002
- [24] BOSSEL, H.: *Simulation dynamischer Systeme – Grundwissen, Methoden, Programme*. Braunschweig : Vieweg-Verlag, 1989
- [25] RUDD, C. ; LLOYD, V. ; OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Service Design*. The Stationary Office (TSO), 2007 (IT Infrastructure Library (ITIL)). – ISBN 978-0113310470
- [26] GARSCHHAMMER, M. ; HAUCK, R. ; HEGERING, H.-G. ; KEMPTER, B. ; LANGER, M. ; NERB, M. ; RADISIC, I. ; ROELLE, H. ; SCHMIDT, H.: Towards generic Service Management Concepts – A Service Model Based Approach. In: *Proceedings of 7th IEEE/IFIP International Symposium on Integrated Management (IM 2001)*. Seattle, Washington, USA : IEEE Publishing, Mai 2001
- [27] GARSCHHAMMER, M. ; HAUCK, R. ; KEMPTER, B. ; RADISIC, I. ; ROELLE, H. ; SCHMIDT, H.: The MNM Service Model – Refined Views on Generic Service Management. In: *Journal of Communications and Networks* 3 (2001), Dezember, Nr. 4
- [28] LANGER, M.: *Konzeption und Anwendung einer Customer Service Management Architektur*, Technische Universität München, Diss., März 2001
- [29] SCHULTE-ZURHAUSEN, M.: *Organisation*. Verlag Franz Vahlen GmbH, 2005. – ISBN 978-3800632053
- [30] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Service Delivery*. The Stationary Office (TSO), 2001 (IT Infrastructure Library (ITIL)). – ISBN 978-0113300174

- [31] International Organization for Standardization (ISO) & International Electrotechnical Commission (IEC): *ISO/IEC 20000-2:2005 – Information Technology – Service Management – Part 2: Code of Practice*. Dezember 2005
- [32] OBJECT MANAGEMENT GROUP (OMG) (Hrsg.): *OMG Unified Modeling Language Specification*. Object Management Group (OMG), März 2003
- [33] OESTERREICH, B.: *Analyse und Design mit UML 2.1 – Objektorientierte Softwareentwicklung*. Oldenbourg Wissenschaftsverlag, 2006. – 8. aktualisierte Auflage, inkl. Poster mit UML-Notationsübersicht & OEP-Vorgehensübersicht, ISBN 978-3486272666
- [34] STÖRRLE, H.: *UML 2 erfolgreich einsetzen – Einführung und Referenz*. Addison-Wesley, 2005. – ISBN 978-3827325839
- [35] WORLD WIDE WEB CONSORTIUM (W3C) (Hrsg.): *Extensible Markup Language (XML)*. World Wide Web Consortium (W3C), Februar 2004
- [36] BECKER, J. ; ROSEMAN, M. ; SCHÜTTE, R.: Grundsätze ordnungsmäßiger Modellierung. In: *Wirtschaftsinformatik* 37 (1995), Nr. 5
- [37] BECKER, J.: Die Grundsätze ordnungsmäßiger Modellierung und ihre Einbettung in ein Vorgehensmodell zur Erstellung betrieblicher Informationsmodelle / Gesellschaft für Informatik. – Forschungsbericht
- [38] DREO RODOSEK, G.: A Generic Model for IT Services and Service Management. In: *Integrated Network Management VIII – Managing It All*. Colorado Springs, USA : Kluwer Academic Publishers, März 2003
- [39] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Security Management*. The Stationary Office (TSO), 2000 (IT Infrastructure Library (ITIL)). – ISBN 978-0113300143
- [40] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Application Management*. The Stationary Office (TSO), 2002 (IT Infrastructure Library (ITIL)). – ISBN 978-0113308668
- [41] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *ICT Infrastructure Management*. The Stationary Office (TSO), 2002 (IT Infrastructure Library (ITIL)). – ISBN 978-0113308651
- [42] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Planning to Implement Service Management*. The Stationary Office (TSO), 2002 (IT Infrastructure Library (ITIL)). – ISBN 978-0113308774

- [43] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Business Perspective: The IS View on Delivering Services to the Business*. The Stationary Office (TSO), 2003 (IT Infrastructure Library (ITIL)). – ISBN 978-0113308941
- [44] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Introduction to ITIL*. The Stationary Office (TSO), 2005 (IT Infrastructure Library (ITIL)). – ISBN 978-0113309733
- [45] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *ITIL Software Asset Management*. The Stationary Office (TSO), 2003 (IT Infrastructure Library (ITIL)). – ISBN 978-0113309436
- [46] OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *ITIL Small-scale Implementation*. The Stationary Office (TSO), 2006 (IT Infrastructure Library (ITIL)). – ISBN 978-0113309801
- [47] LACY, S. ; MACFARLANE, I. ; OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Service Transition*. The Stationary Office (TSO), 2007 (IT Infrastructure Library (ITIL)). – ISBN 978-0113310487
- [48] CANNON, D. ; WHEELDON, D. ; OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Service Operation*. The Stationary Office (TSO), 2007 (IT Infrastructure Library (ITIL)). – ISBN 978-0113310463
- [49] CASE, G. ; OFFICE OF GOVERNMENT COMMERCE (OGC) (Hrsg.): *Continual Service Improvement*. The Stationary Office (TSO), 2007 (IT Infrastructure Library (ITIL)). – ISBN 978-0113310494
- [50] British Standards Institution (BSI): *BS 15000-1 – Specification for Service Management*. 2002
- [51] British Standards Institution (BSI): *BS 15000-2 – Code of Practice for Service Management*. 2002
- [52] International Organization for Standardization (ISO) & International Electrotechnical Commission (IEC): *ISO/IEC 20000-1:2005 – Information Technology – Service Management – Part 1: Specification*. Dezember 2005
- [53] TELEMANAGEMENT FORUM: *Enhanced Telecom Operations Map (eTOM) – The Business Process Framework For The Information and Communications Services Industry – Addendum P: An eTOM Primer / TMF*. 2004 (GB921-P). – Forschungsbericht

- [54] TELEMANAGEMENT FORUM: Business Process Framework (eTOM) For The Information and Communications Services Industry – Addendum D: Process Decompositions and Descriptions / TMF. 2008 (GB921-D). – Forschungsbericht
- [55] TELEMANAGEMENT FORUM: Business Process Framework (eTOM) For The Information and Communications Services Industry – Addendum F: Process Flow Examples / TMF. 2008 (GB921-F). – Forschungsbericht
- [56] TELEMANAGEMENT FORUM: Business Process Framework (eTOM) For The Information and Communications Services Industry – Addendum U: User Guidelines for eTOM / TMF. 2007 (GB921-U). – Forschungsbericht
- [57] TELEMANAGEMENT FORUM, IT SERVICE MANAGEMENT FORUM: Building Bridges: ITIL and eTOM – A study by TM Forum in conjunction with itsMF on integrating ITIL and eTOM to provide a pragmatic joint solution for business support in the communications sector / TMF & itsMF. 2008 (TR143). – Forschungsbericht
- [58] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 0 – SID Primer / TMF. 2004 (GB922-0). – Forschungsbericht
- [59] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1A – Common Business Entity Definitions – Agreement (including Service Level Agreement) / TMF. 2005 (GB922-1A). – Forschungsbericht
- [60] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1BI – Common Business Entity Definitions – Business Interaction / TMF. 2005 (GB922-1BI). – Forschungsbericht
- [61] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1BT – Common Business Entity Definitions – Base Types / TMF. 2008 (GB922-1BT). – Forschungsbericht
- [62] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1C – Business View Concepts and Principles – Business Contract / TMF. 2007 (GB922-1C). – Forschungsbericht
- [63] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1J – PROJECT Business Entity Definitions / TMF. 2004 (GB922-1J). – Forschungsbericht

- [64] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1L – Common Business Entity Definitions – Location / TMF. 2004 (GB922-1L). – Forschungsbericht
- [65] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1P – Common Business Entity Definitions – Party / TMF. 2004 (GB922-1P). – Forschungsbericht
- [66] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1POL – Common Business Entity Definitions – Policy / TMF. 2004 (GB922-1POL). – Forschungsbericht
- [67] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1R – Common Business Entity Definitions – Root Business Entities / TMF. 2008 (GB922-1R). – Forschungsbericht
- [68] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 1T – Time Related Business Entity Definitions / TMF. 2004 (GB922-1T). – Forschungsbericht
- [69] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 2 – Customer Business Entity Definitions / TMF. 2005 (GB922-2). – Forschungsbericht
- [70] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 3 – Product Business Entity Definitions / TMF. 2008 (GB922-3). – Forschungsbericht
- [71] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 4SO – Service Overview Business Entity Definitions / TMF. 2008 (GB922-4SO). – Forschungsbericht
- [72] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 4S-QoS – Quality of Service Business Entity Definitions / TMF. 2004 (GB922-4S-QoS). – Forschungsbericht
- [73] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 5LR – Logical Resource and Compound Resource Business Entity Definitions / TMF. 2008 (GB922-5LR). – Forschungsbericht
- [74] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 5PR – Physical Resource Business Entity Definitions / TMF. 2004 (GB922-5PR). – Forschungsbericht

- [75] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 6 – Market/Sales Business Entity Definitions / TMF. 2008 (GB922-6). – Forschungsbericht
- [76] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model Addendum 7RA – Enterprise Domain Revenue Assurance Business Entities / TMF. 2007 (GB922-7RA). – Forschungsbericht
- [77] DEBUSMANN, M.: *Modellbasiertes Service Level Management verteilter Anwendungssysteme*, Universität Kassel, Diss., Juni 2005
- [78] REILLY, J.P. ; CREANER, M. J. ; TELE MANAGEMENT FORUM (TMF) (Hrsg.): *NGOSS distilled: The essential guide to next generation telecoms*. The Lean Corporation, 2005 (Transforming the Telecommunications Industry). – ISBN 978-0955108709
- [79] TELEMANAGEMENT FORUM: SLA Management Handbook: Volume 1 – Executive Overview / TMF. 2005 (GB917-1). – Forschungsbericht
- [80] TELEMANAGEMENT FORUM: SLA Management Handbook: Volume 2 – Concepts and Principles / TMF. 2005 (GB917-2). – Forschungsbericht
- [81] TELEMANAGEMENT FORUM: SLA Management Handbook: Volume 3 – Service and Technology Examples / TMF. 2005 (GB917-3). – Forschungsbericht
- [82] TELEMANAGEMENT FORUM: SLA Management Handbook: Volume 4 – Enterprise Perspective / TMF. Erscheinungsdatum unbekannt (GB917-4). – Forschungsbericht. – Publikation angekündigt
- [83] SCHAAF, T.: Frameworks for Business-driven Service Level Management – A Criteria-based Comparison of ITIL and NGOSS. In: *Proceedings of 2nd IEEE/IFIP International Workshop on Business-driven IT Management (BDIM 2007)*. München : IEEE Publishing, Mai 2007
- [84] STURM, R. ; MORRIS, W. ; JANDER, M.: *Foundations of Service Level Management*. Sams, 2000. – ISBN 978-0672317439
- [85] BERNHARD, M. G. (Hrsg.) ; LEWANDOWSKI, W. (Hrsg.) ; MANN, H. (Hrsg.): *Service-Level-Management in der IT – Wie man erfolgskritische Leistungen definiert und steuert*. Symposium Publishing, 2004. – Inhalt: Fachbeiträge, Analysen, Interviews, ISBN 978-3936608281

- [86] HILES, A.: *The Complete Guide to IT Service Level Agreements – Aligning IT Services to Business Needs*. Rothstein Associates Inc., 2002. – ISBN 978-1931332132
- [87] TYURIN, N.: *Aufbau und Zusammenhang der drei Service Level Management Vertragstypen SLA, Ola, UC*. Grin Verlag für akademische Texte, 2007. – ISBN 978-3638693349
- [88] HASSENPFUG, P.: *Analyse von Service Level Management-Prozessen in der BMW Group*, Ludwig-Maximilians-Universität München, Diplomarbeit, Februar 2004
- [89] SCHERER, A.: *Entwicklung einer Methodik zur Informations- und Datenmodellierung in IT-Service-Management-Prozessen am Beispiel der ITIL-Prozesse Service Level Management und Configuration Management*. Grin Verlag für akademische Texte, 2007. – ISBN 978-3638939997
- [90] BMC SOFTWARE, INC. (Hrsg.): *Action Request System 6.3 – Developing AR System Applications: Advanced*. Januar 2005
- [91] FALLSIDE, D. C. ; WALMSLEY, P.: *XML Schema Part 0: Primer*, Oktober 2004
- [92] WALMSLEY, P.: *Definitive XML Schema*. Prentice Hall, 2001 (The Charles F. Goldfarb Definitive XML Series). – ISBN 978-0130655677
- [93] SHERIDAN, T. B.: *Humans and Automation: System Design and Research Issues*. Wiley Interscience, 2002 (Wiley Series in System Engineering and Management). – ISBN 978-0471234289
- [94] BRENNER, M. ; GARSCHHAMMER, M. ; SAILER, M. ; SCHAAF, T.: CMDB – Yet Another MIB? On Reusing Management Model Concepts in ITIL Configuration Management. In: *Large Scale Management of Distributed Systems, Proceedings of 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2006)*. Dublin, Irland : Springer-Verlag, Oktober 2006
- [95] CASE, J. ; MCCLOGHRIE, K. ; ROSE, M. ; WALDBUSSER, S.: RFC 1902: Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2) / Internet Engineering Task Force (IETF). 1996. – RFC
- [96] DISTRIBUTED MANAGEMENT TASK FORCE (DMTF): *Common Information Model (CIM) Schema, Version 2.9*, Juni 2005

- [97] TELEMANAGEMENT FORUM: Shared Information/Data (SID) Model / TMF. 2008 (GB922 Solution Suite Release 8.0). – Forschungsbericht

- [98] BRENNER, M. ; SCHAAF, T. ; SCHERER, A.: Towards an Information Model for ITIL and ISO/IEC 20000 processes. In: *Making Management Scalable, Robust, Cost-Effective and Revenue-Generating, Proceedings of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)*. New York, USA, Juni 2009. – Publikation angekündigt/zur Veröffentlichung akzeptiert

Index

A

Aktorenkommunikation	272
Auswertung	281
Klassifikation	268
Anforderungsanalyse	51
Anwendungsfälle	99
Funktionale Anforderungen	72
Morphologischer Kasten	54
Nicht-funktionale Anforderungen	
127	
Systematik	36
Szenario 1	59
Szenario 2	63
Szenario 3	66
Vollständigkeit	70
Anfrage-Antwort-Kommunikation	270,
350	
API ... <i>siehe</i> Application Programming	
Interface	
Application Programming Interface	359
Architektur	<i>siehe</i>
Managementarchitektur	
Artefakt ... <i>siehe</i> Informationsartefakt	
Artefakt-Browser	363
Automatisierung	
Ausschlusskriterien	337
Potenzialanalyse	337
Stufenschema	333
Availability Management	286

B

BDIM	<i>siehe</i> Business-driven IT
Management	
Bericht	179
Berichtsspezifikation	209
Berichtswesen	257
Budgeting and Accounting	287
Business-driven IT Management	2, 46

C

Capacity Management	285
Change Management	290
Chief Information Officer	21
CIO ... <i>siehe</i> Chief Information Officer	
COBIT	<i>siehe</i> Control Objectives
for Information and Related	
Technology	
Configuration Management	292
Continuity Management	286
Control Objectives for Information and	
Related Technology	6
Cooperation Support	<i>siehe</i>
Kooperationsunterstützung	
CSM	<i>siehe</i> Customer Service
Management	
Customer Relationship Management	288
Customer Service Management	44

D

Datenbank	179
Datenbank-Sicht	180
Datenmodellierung	319
Datensatz	178
Decision/Execution Support	334
Degradierung <i>siehe</i> Dienstdegradierung	
Dienst	193
-Client	43
-Template	138
-übergabepunkt	198
-aktivierung	224
-anbieter	170
-anforderungen (funktional)	223
-bericht	210
-degradierung	204
-entwurf	224
-erbringepartei	198, 240
-hierarchie	189

-katalog 47, 196, 229
 -lebenszyklus 44, 76, 158, 195
 -management
 -integriert-kooperativ . 282, 353
 Rahmenwerk 135
 -messung 207, 256
 -modell 137
 -modularisierung 191
 -nehmer *siehe* Kunde
 -portfolio 45, 138, 196
 -verwalter 172
 -qualität 43
 -smanagement 137
 -sparameter 43
 -vereinbarung 3, 48, 202
 Entwurfsempfehlungen 159
 Initialisierung 243
 Verhandlung 243
 -verwaltung 221
 -zugangspunkt 43
 Änderung 225
 Außerbetriebnahme 229
 Erbringungsparameter ... 195, 203
 Leistungsparameter 195
 Leistungswert 40
 Modellbasierte Definition 41
 Nutzwert 40
 Systemtechnische Definition .. 40
 Wertorientierte Definition 40
 Dokument 177
 Dokumentation 178

E

EMANICS *siehe* European Network
 of Excellence for the Mana-
 gement of the Internet and
 Complex Services
 Enhanced Telecom Operations Map 147
 Entwicklungsrahmen 35
 Entwurfsrichtlinien 300
 eTOM *siehe* Enhanced Telecom
 Operations Map
 European Network of Excellence for the
 Management of the Internet
 and Complex Services ... 19
 Extensible Markup Language 128
 Externer Zulieferer *siehe* Lieferant

F

Funktionsbereiche 216
 Dienstinformationsmanagement
 221

Messdaten- und Berichtsmanage-
 ment 256
 Stakeholder-Management 237
 Vereinbarungsmanagement ... 242

G

Generisches XSD-Modul 316
 Geschäftsprozess 2, 46
 GOM. *siehe* Grundsätze ordnungsmäßi-
 ger Modellierung
 Graphical User Interface 361
 Grundsätze ordnungsmäßiger Modellie-
 rung 129
 GUI *siehe* Graphical User Interface

I

Incident Management 292
 Information Management *siehe*
 Informationsverwaltung
 Information Security Management . 286
 Informationsübermittlung 270, 350
 Informationsartefakt 177
 Informationsmodell 149
 Instanziierung 364
 Informationssystem 318
 Informationsverwaltung 333
 Interaktivitätsanalyse 304
 Interprozesskommunikation ... 282, 352
 Intraprozesskommunikation ... 267, 349
 ISO/IEC 20000 145
 IT Infrastructure Library 141
 IT-Management
 dienstorientiert 1
 geschäftsrorientiert 2
 IT-Provider *siehe* Dienstanbieter
 IT-Service-Management 1
 ITIL ... *siehe* IT Infrastructure Library
 ITSM ... *siehe* IT-Service-Management

K

Key Quality Indicator 158
 Kommunikationsmuster 268
 Konfliktanalyse 254
 Kontraktverwalter 172
 Kooperationsunterstützung 334
 Kunde 171, 197, 238

L

Lieferant 171

Lieferantenvertrag 45, 207, 251

M

Management-Informationsbasis 32
 Managementarchitektur 30
 Prozess-Sicht 167
 Funktionsmodell 216
 Informationsmodell 177
 Kommunikationsmodell 266
 Organisationsmodell 170
 System-Sicht 299
 Funktionsmodell 332
 Informationsmodell 313
 Kommunikationsmodell 348
 Organisationsmodell 301
 Managementobjekt 359
 Managementplattform 29, 357
 -Entwickler 304
 -Unterstützer 303
 Basisanwendungen 359
 Entwurf 361
 Grafische Oberfläche 361
 Informationsbaustein 359
 Infrastruktur 359
 Kernsystem 359
 Kommunikationsbaustein 359
 Komponenten 359
 Managementapplikationen 359
 Realisierung 358
 Managementsystem 28
 -Administrator 303
 integriert 357
 IT-gestützt 357
 Manager-Agent-Prinzip 32
 MDA . *siehe* Model Driven Architecture
 Messergebnisdatensatz 207
 Messspezifikation 207
 MIB *siehe* Management-
 Informationsbasis
 MNM-Dienstmodell *siehe* Dienstmodell
 Model Driven Architecture 25
 Modellentwurf 36
 Morphologische Analyse 54
 Multi-Level-SLAs 187

N

Nutzerschnittstelle 308

O

OLA *siehe* Operational Level
 Agreement

Operational Level Agreement *siehe*
 Operative Vereinbarung
 Operative Vereinbarung ... 46, 205, 246
 Ordnungsrahmen 32
 Organisationsdomänen 170

P

Plattform . *siehe* Managementplattform
 Policy *siehe* Richtlinie
 Problem Management 293
 Procedure *siehe* Verfahren
 Provider *siehe* Dienstanbieter
 Prozess 27
 Prozessmanager 172
 Prozessorientierung 25, 141
 Prozessrahmenwerk 282

Q

QoS *siehe* Quality of Service
 QoS-Parameter *siehe*
 Dienstqualitätsparameter
 QoSSL *siehe* Quality of Service
 Specification Language
 Quality of Service . *siehe* Dienstqualität
 Quality of Service Specification Language
 18

R

RACI-Diagramm 174
 Referenz-Workflow *siehe* Workflow, 220
 Release Management 291
 Richtlinie 27
 Rolle 171

S

SAP *siehe* Service Access Point
 Service *siehe* Dienst
 Service Access Point *siehe*
 Dienstzugangspunkt
 Service Information Specification Lan-
 guage 17
 Service Level Agreement *siehe*
 Dienstvereinbarung
 Service Monitoring Architecture 17
 Service Template Model *siehe*
 Dienst-Template
 Service-Level 48
 Service-Level-Management 3
 Analyse- und Bewertungsschema
 135

Anforderungen *siehe*
Anforderungsanalyse
Architektur *siehe*
Managementarchitektur
Bestehende Ansätze 131
Betrachtungsbereiche 216
Forschungsprojekte 19
Grundannahmen 75
Grundkonzepte 181
Zielsetzung 74
Service-Level-Manager 172
Shared Information/Data Model... 148,
355
SID *siehe* Shared Information/Data
Model
Signalisierung 269, 349
SISL *siehe* Service Information
Specification Language
SLA *siehe* Service Level Agreement
SLA Parameter Framework 158
SLA-Verletzung 211
SLM .. *siehe* Service-Level-Management
SMONA *siehe* Service Monitoring
Architecture
SQL .. *siehe* Standard Query Language
Standard Query Language 361
Standard-Dienstvereinbarung .. 243, 345
Standardisierter Dialog 270, 351
Stereotyp 192
Steuer-Frontend 308
Supplier Relationship Management . 289
Support-Level-Profil 200, 232
Supportive Agreement *siehe* Unterstüt-
zende Vereinbarung
System 28
Systembenutzerrollen 302
Systemorientierung 27

T

Teilergebnisse 15
Tool Support *siehe*
Werkzeugunterstützung
Trägersystem 358

U

UC *siehe* Underpinning Contract
UML *siehe* Unified Modeling Language
Underpinning Contract *siehe*
Lieferantenvertrag
Unified Modeling Language 128
Unterstützende Vereinbarung 205

V

V-Modell 303
Verantwortungsmodelle 174
Vereinbarungskonflikt 213
Verfahren 27
Vollautomatisierung 334
Vorgehensmodell 11

W

Werkzeugunterstützung 154
Workflow 168
-Management-System 363
-Steuerung 334
Automatisierbarkeitsanalyse . 339
Workflow Control *siehe*
Workflow-Steuerung

X

XML *siehe* Extensible Markup
Language
XML Schema 314
Datentypen 315
XML Schema Definition *siehe* XML
Schema
XSD *siehe* XML Schema Definition

Z

Zugriffskontrolle 308