

Gestaltung und Verbesserung von Services
unter besonderer Beachtung von Cloud Computing
und Serviceorientierten Architekturen

DISSERTATION

der Universität St. Gallen,

Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften

sowie Internationale Beziehungen (HSG)

zur Erlangung der Würde eines

Doktors der Wirtschaftswissenschaften

vorgelegt von

Johannes Maximilian Ahrens

aus Deutschland

Genehmigt auf Antrag der Herren

Prof. Dr. Robert Winter

und

PD Dr. Joachim Schelp

Dissertation Nr. 4442

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

St. Gallen, den 02. November 2015

Der Rektor:

Prof. Dr. Thomas Bieger

Vorwort

Die vorliegende Arbeit entstand über einen längeren Zeitraum, während dem ich zunächst für die Deutsche Telekom und später in einem von mir gegründeten Unternehmen arbeitete. Während dieser Jahre wurde ich von vielen Kollegen, Freunden und Projektpartnern unterstützt – bei ihnen allen möchte mich an dieser Stelle herzlich bedanken.

Zunächst gilt mein Dank Herrn Prof. Dr. Robert Winter, der mich nicht nur inhaltlich intensiv betreut hat, sondern mich auch motivierte, die Arbeit trotz der anderweitigen Verpflichtungen abzuschliessen. Weiterhin möchte ich mich für die Unterstützung von PD Dr. Achim Schelp und Prof. Dr. Stephan Aier vom Institut für Wirtschaftsinformatik der Universität St. Gallen (IWI-HSG) bedanken. Dr. Schelp hat freundlicherweise das Ko-Referat übernommen und mir in mehreren Feedbackrunden sehr hilfreiche Hinweise gegeben. Dr. Aier hat mir den Zugang zum IWI-HSG vermittelt und mich darüber hinaus motiviert und angetrieben, diese Arbeit fertigzustellen. Zusätzlich hat er – gemeinsam mit Dr. Matthias Stutz – einen der im Weiteren aufgeführten Beiträge mit mir verfasst.

Neben den Kollegen am IWI-HSG möchte ich mich bei meinen Kollegen und Projektpartnern während meiner Zeit bei den Deutsche Telekom Laboratories (T-Labs) bedanken. Die T-Labs haben mir eine Arbeitsumgebung geboten, welche es mir ermöglichte, einen Großteil der inhaltlichen Grundlagen für diese Arbeit zu schaffen. Dr. Udo Bub gilt besonderer Dank, da er mich zu den T-Labs gebracht hat und mich anregte, eine Promotion zu beginnen. Dr. Philipp Offermann und Dr. Christian Schröpfer haben als Projektpartner mit mir einige der eingereichten Beiträge publiziert. Dr. habil Marten Schönherr hat mir über die gesamte Zeit mit Ratschlägen geholfen und ist mir über die Jahre zum Freund geworden.

Meinen Eltern danke ich, dass sie mich immer unterstützt haben und dadurch die akademische Ausbildung ermöglichten. Ganz besonders danke ich meinem Schwiegervater Prof. Dr. Hans-Jürgen Weissbach für die ausdauernden Korrekturarbeiten. Auch hat die Dissertation viele Urlaube und Wochenenden verschlungen, während denen meine Frau und meine Kinder auf mich verzichten mussten – dafür gilt meiner Familie der größte Dank.

Berlin, im November 2015

Johannes Maximilian Ahrens

Inhaltsübersicht

Vorwort	iii
Inhaltsübersicht	iv
Inhaltsverzeichnis	vi
Abbildungsverzeichnis.....	xii
Tabellenverzeichnis.....	xiii
Abkürzungsverzeichnis	xv
Kurzfassung	xx
Abstract	xxi
Teil A - Dachbeitrag.....	1
1 Motivation	1
2 Zielsetzung dieser Arbeit.....	7
3 Stand der Forschung und Forschungslücke	13
4 Inhaltliche Grundlagen	26
5 Forschungsbeitrag dieser Dissertation.....	30
6 Diskussion und weiterer Forschungsbedarf	68
Teil B – Einzelbeiträge	74
1 Beitrag I: Entwurf eines Enterprise Architecture Frameworks für Serviceorientierte Architekturen	74
2 Beitrag II: Integriertes Performance Management als Teil des Enterprise Architecture Management	89
3 Beitrag III: Deriving SOA Evaluation Metrics in an Enterprise Architecture Context.....	101
4 Beitrag IV: A Flexible Approach to Service Management-Related Service Description in SOAs	111
5 Beitrag V: Cloud Computing and the Impact on Enterprise IT	133
Referenzen	I

Appendix A – Metamodell Spezialisierung	XIII
Appendix B – Ergebnisse der Literaturrecherche.....	XIV
Lebenslauf.....	XLV

Inhaltsverzeichnis

Vorwort	iii
Inhaltsübersicht	iv
Inhaltsverzeichnis	vi
Abbildungsverzeichnis.....	xii
Tabellenverzeichnis.....	xiii
Abkürzungsverzeichnis	xv
Kurzfassung	xx
Abstract	xxi
Teil A - Dachbeitrag.....	1
1 Motivation	1
1.1 Praxisbeispiel 1 – Probleme des Service Design und Management	1
1.2 Praxisbeispiel 2 – Probleme der Bedarfsgerechtigkeit der Services	3
1.3 Forschungsfeld: Services und deren Nutzung in Unternehmen	5
1.4 Relevante Problemstellung	6
2 Zielsetzung dieser Arbeit.....	7
2.1 Eingrenzung des Forschungsgegenstandes.....	7
2.2 Konkretisierung der Forschungsfrage	10
2.3 Inhaltliche Struktur der Arbeit.....	11
2.4 Forschungsmethode	12
3 Stand der Forschung und Forschungslücke	13
3.1 Servicegestaltung	19
3.2 Serviceverbesserung	22
3.3 Forschungslücke.....	24
4 Inhaltliche Grundlagen	26
4.1 Basisdefinition Service-orientierte Architekturen	27
4.2 Basisdefinition Cloud Computing	27

4.3	Basisdefinition IT-Servicemanagement	28
4.4	Business Engineering Business Metamodell	29
5	Forschungsbeitrag dieser Dissertation.....	30
5.1	Forschungsfrage 1	32
5.2	Forschungsfrage 2	34
5.3	Entwicklung eines umfassenden Servicemodells	36
5.3.1	Existierende Definitionen und Modelle des Servicebegriffes	37
5.3.2	Rekonstruktion eines SOA-Servicemodells	38
5.3.3	Rekonstruktion eines ITSM-Servicemodells	43
5.3.4	Rekonstruktion eines Cloud Computing-Servicemodells	48
5.3.5	Konsolidiertes Service-Metamodell.....	54
5.4	Zusammenfassung der Beiträge aus Teil B.....	63
5.4.1	Beitrag I – Entwurf eines Enterprise Architecture Frameworks für serviceorientierte Architekturen.....	64
5.4.2	Beitrag II – Integriertes Performance Measurement als Teil des Enterprise Architecture Management	65
5.4.3	Beitrag III – Deriving SOA Evaluation Metrics in an Enterprise Architecture Context.....	66
5.4.4	Beitrag IV – A Flexible Approach to Service Management-Related Service Description in SOAs	67
5.4.5	Beitrag V - Cloud Computing and the Impact on Enterprise IT	68
6	Diskussion und weiterer Forschungsbedarf	68
6.1	Anwendbarkeit der Forschungsergebnisse.....	68
6.2	Kritische Würdigung.....	69
6.2.1	Evaluation der Forschungslücke (Eval 1).....	70
6.2.2	Evaluation des Gestaltungsentwurfes (Eval 2)	70
6.2.3	Evaluation des Artefakts in einem Testaufbau (Eval 3)	71
6.2.4	Evaluation des Artefakts in einer realen Implementierung (Eval 4)	71

6.2.5 Zusammenfassende kritische Würdigung	72
6.3 Weiterer Forschungsbedarf	72
6.3.1 Methodenartefakte aus der Nutzersicht darstellen und dezidiert evaluieren.....	72
6.3.2 Evaluation sämtlicher Artefakte in einer realen Implementierung (Eval 4).....	73
6.3.3 Integration in ein etabliertes Framework	73
Teil B – Einzelbeiträge	74
1 Beitrag I: Entwurf eines Enterprise Architecture Frameworks für Serviceorientierte Architekturen	74
1.1 Abstract.....	74
1.2 Einleitung	74
1.2.1 Ziel des Artikels.....	74
1.2.2 Methode	75
1.2.3 Gliederung.....	75
1.3 Enterprise Architecture Frameworks.....	75
1.3.1 Das Ziel von Enterprise Architecture Frameworks.....	75
1.3.2 Existierende Frameworks.....	76
1.3.3 GERAM	76
1.3.4 GERAM und serviceorientierte Architekturen	79
1.4 Serviceorientierte Architekturen	79
1.4.1 Ziel von serviceorientierten Architekturen	79
1.4.2 Anforderungen einer SOA an Elemente der GERAM	80
1.5 UN/CEFACT Modeling Methodology für SOA	81
1.5.1 Die Methodologie	82
1.5.2 Kritik	83
1.5.3 Erweiterungen der UMM	83
1.5.4 Einordnung der Erweiterung in GERAM	85

1.6	Schlussfolgerungen	88
1.6.1	Erweiterung der UMM	88
1.6.2	Enterprise Architecture Frameworks für serviceorientierte Architekturen	88
2	Beitrag II: Integriertes Performance Management als Teil des Enterprise Architecture Management.....	89
2.1	Abstract.....	89
2.2	Einführung.....	89
2.2.1	Ziel des Artikels.....	89
2.2.2	EA Management	90
2.2.3	SOA.....	91
2.3	Performance Management und Performance Measurement.....	91
2.3.1	Strategische Performance Measurement Konzepte	92
2.3.2	Operative Performance Measurement Konzepte.....	93
2.4	Integriertes Measurement Model	94
2.4.1	IT Performance	95
2.4.2	Gestaltung der IT-Performance Ebene	95
2.4.3	Integriertes Modell.....	97
2.5	Fazit und Ausblick.....	99
3	Beitrag III: Deriving SOA Evaluation Metrics in an Enterprise Architecture Context	101
3.1	Abstract.....	101
3.2	Introduction	101
3.3	Conceptual Foundations	102
3.4	Related Work	103
3.5	Method Construction	104
3.6	Application of the Method	106
3.7	Summary and Outlook	110

4 Beitrag IV: A Flexible Approach to Service Management-Related Service	
Description in SOAs.....	111
4.1 Abstract.....	111
4.2 Introduction	111
4.3 Requirements for Service Description	113
4.3.1 Requirements Overview.....	113
4.3.2 Information Relevant for Service Lifecycle Management.....	113
4.3.3 QoS Guarantees	114
4.4 Related Work	117
4.4.1 Standards for Service Description.....	117
4.4.2 QoS Specific Standards – UML Profile for QoS.....	117
4.4.3 Approaches to Semantic Service Description, Discovery, and Selection	118
4.5 Extension of OWL-S	120
4.5.1 Extension for Service Lifecycle Management	120
4.5.2 Extension for QoS with UML Profile for QoS Description	124
4.6 Service Management Prototype.....	127
4.6.1 Overview – Architecture and Functionality.....	127
4.6.2 Functions and Methodology of the Prototype.....	129
4.7 Importance for Matching, SLA Negotiation and Enforcement.....	131
4.8 Conclusion and Outlook.....	132
5 Beitrag V: Cloud Computing and the Impact on Enterprise IT	133
5.1 Abstract.....	133
5.2 Introduction	133
5.3 Cloud Computing	133
5.4 Enterprise IT in the Cloud – The Challenges	134
5.5 Enterprise IT in the Cloud – The Benefits	136
5.5.1 Use Case 1: Solution Introduction in Enterprises.....	136

5.5.2 Use Case 2: Peak Pooling by Design	137
5.6 Next Generation Infrastructures and the Cloud	140
5.6.1 SLA / Consistency	140
5.6.2 Security Zones	140
5.7 Conclusion	141
Referenzen	I
Appendix A – Metamodell Spezialisierung	XIII
Appendix B – Ergebnisse der Literaturrecherche	XIV
Lebenslauf	XLV

Abbildungsverzeichnis

Abbildung 1: Inhaltliche Struktur der Arbeit	11
Abbildung 2: Analyse Rahmen für die konstruktionsorientierte Forschung	13
Abbildung 3: Schritte der Literaturrecherche (aus (vom Brocke u. a., 2009))	14
Abbildung 4: Eingrenzung des Trefferspektrums der Beiträge zur Servicegestaltung.....	20
Abbildung 5: Eingrenzung des Trefferspektrums von Beiträgen zur Serviceverbesserung	22
Abbildung 6: Metamodell SOA Service	43
Abbildung 7: Metamodell ITSM Service	48
Abbildung 8: Metamodell Cloud Service	54
Abbildung 9: BE Business-Metamodell Ebenen.....	55
Abbildung 10: BE Business-Metamodell Ausschnitt.....	56
Abbildung 11: Ausschnitt des Metamodells Service auf der Alignmentebene	58
Abbildung 12: Ausschnitt des Metamodells Services auf den Technologieebenen ..	60
Abbildung 13: BE Business Metamodell Service Spezialisierung.....	61
Abbildung 14: Einfache Darstellung des Servicebegriffs.....	62
Abbildung 15: Evaluationsschritte (nach (Sonnenberg & vom Brocke, 2012)).....	70
Abbildung 16: GERAM Elemente	77
Abbildung 17: GERA Lebenszyklusphasen.....	78
Abbildung 18: Beispiel des UMM Business-Process-View.....	82
Abbildung 19: Geschäftsprozessmodell mit Verfeinerung.....	84
Abbildung 20: Geschäftsprozessmodell in der BPMN.....	85
Abbildung 21: Kombiniertes top-down und bottom-up Vorgehen im Rahmen des PPM.....	94
Abbildung 22: Dekomposition eines IT-Services (Zarnekow u. a., 2005).....	97
Abbildung 23: Integriertes Modell.....	99
Figure 24: Enterprise Architecture, layers and services (Winter & Fischer, 2006) ..	103

Figure 25: Example GQM abstraction sheet (Latum u. a., 1998)	105
Figure 26: Application of the Goal/Question/Metric method in the context of SOA.	105
Figure 27: Adopted GQM method, additional steps in shaded grey.....	106
Figure 28: Case study goal hierarchy	107
Figure 29: Simplified abstraction sheet	108
Figure 30: GQM EID hierarchy	108
Figure 31: Example questionnaire for measuring the customization level	110
Figure 32: Definition of <i>ServiceVersion</i> in OWL-S	123
Figure 33: Instance of a service description for CalculateRoute with details for <i>ServiceVersion</i>	124
Figure 34: Example QoS requirements in a UML Activity diagram	125
Figure 35: QoS offering for GetCreditService in the service description.....	126
Figure 36: Overview of service management prototype.....	128
Figure 37: Cost comparison for a small cloud use case.....	138
Figure 38: As-is situation for peak pooling	138
Figure 39: To-be situation	139

Tabellenverzeichnis

Tabelle 1: Suchbegriffe für die Literaturrecherche	17
Tabelle 2: Zusammenfassung des Forschungsstandes.....	25
Tabelle 3: Übersicht der Forschungsbeiträge	31
Tabelle 4: Typen von Services aus der SOA Literatur	39
Tabelle 5: Attribute des SOA Servicemodells	41
Tabelle 6: Servicetypen aus der ITSM Literatur	44
Tabelle 7: Entitäten des ITSM Services	45
Tabelle 8: Stakeholder von ITSM-Services	47
Tabelle 9: CC-Servicetypen	49

Tabelle 10: Attribute von CC Services	51
Tabelle 11: Stakeholder von Cloud Services	53
Tabelle 12: Titel und Forschungsmethode der Beiträge	64
Tabelle 13: Erweiterte UMM im Rahmen von GERAM	87
Table 14: QoS information	115
Table 15: Defined elements for service lifecycle management	121

Abkürzungsverzeichnis

Abkürzung	Erläuterung
AIX	Advanced Interactive Execution
ARIS	Architektur integrierter Informationssysteme
ASP	Application Service Provisioning
BE	Business Engineering
BPEL	Business Process Execution Language
BPMN	Business Process Modeling Notation
BSC	Balanced Scorecard
CAPEX	Capital Expenditures
CC	Cloud Computing
CEFACT und UN / CEFACT	United Nations / Centre for Trade Facilitation and Electronic Business
CIMOSA	Computer Integrated Manufacturing Open System Architecture
CMMI-SVC	Capability Maturity Model Integration for Services
CobIT	Control Objectives for Information and Related Technology
CRM	Customer Relationship Management
CSI	Continual Service Improvement
DAML	Darpa Agent Markup Language
DoDAF	Department of Defense Architecture Framework
DSD	DIANE Service Description

Abkürzung	Erläuterung
EA	Enterprise Architecture
EAF	Enterprise Architecture Framework
EAI	Enterprise Application Integration
EAM	Enterprise Architecture Management
EDV	(veraltet für) IT
EEM	Enterprise Engineering Methodology
EET	Enterprise Engineering Tools
e.g.	for example (exemplum gratiae)
EID	Extended Influence Diagram
EML	Enterprise Modeling Language
EMO	Enterprise Modules
ENAPS	European Network of Advanced Performance Studies
EOS	Enterprise Operational System
FIS	Financial Information System
FF	Forschungsfrage
GEMC	Generic Enterprise Modeling Concept
GERAM	Generalised Enterprise Reference Architecture
GERAM	Generalised Enterprise Reference Architecture and Methodology
GRAI - GIM	Graphe Résultats et Activités Interreliées – GRAI Integrated Methodology

Abkürzung	Erläuterung
GQM	Goal Question Metric
IFAC	International Federation of Automatic Control
IFIP	International Federation for Information Processing
ISO	International Organization for Standardization
IT	Information Technology, Informationstechnik
ITIL	IT-Infrastructure Library
ITSM	IT-Servicemanagement
ITUP	IBM Tivoli Unified Process
JUDDI	Java Universal Description, Discovery and Integration
LNCS	Lecture Notes on Computer Science
MDD	Model-driven Design
MOF	Microsoft Operations Framework
NAICS	North American Industrial Classification System
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OCL	Object Constraint Language
OLA	Operations-Level Agreement
OMG	Object Management Group
OPEX	Operational Expenditures
OWL-S	Web Ontology Language for Services

Abkürzung	Erläuterung
OWLS-MX	OWL-S Hybrid semantic matchmaking
PEM	Partial Enterprise Model
PERA	Purdue Enterprise Reference Architecture
PPM	Performance Measurement System
QoS	Quality of Service
RDQL	Resource Description Framework Query Language
RDF	Resource Description Framework
RMI	Remote Method Invocation
SaaS	Software as a Service
SERVQUAL	Service und Qualität
SG	Servicegestaltung
SLA	Service Level Agreement
SLAang	Service Level Agreement definition language
SOA	Service-oriented Architecture
SQL	Structured Query Language
SRI	Stanford Research Institute
SV	Serviceverbesserung
TCO	Total Costs of Ownership
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modeling Language

Abkürzung	Erläuterung
UMM	UN/CEFACT Modeling Methodology
UN	United Nations
UNSPSC	United Nations Standard Product and Services Classification
V (+Zahl)	Version (Nr.)
VHB	Verband der Hochschullehrer für Betriebswirtschaft
WI	Wirtschaftsinformatik
WS	Web Services
WSDL	Web Services Description Language
WSDL-S	WSDL-S with Semantic Extension
WSLA	Web Services Level Agreement
WSMF	Web Services Modeling Framework
WSMO	Web Services Modeling Ontology
WSOL	Web Services Offerings Language
W3C	Worldwide Web Consortium
XML	Extensible Markup Language
X86	Bezeichnung einer Mikroprozessor-Architektur

Kurzfassung

Viele Unternehmen sind heute mit der praktischen Umsetzung unterschiedlicher Facetten des Serviceparadigmas konfrontiert. Projekte zur Einführung unterschiedlicher Services im Rahmen der Implementierung einer Serviceorientierten Architektur (SOA) oder von Cloud Computing (CC) Services erreichen aber häufig nicht die gesteckten Ziele. Diese Problematik kann auftreten, wenn einerseits die angewandten Methoden nicht für den spezifischen Servicetyp geeignet sind und andererseits die existierenden Methoden Schwächen aufweisen.

Mit dieser Problematik beschäftigt sich die vorliegende Dissertation. Sie beantwortet die Frage, welche Servicetypen unter Beachtung der vielfach diskutierten Paradigmen wie SOA, CC oder IT-Service-Management (ITSM) existieren und in welchem semantischen Verhältnis diese Services zueinander stehen. Dazu rekonstruiert die Dissertation ein umfassendes Servicemodell und integriert dieses in das St. Galler Business Engineering Business-Metamodell. Der zentrale Beitrag dieser Dissertation bezieht sich auf die Frage, welche Methoden für die Gestaltung und Verbesserung eines Services notwendig sind. Zur Beantwortung dieser Frage werden eine Reihe von Methodenfragmenten sowie ein Modell vorgestellt. Damit die Fragestellung im Rahmen dieser Arbeit umfassend behandelt werden kann, beschränkt sich die Dissertation dabei auf einen eng umgrenzten Servicetyp. Sowohl bei der Fragestellung nach der Gestaltung von Services als auch der Verbesserung von Services beziehen die entwickelten Methoden funktionale und nicht-funktionale Aspekte mit ein. Damit füllt die Arbeit eine spezifische Forschungslücke der Methodenunterstützung bei der Gestaltung von Services.

Als weiterer wesentlicher Aspekt wird in der vorliegenden Arbeit die Frage gestellt, inwiefern die Methoden für Gestaltung und Verbesserung von Services eine doppelte Perspektive, nämlich sowohl die spezifische Sichtweise des Servicekonsumenten als auch die des Serviceproduzenten einnehmen müssen. Auf diese Frage kann die Dissertation keine umfassende Antwort geben. Jedoch wird aufgezeigt, dass für Konsumenten und Produzenten eines Services deutlich unterschiedliche Aspekte im Vordergrund stehen und so eine Erweiterung der Methoden einem undifferenzierten Ansatz überlegen sein sollte.

Stichwörter: *Servicegestaltung, Serviceverbesserung, Service-Metamodell, Serviceorientierte Architektur, Cloud Computing, IT-Service-Management, Methode, Methodenkonstruktion*

Abstract

Many companies are confronted with different aspects of the service paradigm. Projects that focus on the introduction of services into the enterprise's information technology (IT) landscape, as for example the implementation of a service-oriented architecture (SOA) or the integration of cloud computing (CC) services, often fail to achieve the goals previously defined. This may result from the insufficient methodological support of these projects due to shortcomings of the existing methods or usage of methods that are not fitting to the services addressed.

This challenge is being addressed by this dissertation that supplies answers to the question what types of services exist within the various paradigms (SOA, CC, IT Service Management) and how their respective relationship could be semantically expressed. This is done by the reconstruction of a complete and detailed meta-model of the various service types, including the integration of that model into the Business Engineering Business Metamodel, which is developed and maintained at the University of St. Gallen. Beyond this, the main contributions of this dissertation are method fragments and models supporting the design and improvement of a specific service type. These artifacts address a research gap by including functional and non-functional aspects during the phases of design and improvement of the service lifecycle.

Furthermore in this dissertation addresses the question whether methods for service design and improvement shall explicitly include a service consumer perspective in addition to the usual provider perspective. A completely satisfactory answer cannot be provided within the work presented. Nevertheless research results indicate that due to the different viewpoint the two stakeholders have on a specific service, methods addressing both perspectives should deliver superior results.

Keywords: *Service Design, Service Improvement, Service Metamodel, Service-oriented Architecture, Cloud Computing, IT Service Management, Method, Method construction*

Teil A - Dachbeitrag

1 Motivation

Das Serviceparadigma wird in vielen Unternehmen in unterschiedlicher Form realisiert. Ihm wird ein erheblicher Einfluss auf den Unternehmenserfolg zugeschrieben. Services sollen die Innovationsfähigkeit (Agarwal, Choi, Ramamurthy, Selen, & Selim, 2012) und die Flexibilität der Unternehmen erhöhen sowie deren Kostenstrukturen optimieren (Svejvig, Storgaard, & Møller, 2013). Zu den bekannten Serviceparadigmen zählen Serviceorientierte Architekturen (SOA), Software Services, Microservices und Cloud Services¹. Konzepte wie SOA werden teilweise schon länger in der Forschung diskutiert und in der Praxis der Unternehmen implementiert (McCoy & Natis, 2003). Andere Servicekonzepte sind erst in den letzten Jahren intensiv diskutiert worden wie beispielsweise das Cloud Computing (CC) (McDonald & Aron, 2011). Diese Arbeit beschäftigt sich mit Methoden der Gestaltung und dem Management solcher Services in IT-Systemen von Unternehmen. Viele Projekte, in deren Rahmen ein Serviceparadigma implementiert werden soll scheitern, da die fachlichen Ziele nicht konsistent mit den technischen Möglichkeiten sind oder der Bedarf der Servicenutzer nicht ausreichend erhoben wurde (Brown, 2007). Als Erfolgsfaktor wird eine starke Einbindung in die Strukturen des Unternehmens genannt, womit Unterstützung des Top-Managements, Einbindung in die Unternehmenskultur und auch Integration von wichtigen Zulieferern gemeint sind (Tan, Cater-Steel, & Toleman, 2009). Als Einstieg und zur Verdeutlichung der Relevanz des Forschungsbeitrags dieser Arbeit finden sich im Folgenden zwei Beispiele, die exemplarisch aufzeigen, wie eine Umsetzung des Serviceparadigmas in der Praxis scheitern kann. Dabei erfolgten bestimmte Aspekte der Servicegestaltung (insbesondere Beispiel 1) sowie der Evaluation und Verbesserung der Services (insbesondere Beispiel 2) nicht mithilfe einer Methode, was dazu führte dass die bereitgestellten Servicearchitekturen die avisierten Ziele nicht erfüllten.

1.1 Praxisbeispiel 1 – Probleme des Service Design und Management

Bei diesem Beispiel handelt es sich um ein umfangreiches Programm zur Gestaltung einer SOA, welche die Unterstützungsprozesse von Telekommunikationsprodukten

¹ Abschnitt 4 enthält Definitionen der im Weiteren betrachteten Serviceparadigmen.

(z.B. die Abrechnung der Dienstleistung) in Form von Services automatisieren sollte. Das Projekt fand bei einem großen international agierenden Telekommunikationskonzern statt und wurde als Programm strukturell in der Organisationseinheit für die Informationstechnologie (IT) verankert sowie mit umfangreichen Mitteln ausgestattet. Das Projekt wurde direkt von einem Vorstand des Unternehmens verantwortet.

Ziel war eine Reduktion der notwendigen Zeit für die Bereitstellung der Unterstützungsprozesse für neue Produkte zur Stärkung der Innovationsfähigkeit. Dies sollte durch Wiederverwendung von Services durch andere Prozesse ermöglicht werden. Damit verbunden war das Ziel einer effizienteren Produktion der gesamten Unterstützungsprozesse, da somit die Kosten für die Entwicklung und den Betrieb der Services auf verschiedene Produkte hätten verteilt werden können. In einem top-down-Vorgehen wurden anhand von Referenzprozessmodellen und den aktuell implementierten Produktionsprozessen die relevanten Services identifiziert und definiert. Dabei entstand ein umfassendes Serviceportfolio.

Dieses strukturierte Vorgehen führte dennoch nicht zu dem erhofften Erfolg. Die gestalteten Services wurden nur in sehr geringem Umfang für unterschiedliche Produkte genutzt. Insbesondere die vielfach notwendige Einbindung unterschiedlicher Lieferanten wurde bei der Servicegestaltung nicht ausreichend betrachtet. Daher war es für viele Produkte effizienter, nicht einen Service mit einem anderen Produkt gemeinsam zu nutzen, sondern vielmehr einen auf die eigenen Bedürfnisse ausgelegten Service zu entwickeln. Auch das Ziel der vereinfachten Einführung neuer Produkte durch die Wiederverwendung bestehender Services wurde nur teilweise erreicht. Die Wiederverwendung scheiterte häufig, da die nicht-funktionalen Anforderungen - wie zum Beispiel der möglichen Anzahl der Transaktionen pro Minute oder Sicherheitsanforderungen - der verschiedenen Kontexte, in die der Service eingebunden werden sollte, zu divergent waren. Diese Anforderungen wurden bei dem gewählten Vorgehen der Gestaltung der Services nicht ausreichend beachtet, waren aber bei der Entscheidung über eine mögliche Wiederverwendung von zentraler Bedeutung. Aus diesen Gründen entstand in der Gesamtbetrachtung durch die Serviceorientierung kein Effizienzvorteil. Vielmehr erzeugte die SOA einen weiteren „Overhead“, da für ihre Steuerung eine Governance-Struktur angelegt wurde, die zusätzliche Anforderungen an die Entwicklung der Services – auch der nicht durch mehrere Produkte genutzten – stellte.

Dieses erste Beispiel zeigt auf, inwiefern das Fehlen einer methodischen Unterstützung zur Gestaltung von Services den Gesamterfolg eines Projektes beeinträchtigen kann. Obschon das Projekt organisatorisch und kulturell in dem Unternehmen gut eingebunden war, wurden bei der konkreten Servicegestaltung wichtige Aspekte – wie die nicht-funktionalen Anforderungen – nicht ausreichend betrachtet.

1.2 Praxisbeispiel 2 – Probleme der Bedarfsgerechtigkeit der Services

Die IT-Abteilung eines Softwareherstellers ist für die Bereitstellung von IT-Services für sämtliche Fachabteilungen verantwortlich. Diese zentrale IT-Abteilung hat dabei die Zielsetzung, die Datensicherheit zu garantieren sowie eine effiziente und effektive Unterstützung der Fachbereiche zu ermöglichen. Das Servicespektrum reicht dabei von der Bereitstellung und dem Betrieb ganzer Applikationen für Kunden des Softwarehauses als Software as a Service (SaaS)² bis zu kleinteiligen Diensten wie der Bereitstellung von Infrastrukturdiensten. Die Herausforderung für die IT-Abteilung bestand darin, Services sicher und effizient zu produzieren, dabei aber nicht so stark auf die abteilungsinterne Optimierung der IT-Abteilung zu fokussieren, dass eine effektive Unterstützung der Fachabteilungen nicht mehr gewährleistet werden konnte. In diesem Unternehmen ist das nicht vollumfänglich gelungen.

So stellte die IT-Abteilung zu einem bestimmten Zeitpunkt fest, dass viele Abteilungen in der Softwareentwicklung Infrastrukturdienste für die Speicherung von Daten und den Betrieb von Serversystemen am externen Markt erworben hatten. Diese „Schatten-IT“ hatte unbemerkt eine erhebliche Größenordnung erreicht und war daher zunächst der Finanzabteilung aufgefallen. Da den Fachabteilungen durch die IT-Governancestruktur Entscheidungsfreiheit für die Auswahl ihrer Services gegeben worden war, stellte die Nutzung externer Dienstleister ein regelkonformes Verhalten seitens der Fachabteilungen dar. Jedoch gab es zwei wesentliche Probleme im Zusammenhang mit der Nutzung der externen Dienstleister: Einerseits war die IT-Abteilung nicht in der Lage, die Sicherheitsanforderungen des Unternehmens durchzusetzen und andererseits stellte sich die Nutzung der externen Dienstleister als nicht kosteneffizient dar.

Die Hauptmotivation der Fachabteilungen, externe Dienstleister einzusetzen, waren die funktionalen Mängel bei den intern produzierten Services. So gab es beispiels-

² Eine Definition des SaaS-Modells und anderer Cloud-Services findet sich in Abschnitt 4.

weise keine programmierbaren Schnittstellen, die sich zwar zum Marktstandard entwickelt hatten, jedoch von der IT-Abteilung als nicht-relevantes Servicemerkmal explizit ausgeschlossen worden waren. Weiterhin wurden die nicht-funktionalen Anforderungen an die Dienste nicht ausreichend dargestellt, so dass sich bei zunächst vergleichbar erscheinenden externen Diensten im späteren Nutzungsverlauf erhebliche Mehrkosten entwickelten. So waren die intern verwendeten Serversysteme deutlich leistungsfähiger, was bei Einsatz externer Dienstleister durch einen erhöhten Bedarf an Serversystemen kompensiert werden musste. Auch musste die Einbindung der Serversysteme in die Sicherheitsarchitektur durch weitere externe Dienstleister ermöglicht werden, was wiederum eine Erhöhung der Kosten zur Folge hatte.

Dieser Fall zeigt, dass mangelnde Methodenunterstützung bei der Verbesserung von Services sowohl im Hinblick auf qualitativ-funktionale Aspekte als auch bei den nicht-funktionalen Anforderungen zu einer suboptimalen Ressourcenallokation führt. Das dynamische Marktumfeld erfordert von dem Unternehmen eine schnelle Reaktionsfähigkeit auf Marktbedürfnisse als wesentlichen Erfolgsfaktor. Daher wurde die Governancestruktur so ausgelegt, dass die Fachabteilungen entsprechende Freiheitsgrade hatten. Die IT-Abteilung muss daher Services gestalten, die mit den Angeboten des externen Marktes konkurrieren und dabei dennoch die internen Anforderungen besser und effizienter umsetzen können. Hierfür ist eine kontinuierliche Anpassung des Servicespektrums notwendig.

Beide Praxisbeispiele zeigen die Komplexität der Umsetzung von Serviceparadigmen in der Unternehmenspraxis auf. Obwohl in beiden Fällen nicht unstrukturiert vorgegangen wurde, zeigte sich, dass nicht nur ein methodisches Vorgehen notwendig ist, sondern diese Methode auch in die Strukturen des gesamten Unternehmens eingebunden sein muss. Weiterhin machen die Beispiele deutlich, dass Services in vielen Technologien und Granularitäten existieren und auf unterschiedliche Weise bezogen werden können, so dass vielfach unklar ist, auf welche Art von Service ein bestimmtes Vorgehen ausgelegt ist.

Zur Bewältigung dieser Herausforderungen versucht die vorliegende Arbeit einen Beitrag zu leisten, indem sie Methodenfragmente für die Gestaltung und Verbesserung von Services entwickelt. Zur besseren Einordnung dieser Methoden in den Unternehmenskontext wird im Dachbeitrag dieser Arbeit ein Metamodell unterschiedli-

cher Servicetypen präsentiert, welches in ein ganzheitliches Metamodell eines Unternehmens integriert wird.

1.3 Forschungsfeld: Services und deren Nutzung in Unternehmen

Das Bestreben, mit Hilfe von Komponentisierungsmechanismen eine bessere Kontrolle über komplexe Strukturen zu gelangen, wird in der Informatik und Wirtschaftsinformatik schon lange diskutiert. Seit den 1970er Jahren wurden unterschiedliche Ansätze entwickelt, die vor allem von der Technologie sowie teilweise auch vom Wandel der Unternehmensorganisation getrieben wurden.

Angefangen hat diese Entwicklung mit der Strukturierung von Software durch Methoden und Technologien wie der Objektorientierten Softwareentwicklung (Rumbaugh, Blaha, Premerlani, Eddy, & Lorenzen, 1991). Primäres Ziel dieses Programmierparadigmas ist die Strukturierung von Software in beherrschbare, sinnvoll unterteilte Teile. Der nächste Schritt in der Softwareentwicklung war das Konzept der verteilten objektorientierten Anwendungen (Distributed Object Computing) (Vinoski, 1993). Dies sollte ermöglichen, auch verteilte Systeme einfacher zu vernetzen und somit Medienbrüche zwischen Softwaresystemen zu unterbinden. Diese grundlegenden Konzepte zur Kapselung von Teilsystemen und verteilter Ausführung der Teilsysteme wurden in der Folge durch verschiedene Generationen der Technologienentwicklung aufgegriffen. Es entwickelten sich in der Softwaretechnik Paradigmen und Technologien wie Webservices (Alonso, Casati, Kuno, & Machiraju, 2004) und Enterprise Service-Busse (Chappell, 2004). Diese vorwiegend technischen Entwicklungen sollten helfen, die Anzahl der technischen Schnittstellen und Schnittstellentechnologien zu verringern. Zuletzt hat sich als weiteres Softwarearchitektur-Paradigma die Microservice-Architektur entwickelt (Thones, 2015). In einer Microservice-Architektur werden nicht nur die Funktionen als Dienst exponiert, die für Integrationen notwendig sind, sondern es wird auch innerhalb der Applikation eine Servicestruktur aufgebaut, wodurch eine Vereinfachung der Aktualisierung und Skalierung von Teilen der Applikation erreicht werden soll (Namiot & Sneps-Sneppe, 2014).

Parallel zu den Entwicklungen in der Softwaretechnik entstanden auch in den organisationalen und fachlichen Teilen der Unternehmen Paradigmen zur Strukturierung von IT-Systemen mit Hilfe von Services. Ein wesentliches Thema in der Wirtschaftsinformatik (WI) sind unterschiedliche Szenarien der Integration (Aier u. a., 2009). Eine Form dieser Integration befasst sich mit der Integration von fachlichen Strukturen und den unterliegenden IT-Systemen (Business/IT-Alignment) (Aier &

Winter, 2009). Ziel dieser Ansätze ist es, die unterschiedlichen Gestaltungsziele der fachlichen Ebene wie Prozesseffizienz oder –effektivität (Winter, 2008a) mit denen der IT-Ebene wie Wiederverwendung und Performanz (Aier & Winter, 2009) in einer Integrationsarchitektur transparent darzustellen (Aier & Winter, 2009).

Als Konzept zur Integration der technischen Systeme entlang der fachlichen Strukturen wurde zunächst das Konzept der Enterprise Application Integration (EAI) diskutiert. Unternehmen strebten mit EAI eine Integration von existierenden Softwaresystemen entlang der Geschäftsprozesse an, um so medienbruchfreie und reibungslos ausführbare Geschäftsprozesse zu erreichen (Linthicum, 1999). Dieser Anspruch der Orientierung an Geschäftsprozessen konnte jedoch in der Praxis nicht erreicht werden, weshalb EAI vorwiegend als technische Integrationskonzeption gesehen wird (Aier & Schelp, 2008). SOA kann als eine Weiterentwicklung des EAI Konzeptes gesehen werden (Hassan, 2009). Serviceorientierte Architekturen sind jedoch im Gegensatz zu den EAI-Ansätzen nicht nur auf die Integration bestehender Systeme entlang der Geschäftsprozesse ausgelegt, sondern auf eine Entwicklung der Systeme nach fachlichen Kriterien (Agarwal u. a., 2012). Wesentliche Ziele bei der Gestaltung von SOA sind eine effektive Unterstützung von Geschäftsprozessen, eine höhere Agilität und eine effizientere Bereitstellung der IT-Funktionalität (Becker, Widjaja, & Buxmann, 2011).

Ein weiteres Servicekonzept, welches in Forschung und Praxis umfassend behandelt wird, ist das Cloud Computing (CC)-Paradigma. CC-Services sind als technische Services definiert, deren Nutzungsbeginn automatisiert erfolgen kann und die nutzungsabhängig skaliert werden können (Mell & Grance, 2011). Einige Autoren sehen CC-Services als grundlegende Veränderung des Bezugs von IT-Services an (Repschläger & Zarnekow, 2011), insbesondere da CC-Services über breitbandige Netzwerkverbindungen ubiquitär verfügbar sind und somit als globale IT-Infrastruktur ähnlich dem Stromnetz verwendet werden können (Gierczak, Söllner, & Leimeister, 2012; Youseff, Butrico, & Da Silva, 2008). Unternehmen erhoffen sich von der Nutzung von CC-Services eine weitere Optimierung der Kosten durch die Einbindung externer CC-Services sowie eine Steigerung der Flexibilität (Repschläger & Zarnekow, 2011).

1.4 Relevante Problemstellung

Die unterschiedlichen Möglichkeiten der Implementierung des Serviceparadigmas stellen Unternehmen vor die Entscheidung, welche Services in der spezifischen Situ-

ation des Unternehmens hilfreich sein könnten und wie diese gestaltet sein sollten. Diese Aktivitäten bedürfen einer Unterstützung durch Methoden, wie in den angeführten Beispielen aus der Praxis gezeigt wird.

So bedarf es Methoden zur initialen Gestaltung der Services innerhalb der Unternehmensarchitektur unter Beachtung der unterschiedlichen Stakeholderinteressen. Im weiteren Verlauf des Lebenszyklus der Services, während der Nutzungsphase, werden Methoden zum Management der Services erforderlich. Genau mit diesen beiden Problemstellungen befasst sich die vorliegende Dissertation.

Allerdings kann im Rahmen dieser Arbeit die aufgeworfene Fragestellung nicht für sämtliche Servicetypen und alle Bereiche der Servicegestaltung und –verbesserung bearbeitet werden. Daher muss die Fragestellung eingegrenzt werden, ohne jedoch dem Anwender die Möglichkeit zu nehmen die Resultate in den gesamten Unternehmenskontext einzuordnen. Im folgenden Abschnitt wird aus diesem Grund die Fragestellung eingegrenzt und dabei – soweit möglich – in ein bestehendes Framework eingeordnet.

2 Zielsetzung dieser Arbeit

Vor dem Hintergrund der beschriebenen Herausforderungen lässt sich die Fragestellung dieser Arbeit vorerst wie folgt formulieren:

Wie können Gestaltung und Management von Services im Unternehmen methodisch durchgeführt werden?

Wie ausgeführt ist diese Formulierung für eine Forschungsfrage zu weit angelegt und wird daher im folgenden Abschnitt so präzisiert, dass eine Beantwortung im Rahmen dieser Arbeit möglich ist.

2.1 Eingrenzung des Forschungsgegenstandes

Zur Eingrenzung der Fragestellung lassen sich drei Teilaspekte der genannten Frage konkretisieren:

Service definition: Welche Typen von Services gibt es und welche davon sollen betrachtet werden?

Servicegestaltung: Welche Aspekte dieser Services sollen gestaltet werden?

Service management: Welche Aspekte des Managements von Services gibt es und welche davon sollen betrachtet werden?

Um eine klar definierte Forschungsfrage zu generieren, wird im Folgenden das zu behandelnde Problem anhand dieser drei Fragen weiter eingegrenzt.

Für die Fokussierung der Servicedefinition in der vorliegenden Arbeit fehlt in der Literatur ein akzeptiertes Metamodell aller Servicetypen. Ein solches Modell ermöglicht eine Eingrenzung der betrachteten Servicetypen und sichert Konsistenz, Einheitlichkeit, Vollständigkeit und Vergleichbarkeit sowie auch eine Einordnung in den Kontext des Unternehmens (Österle, Winter, Höning, Kurpjuweit, & Osl, 2007). Es existieren lediglich Definitionen und Teilmodelle für einzelne Servicetypen, nicht jedoch eine ganzheitliche Darstellung (siehe Abschnitt 5.3.1). Die Beiträge dieser Dissertation setzen sich mit spezifischen Diensten auseinander, die rein fachlich definiert, aber vollständig technisch implementierbar sind. Es geht also um jene Dienste, die als Schnittstelle zwischen den technischen und den fachlich-organisationsbezogenen Teilen des Unternehmens fungieren. Zur Abgrenzung dieses spezifischen Diensttyps von anderen Diensten und zur Einbettung der Diensttypen in die Unternehmensarchitektur werden im weiteren Verlauf des Dachbeitrages unterschiedliche Servicebegriffe analysiert und daraus ein umfassendes Metamodell rekonstruiert (Abschnitt 5.3). Das entwickelte Servicemodell beinhaltet zwei fundamentale Perspektiven auf jeden Service: die Perspektive des Servicenutzers (Konsumentenperspektive) und die des Serviceanbieters (Providerperspektive). Dies erhält bei den im Fokus liegenden Servicetypen eine besondere Relevanz, da diese als Schnittstelle zwischen zwei Ebenen der Unternehmensarchitektur dienen und somit von unterschiedlichen Stakeholdern aus unterschiedlichen Perspektiven betrachtet werden (Jonkers, Lankhorst, Van Buuren, Hoppenbrouwers, & Bonsangue, 2004).

Neben der Eingrenzung des für diese Arbeit relevanten Servicetyps sollte auch eine Fokussierung auf ausgewählte Tätigkeiten der Servicegestaltung und des Servicemanagements erfolgen. Dies lässt sich anhand etablierter IT-Servicemanagement³ (ITSM) Frameworks durchführen, da diese sämtliche Aktivitäten und Methoden im Lebenszyklus von technischen Diensten strukturieren, ohne dabei jedoch auf einen bestimmten Dienstyp eingeschränkt zu sein (Hochstein, 2006).

Es gibt eine grosse Zahl von Methoden und Frameworks im Bereich IT-Servicemanagement (Mora & Raisinghani, 2014). Als de-facto Standard hat sich jedoch die IT Infrastructure Library (ITIL) durchgesetzt (Hafner, Schelp, & Winter,

³ Für eine Begriffsdefinition des ITSM Begriffes siehe auch Abschnitt 4.3

2006). Daher soll dieses Framework zur weiteren Eingrenzung genutzt werden. ITIL entstammt ursprünglich dem Bereich des IT-Infrastrukturmanagements, hat sich jedoch zu einem ganzheitlichen ITSM-Framework entwickelt (Hochstein, 2006).

ITIL definiert fünf Bereiche, strukturiert nach den Phasen der Entwicklung und Implementation von Diensten (Beims, 2012):

1. Service-Strategie: Hier werden wesentliche Aspekte der Strategie und Organisation des Serviceanbieters entwickelt.
2. Service-Design: In dieser Phase finden die konkreten Gestaltungsaktivitäten der Services statt.
3. Service-Überführung: Diese Aktivitäten befassen sich mit der Überführung von neu entwickelten oder verbesserten Services in den produktiven Betrieb.
4. Service-Betrieb: Hier sind sämtliche Aktivitäten des operativen Servicebetriebs abgebildet.
5. Kontinuierliche Verbesserung: Diese Phase befasst sich mit der kontinuierlichen Auswertung der Services und dem Einsteuern von Verbesserungen.

Basierend auf diesen Phasen soll die für diese Arbeit relevante Fragestellung auf die Phasen Servicegestaltung (Service Design) und kontinuierlichen Verbesserung (Continual Service Improvement) beschränkt werden. Dies ist insofern sinnvoll, da diese beiden Bereiche den zu gestaltenden Service behandeln, wohingegen die Strategiephase (Service Strategie) sich mit der Zielsetzung des ITSM generell und die operativen Bereiche (Service Übergang und Service Betrieb) sich mit der Art der Umsetzung des ITSM befassen (Cartlidge u. a., 2007).

Bezogen auf das Service Design definiert ITIL eine Vielzahl von Aktivitäten vom eigentlichen Service Design bis hin zum Lieferantenmanagement (Office Of Government Commerce, 2005). Innerhalb dieses Bereiches des ITIL Frameworks soll nur der Kernbereich der Servicegestaltung und innerhalb davon wiederum nur die Spezifikation der funktionalen sowie nicht-funktionalen Anforderungen des Dienstes betrachtet werden. Die Anforderungsspezifikation soll dabei aber nicht bis auf die Ebene der Detailspezifikation durchgeführt werden. Beispielsweise ist die Definition konkreter technischer Schnittstellen oder die Auswahl von Technologien zur Implementierung von Diensten nicht mehr relevant.

Das Continual Service Improvement (CSI) definiert den Prozess, der Unternehmen hilft, die Implementierung der Servicestrategie zu kontrollieren und gegebenenfalls anzupassen (Beims, 2012). Der Kernprozess des CSI ist ein siebenstufiges Vorge-

hen, welches dem Deming Cycle ähnelt (Goeken, Alter, Milicevic, & Patas, 2009). Zwei der sieben Schritte befassen sich mit der Fragestellung, was gemessen werden sollte und welche Daten messbar sind. Bezüglich der Fragestellung der Serviceverbesserung steht die Bestimmung dieser beiden Aktivitäten im Vordergrund dieser Arbeit.

2.2 Konkretisierung der Forschungsfrage

Basierend auf den beschriebenen Einschränkungen sowie der Aufteilung nach Produzenten- und Konsumentenperspektive lässt sich nun die im Vorangegangenen dargestellte initiale Fragestellung zu zwei Forschungsfragen präzisieren. Jede dieser Forschungsfragen wird jeweils durch zwei Teilfragen beschrieben. Die erste Forschungsfrage behandelt dabei den Bereich der Gestaltung von Diensten.

Forschungsfrage FF 1:

FF 1.1: Wie lassen sich Services⁴ durch den Serviceproduzenten fachlich-funktional und nicht-funktional spezifizieren?

FF 1.2: Werden andere Anforderungen aus der Perspektive der Servicekonsumenten an die Artefakte aus FF 1.1 gestellt?

Die zweite Frage behandelt den Bereich der Serviceverbesserung.

Forschungsfrage FF 2:

FF 2.1: Wie kann eine kontinuierliche Verbesserung von Services erreicht werden?

FF 2.2: Werden andere Anforderungen aus der Sicht der Servicekonsumenten an die Artefakte der FF 2.1 gestellt?

Die Ergebnisse der Untersuchung dieser Forschungsfragen finden sich in Kapitel 0 beziehungsweise Kapitel 5.2. Im folgenden Abschnitt wird der gesamte inhaltliche Aufbau der Dissertation dargestellt.

⁴ Der in den Forschungsfragen verwendete Begriff „Service“ bezieht sich auf die in Abschnitt 5.3.5 dargestellte Definition. Es handelt sich dabei nicht um eine generische, sondern auf bestimmte Typen von Services eingeschränkte Definition.

2.3 Inhaltliche Struktur der Arbeit

Da es sich bei der vorliegenden Arbeit um eine kumulative Dissertation handelt, besteht die weitere Arbeit aus vier weiteren Abschnitten des Dachbeitrages (Teil A) sowie den in Teil B aufgeführten Einzelpublikationen. Neben der bereits formulierten Motivation für die Arbeit und den daraus entstandenen Forschungsfragen erläutert der Dachbeitrag im Folgenden die verwendete Forschungsmethode (Abschnitt 2.4) sowie den Stand der Forschung (Abschnitt 3). In einem weiteren Kapitel werden relevante inhaltliche Grundlagen dargestellt (Abschnitt 4). In Abschnitt 5 wird der Forschungsbeitrag dieser Dissertation zusammengefasst. Der Abschluss des Dachbeitrages besteht aus einer Diskussion der Inhalte und einem Ausblick auf weitere Forschungsnotwendigkeiten (Abschnitt 6).

Der zweite Hauptabschnitt der Arbeit (Teil B) beinhaltet die einzelnen Veröffentlichungen. Diese werden bereits durch eine Kurzzusammenfassung in Teil A - Abschnitt 5.4 eingeführt. Abbildung 1 stellt den Aufbau der Arbeit dar, dabei sind die Forschungsbeiträge der Dissertation in grau dargestellt.

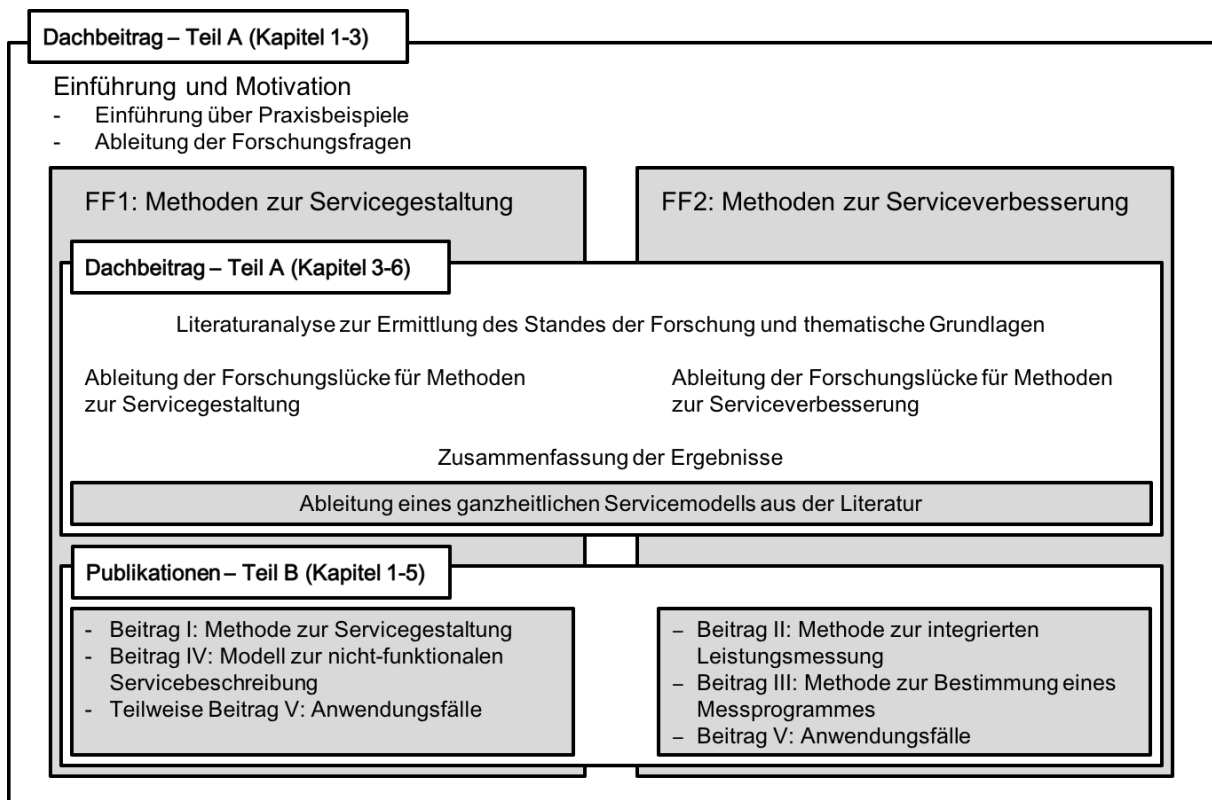


Abbildung 1: Inhaltliche Struktur der Arbeit

2.4 Forschungsmethode

Die Ergebnisse der vorliegenden Arbeit wurden mithilfe des Verfahrens der Methodenkonstruktion erarbeitet (Braun, Hafner, & Wortmann, 2004). Die Methodenkonstruktion stellt einen Teilbereich der gestaltungsorientierten WI dar und versteht sich dabei als konstruktionsorientierte Forschungsdisziplin, deren Ziel die Konstruktion nützlicher Methodenartefakte für die Gestaltung von Informationssystemen ist (Gericke & Winter, 2009).

Die gestaltungsorientierte WI liefert drei unterschiedliche Arten von Forschungsergebnissen (Hevner, March, Park, & Ram, 2004):

1. Designartefakte: Stellen Lösungen für vorher nicht gelöste Fragestellungen dar.
2. Grundlagen: Stellen bereits bewiesene Erkenntnisse der Wissensbasis dar, die erweitert werden können.
3. Methodologien: Dienen der Evaluation der Designartefakte.

Zur Typisierung der Designartefakte schlagen viele Autoren die von March & Smith (1995) vorgestellten Artefakt-Typen Konstrukt (*Construct*), Modell (*Model*), Methode (*Method*) und Instanz (*Instantiation*) vor (Gericke & Winter, 2009; March & Smith, 1995).

Während Konstrukte die Terminologie einer Problemdomäne definieren, stellt ein Modell die Beziehungen zwischen den Elementen der Terminologie her. Eine Methode hingegen stellt eine Abfolge von Aktivitäten dar, welche eine definierte Aufgabe erfüllen. Diese Methoden verwenden dabei Konstrukte und basieren auf einem definierten Modell. Eine Instanziierung bedeutet die Ausbildung einer Methode und eines Modells in Bezug auf einen konkreten Anwendungsfall.

Die Methodenkonstruktion ist der Teil der gestaltungsorientierten WI, der sich mit der Entwicklung von Methoden als Artefakt beschäftigt.

Gericke & Winter (2009) stellen einen Bezugsrahmen für die Einordnung der unterschiedlichen Artefakte des Design Science Prozesses vor (Abbildung 2 aus (Gericke & Winter, 2009)). Dieser Rahmen unterteilt zunächst die gestaltungsorientierte Forschung in die Bereiche Konstruktionsforschung (Design Science) und Artefaktkonstruktion (Design Research). Die Artefaktkonstruktion wird wiederum in die Bereiche Konstruktion und Evaluation generischer Artefakte und deren situative Anpassung unterteilt (Feld „3“ in Abbildung 2).

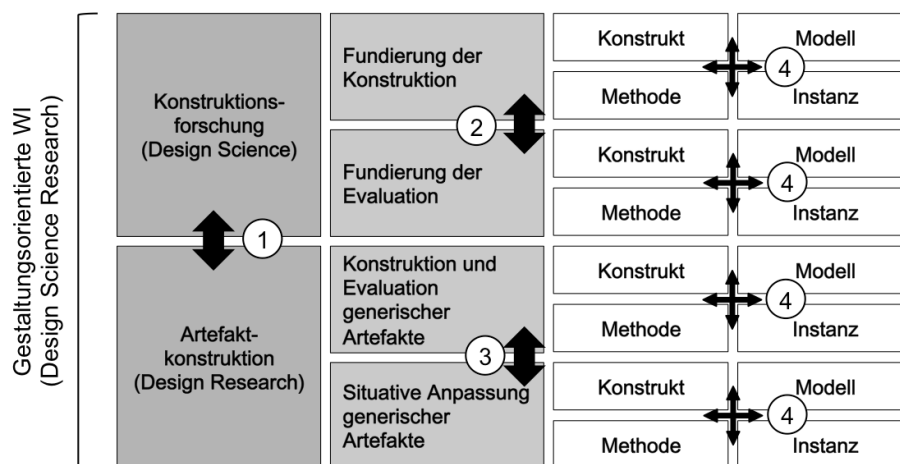


Abbildung 2: Analyse Rahmen für die konstruktionsorientierte Forschung

Basierend auf diesem Analyserahmen lässt sich die vorliegende Arbeit folgendermassen einordnen: In dieser Arbeit werden ein Modell und Methoden entwickelt (Artefaktkonstruktion), die für eine spezifische Fragestellung generisch konstruiert wurden.

3 Stand der Forschung und Forschungslücke

In diesem Abschnitt soll zunächst eine Übersicht über den Stand der Forschung in Bezug auf Servicegestaltung und -verbesserung (Kapitel 3.1 beziehungsweise 3.2) gegeben werden. Auf dieser Darstellung aufbauend, werden die im Unterkapitel 3.3 relevanten Forschungslücken entwickelt. Für die Darstellung des Forschungsstandes wurde eine strukturierte Literaturrecherche durchgeführt (Kitchenham, 2004). Eine solche Literaturrecherche muss einem strukturierten und nachvollziehbaren Verfahren folgen. Vom Brocke et al. (2009) schlagen hierzu ein Vorgehen in fünf Schritten vor, welches hier angewendet werden soll (siehe Abbildung 3).

Die Schritte I und II dienen der Definition der Zielsetzung und der Erläuterung des relevanten Suchraumes („Scope“). In Schritt I soll zunächst die Zielsetzung der Literaturrecherche definiert werden. Hierzu wird eine Taxonomie bestehend aus 6 Eigenschaften vorgeschlagen (vom Brocke u. a., 2009):

1. Fokus der Recherche: In dieser Recherche sollen Forschungsergebnisse gesucht werden, nicht etwa Forschungsmethoden oder Anwendungsfälle. Dadurch kommen als relevante Quellen ausschliesslich Publikationen in Frage, in denen Forschungsergebnisse präsentiert werden. Somit werden keine

Anwendungsberichte aus der Praxis oder vergleichbare Publikationen aufgenommen.

2. Ziel der Recherche: Das Ziel der Suche ist eine Integration der Ergebnisse, um so einen umfassenden Stand der Forschung darstellen zu können.
3. Organisation der Ergebnisse: Die Ergebnisse der Arbeit werden entlang der Forschungsfragen dargestellt.
4. Perspektive der Recherche: Das Ziel dieser Recherche ist ein neutrale Perspektive, es soll keine spezifische Position dargestellt werden. Daher findet keine Fokussierung auf bestimmte Forschergruppen oder Publikationsorgane statt.
5. Zielgruppe der Recherche: Die relevante Zielgruppe für die Literaturrecherche besteht aus Forschern der WI. Somit werden ausschliesslich Beiträge aus Forschungsorganen der WI einbezogen.
6. Abdeckung der Recherche: Die Literaturrecherche sollte möglichst umfassend sein. Aufgrund der Vielzahl der Beiträge erfolgt eine Darstellung jedoch nur in Form von ausgewählten Beiträgen sowie Gruppierungen von Beiträgen.

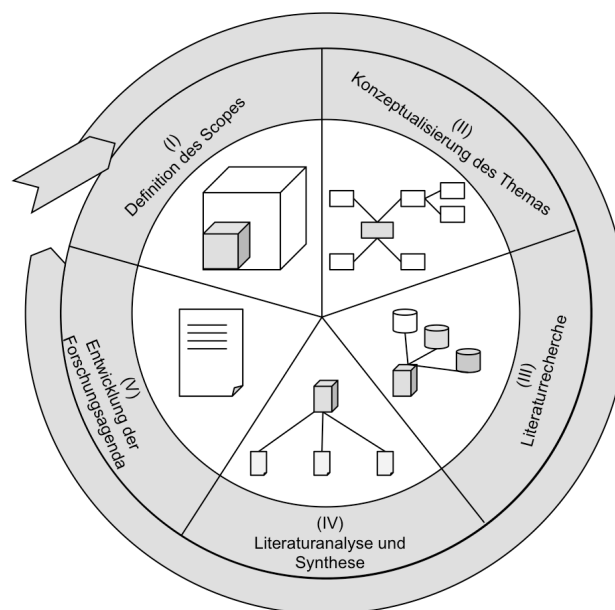


Abbildung 3: Schritte der Literaturrecherche (aus (vom Brocke u. a., 2009))

Schritt II des Vorgehens dient der Konzeptualisierung der Fragestellung sowie einer Präzisierung der Fragen, auf welche die Literaturrecherche möglicherweise Antworten liefern kann (vom Brocke u. a., 2009). Diese Inhalte wurden in den vorangegangenen Abschnitten durch die Eingrenzung des Forschungsgegenstandes (Abschnitt 2.1) und die Darstellung der existierenden Begriffe (Abschnitt 1.3) adressiert.

Für die Suche der relevanten Literatur (Phase III) werden verschiedene Suchtechniken vorgeschlagen (vom Brocke u. a., 2009). Initiale Suchtechnik ist die Recherche in einer definierten Menge an Publikationsorganen anhand von Suchbegriffen. Diese wurde mit den in aufgeführten Begriffen durchgeführt. Ausgehend von der so aufgefundenen Menge an Beiträgen kann rückwärts oder vorwärts gesucht werden. Eine rückwärtige Suche bedeutet, dass in einem Artikel zitierte Referenzen ausgewählt und analysiert werden; eine vorwärts gerichtete Suche sucht hingegen nach Arbeiten, die den zunächst gefundenen Artikel zitieren. Dies ermöglicht die Identifikation älterer Beiträge (rückwärts) oder neuerer Beiträge (vorwärts) sowie die Ausweitung auf weitere Publikationsorgane anhand relevanter Beiträge (vom Brocke u. a., 2009).

Bei der Auswahl der relevanten Publikationsorgane wurde auf das VHB Jourqual Ranking (Verband der Hochschullehrer für Betriebswirtschaft) zurückgegriffen (VHB, 2015). Aus diesem Ranking wurden alle Organe gewählt, die ein A+, A oder B-Ranking haben und Themen der Wirtschaftsinformatik behandeln. Neben der Empfehlung nach Möglichkeit ausschliesslich in höherwertigen Organen zu suchen (vom Brocke u. a., 2009), ist diese Auswahl auch durch den Fokus der Recherche begründet. Da in der Literaturrecherche ausschliesslich Forschungsergebnisse der WI gesucht werden und keine Publikationen aus der Praxis wie beispielsweise Anwendungsberichte (siehe „Fokus der Recherche“ zu Schritt I), macht nur eine Suche in Forschungspublikationen Sinn. Weiterhin wird das Thema schon länger in der Forschung diskutiert, weshalb davon auszugehen ist, dass relevante Beiträge mittlerweile in den Organen mit A und B-Rankings erscheinen, obwohl in diesen der Zeitraum bis zur Publikation gewöhnlich deutlich länger ist. Abschliessend lässt sich festhalten, dass mit der fokussierten Suche auf die A- und B-Organen bereits eine grosse Menge an Publikationen aufgefunden werden konnte und eine weitere Ausweitung an der Quantität gescheitert wäre.

Zur Erreichung eines möglichst umfassenden Trefferspektrums wurden direkte und indirekte Suchbegriffe definiert und mit deren Hilfe die Metadaten (Titel, Kurzzusammenfassungen und Schlagworte) durchsucht. Als Suchzeitraum wurden die Jahre 2000 – 2015 gewählt, da die frühen Beiträge zu den relevanten Serviceparadigmen in den Jahren kurz nach 2000 erschienen sind. Stichprobenartige Suche in früheren Zeiträumen brachte erwartungsgemäß keine weiteren Ergebnisse. Die gewählten Suchbegriffe und die Anzahl der gefundenen Beiträge sind in Tabelle 1 für beide Forschungsfragen aufgeführt. Sämtliche Suchbegriffe wurden in deutscher und englischer Schreibweise verwendet.

Als direkte Suchbegriffe wurden Kombinationen aus dem Begriff „Methode“ und einem weiteren Begriff gewählt. Ziel der Recherche war es, relevante Beiträge zur Methodenforschung für die Gestaltung und Verbesserung von Services zu finden. Daher wurde zunächst nach sämtlichen Beiträgen gesucht, welche die Begriffe „Service“ und „Methode“ enthielten. Diese Suche erwies sich als zu unspezifisch (siehe Tabelle 1), so dass für die einzelnen Forschungsfragen weitere Begriffe zur Fokussierung auf die Fragestellungen der Gestaltung und Verbesserung von Diensten eingeführt wurden. Sämtliche Suchbegriffe wurden auch variiert angewendet sowie in Deutsch und Englisch genutzt (Tabelle 1). Zusätzlich zu den direkten Suchbegriffen wurden auch indirekte Suchbegriffe eingesetzt. Indirekt sind diese Begriffe insofern, als nicht direkt nach einem Methodenbeitrag gesucht wurde, sondern stattdessen die oben genannten Service Paradigmen (SOA, CC, ITSM) eingesetzt wurden.

Als weitere indirekte Suchbegriffe wurden „Enterprise Service“ und „Software Service“ (mit der Variation „Microservice“) herangezogen, die jedoch keine nennenswerte Anzahl an Treffern hervorbrachten.

Im Zuge der Analyse ließen sich ausser Häufungen von Beiträgen zu einigen Themen keine regelrechten Diskussionsthreads beziehungsweise sehr häufig zitierte Artikel nachweisen.

Tabelle 1: Suchbegriffe für die Literaturrecherche

Suchbegriff	Variation / Übersetzung	Anzahl
Service UND Methode	„service“, „method“	2442
Forschungsfrage 1 - Servicegestaltung		
Direkte Suchbegriffe		
Servicegestaltung und Methode	„Service Identifikation“, „Dienstgestaltung“, „Dienst Identifikation“, „service identification“, „service design“, „method**“	176
Indirekte Suchbegriffe		
Serviceorientierte Architektur UND Gestaltung	„service oriente architecture“, service-oriented architecture“, „SOA“, „design“, „identification“	20
Cloud UND Gestaltung	„design“, „*aaS“	13
ITSM UND Gestaltung	„IT Service Management“, „ITSM“, „design“, „identification“	12
Forschungsfrage 2 – Serviceverbesserung		
Direkte Suchbegriffe		
Serviceverbesserung UND Methode	„Dienst Verbesserung“, „service improvement“, „method“	41
Indirekte Suchbegriffe		
Serviceorientierte Architektur UND Verbesserung	„service oriented architecture“, „service-oriented architecture“, SOA, „improvement“	2
ITSM UND Verbesserung	„IT Service Management“, „ITSM“, „improvement“	1

Tabelle 1: Suchbegriffe für die Literaturrecherche

Suchbegriff	Variation / Übersetzung	Anzahl
Cloud UND Verbesserung	„improvement“, „*aaS“	3

In Schritt IV des Vorgehens wurden die Beiträge analysiert und zusammengefasst (vom Brocke u. a., 2009). Dabei wurden die aufgelisteten Suchergebnisse zunächst anhand der Inhalte des Titels und der Kurzzusammenfassung auf Relevanz untersucht und das Trefferspektrum in vier Schritten weiter entwickelt, um eine möglichst hohe Relevanz der Beiträge zu erreichen.

1. Eingrenzung über den semantischen Zusammenhang: Zunächst wurden alle gefundenen Artikel dahingehend untersucht, ob die beiden Suchbegriffe in einem inhaltlichen Verhältnis zueinander stehen. Nur wenn sich beispielsweise der Begriff „Management“ inhaltlich auf den Begriff „Service“ bezieht, wurde der Artikel in die weiter zu betrachtende Menge an Artikeln aufgenommen.
2. Eingrenzung anhand der Relevanzkriterien K1 und K2: Da im Rahmen dieser Arbeit Methoden- und Modellartefakte entwickelt werden, die auf einem nicht rein technischen Servicebegriff basieren, müssen relevante Beiträge in Bezug auf die Forschungsfragen den folgenden Kriterien genügen:
 - a. K1: Liefert der Beitrag ein Artefakt im Sinne der gestaltungsorientierten Forschung (Methode, Technik, etc. nach (Braun u. a., 2004))
 - b. K2: Liegt dem Beitrag ein nicht rein technisches Dienstverständnis zu Grunde
3. Vorwärts gerichtete Suche: Mit der so reduzierten Anzahl an Beiträgen wurde im nächsten Schritt eine vorwärts gerichtete Suche durchgeführt. Bei dieser Art der Suche werden Beiträge identifiziert, die einen der zuerst aufgefundenen Beiträge zitieren. Dadurch wird der Suchraum auf Beiträge außerhalb des VHB Jourqual Spektrums (VHB, 2015) oder auch der konkreten Suchbegriffe erweitert.
4. Erneute Eingrenzung durch die Kriterien K1 und K2: Auf diese so erweiterte Menge an identifizierten Beiträgen werden erneut die Kriterien K1 und K2 angewandt.

Dieses Vorgehen wurde für die beiden Forschungsfragen jeweils einzeln durchlaufen (siehe Abbildung 4 und Abbildung 5). Mithilfe dieser Eingrenzung wurde eine Menge relevanter Beiträge identifiziert, deren Inhalt komplett analysiert wurde. Insbesondere die semantische Eingrenzung durch die Kriterien K1 und K2 war mithilfe einer reinen Stichwortsuche nicht möglich, weshalb auch die vorwärts gerichtete Suche anhand der relevanten Beiträge eine sinnvolle Strategie erschien, um den Suchraum bezüglich der Publikationsorgane zu erweitern.

In den folgenden beiden Abschnitten werden für jede Forschungsfrage die Forschungsergebnisse der gefundenen Beiträge zusammengefasst dargestellt. Aufgrund der immer noch erheblichen Anzahl der gefundenen Beiträge werden, wie in Schritt I.6 dargestellt, nur ausgewählte Beiträge als Beispiel für eine Gruppe ähnlicher Arbeiten genannt. Eine komplette Liste der referenzierten Beiträge befindet sich im Appendix B.

3.1 Servicegestaltung

Mit der Servicegestaltung befasst sich eine große Anzahl an Arbeiten. Auch nach dem ersten Schritt der Eingrenzung sind noch 170 Beiträge zu analysieren (siehe Abbildung 4). Die Einschränkung der Suchergebnisse durch die Kriterien K1 und K2 ist jedoch zielführend. Einige Beiträge (6) generieren kein Artefakt, sondern stellen den Stand der Forschung oder einzelne Fallstudien dar (Becker, Widjaja, & Buxmann, 2011; Stephen & MacDonell, 2011). Ein großer Anteil der aufgefundenen Beiträge behandelt rein technische Themen (122), darunter sind algorithmische Fragestellungen der Softwareentwicklung (Andreas, Krivelevich, & Goerd, 2005; Morfonios, Konakas, Ioannidis, & Kotsis, 2007) sowie teilweise auch Arbeiten, die sich mit technischen Fragen der Adaption von bestehender Software befassen (Baghdadi, 2006). Neben diesen sehr technikfokussierten Beiträgen gibt es eine Gruppe an Beiträgen (12), die eine rein betriebswirtschaftliche oder organisatorische Perspektive einnehmen (Fleischmann, Schmidt, Stary, & Augl, 2013; Oshri, Kotlarsky, & Willcocks, 2007). 17 weitere Beiträge entfallen durch die Filterung des Kriteriums K2 aus dem weiter zu betrachtenden Spektrum, da sie sich mit sehr spezifischen Teilthemen der Gestaltung, wie beispielsweise der Gestaltung einer High-Performance Computing Plattform (Beck, Tönker, & Dernbecher, 2013) oder Sicherheitsfragestellungen befassen (Loske, Widjaja, & Buxmann, 2013; Trujillo-Rasua & Domingo-Ferrer, 2013).

Die vorwärts gerichtete Suche anhand der gefundenen Beiträge führt zu einer erneuten umfangreichen Erweiterung der Suchergebnisse (96 zusätzliche Referenzen). Die große Menge an Beiträgen entstand, da bei diesem Schritt komplette Literaturrecherchen über Beiträge zur Servicegestaltung im SOA Kontext gefunden wurden (Gu & Lago, 2010; Huergo u. a., 2014; Kohlborn, Korthaus, Chan, & Rosemann, 2009; Mora & Raisinghani, 2014; Ramollari, Drandis, & Simons, 2007). Diese Übersichtsarbeiten beinhalten ein breites Spektrum an Arbeiten zur Servicegestaltung, die fast alle den Kriterien des Suchvorgehens bis einschliesslich K1 entsprechen (6 Arbeiten stellen kein Methodenartefakt vor). Durch Anwendung von K2 kann die Anzahl relevanter Arbeiten jedoch erheblich reduziert werden, da viele der gefundenen Methoden einen ausschliesslich technisch definierten Servicetyp adressieren.

Ein Grossteil der im Schritt vier gefundenen Arbeiten (40) befasst sich mit der Gestaltung rein technischer Services. Diese Klassifikation erfolgte durch unterschiedliche Kriterien wie beispielsweise der expliziten Benennung von Technologien im Titel der Beiträge (beispielsweise der Bezug auf Java-Technologien in (S. Li & Tahvildari, 2008)), oder eindeutig der Softwareentwicklung zugeordnete Methoden (beispielsweise der Bezug auf „Software Product Line Development“ in (Lee, Muthig, & Naab, 2008)).

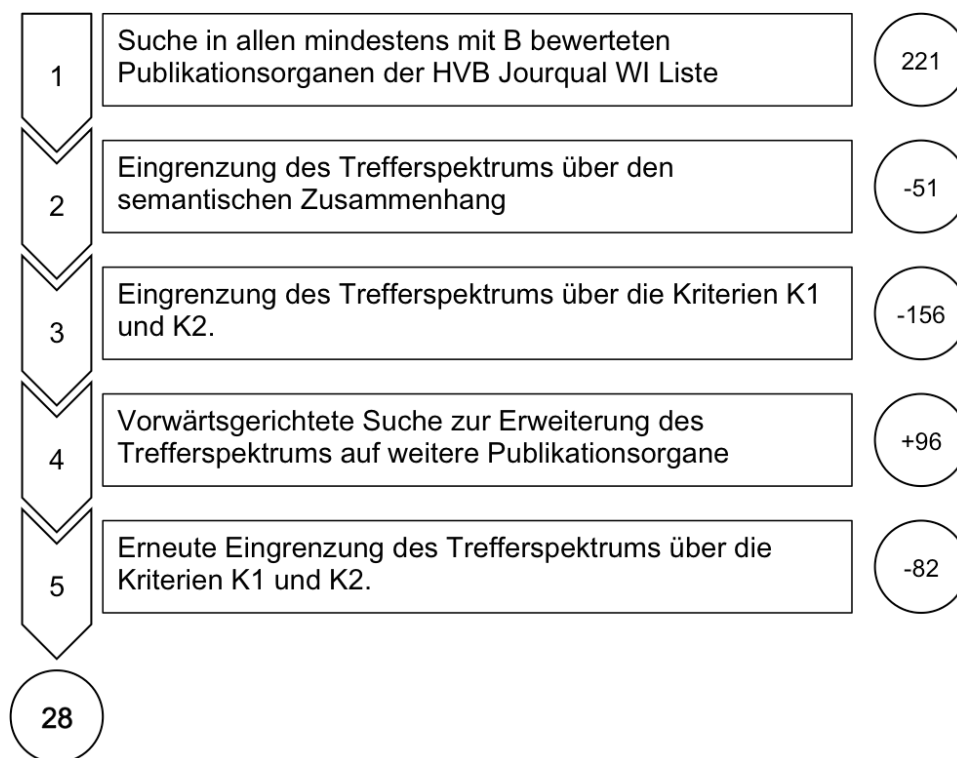


Abbildung 4: Eingrenzung des Trefferspektrums der Beiträge zur Servicegestaltung

Eine weitere Gruppe (14) von Beiträgen behandelt die Fragestellung der Kapselung von bestehenden Systemen und deren Bereitstellung als Dienst. Weiterhin gibt es einige Beiträge (7), die spezifische Branchen adressieren (W. Li, Liu, Yang, & Yu, 2014). Eine kleinere Gruppe der Arbeiten aus der vorwärts gerichteten Suche (6) war teilweise nicht nachvollziehbar publiziert oder aber nicht auffindbar.

Dieses Vorgehen führt zu einer Ergebnismenge von insgesamt 28 Arbeiten, die detailliert analysiert wurden. Auffällig ist, dass der Großteil der Forschungsbeiträge bis zum Jahr 2010 publiziert wurden, was auch in anderen Arbeiten bestätigt wurde (Huergo u. a., 2014). Die erste Gruppe besteht aus 7 Arbeiten, welche eine formal definierte Vorgehensweise wie beispielsweise Algorithmen beinhalten (Azevedo, Santoro, & Baião, 2009; Bianchini & Capiello, 2009; Caetano, Silva, & Tribolet, 2010; Chaari, Biennier, Favrel, & Benamar, 2007; Dwivedi & Kulkarni, 2008; Menascé, Casalicchio, & Dubey, 2008; Z. Wang, Xu, & Zhan, 2005).

Eine weitere Gruppe von Arbeiten definiert das Vorgehen nicht formal, gibt aber detaillierte Handlungsanweisungen vor. Die in diesen Beiträgen präsentierten Methoden gehen häufig top-down vor, wobei die Services aus den Prozessmodellen entwickelt werden (7 Beiträge) (Bianchini, Capiello, De Antonellis, & Pernici, 2014; Braunwarth & Friedl, 2010; Fareghzadeh, 2008; Klose, Knackstedt, & Beverungen, 2007; Kohlmann & Alt, 2007; Souza u. a., 2011; Stewart & Chakraborty, 2010). Zwei Beiträge dieser Gruppe ergänzen das Vorgehen durch eine bottom-up-Analyse, die auch technologische Gegebenheiten in die Dienstgestaltung einbezieht (Erradi, Anand, & Kulkarni, 2006; Shirazi, Fareghzadeh, & Seyyedi, 2009).

Fünf weitere Arbeiten leiten die Services ebenfalls top-down ab, gehen dabei jedoch nicht von dem Prozessmodell aus, sondern starten von einem Zielmodell (Andersson, Johannesson, & Zdravkovic, 2009; S. Kim, Kim, & Park, 2008; Weigand & Johannesson, 2009) beziehungsweise von einer Unternehmensontologie (Alizadeh & Mohsenzadeh, 2011; Yousef, Odeh, Coward, & Sharieh, 2009).

Zwei Autoren befassen sich detailliert mit einer Methode zur Ableitung unterschiedlicher Servicegranularitäten (Aier, 2006; Heinrich & Zimmermann, 2012).

Die Arbeiten einer letzten Gruppe (Arsanjani, Ghosh, & Allam, 2008; Beverungen, Knackstedt, & Müller, 2008; Birkmeier, Gehlert, Overhage, & Schlauderer, 2013; Offermann, 2008b; Winkler & Buhl, 2007) stellen jeweils eine umfassende Methode vor, bei welcher die Gestaltung der Services bis zur Implementierung erfasst ist. Darunter befindet sich eine Publikation mit der Weiterentwicklung der in Abschnitt B -

Kapitel 1 vorgestellten Arbeit (Offermann, 2008a). Weiterhin ist eine kommerzielle Methode nicht detailliert in der Veröffentlichung ausgeführt (Arsanjani u. a., 2008).

3.2 Serviceverbesserung

Im Bereich des Servicemanagements ergibt die Suche in den gelisteten Publikationsorganen im Vergleich zu der Servicegestaltung eine erheblich geringere Anzahl an Treffern (47). Abbildung 5 zeigt den weiteren Verlauf der Eingrenzung auf die relevanten Beiträge an.

Zunächst wurden 12 Arbeiten identifiziert, bei denen die Begriffe „Verbesserung“ und „Service“ nicht in einem inhaltlichen Zusammenhang stehen. Hierbei handelt es sich um Übersichten zum Stand der Forschung, in denen mehrere Themen behandelt werden (Alter, 2013; Wohlin, 2001): Entweder um Beschreibungen von spezifischen Implementierungen (Scannapieco, Virgillito, Marchetti, Mecella, & Baldoni, 2004) oder um technische Beiträge, in denen beide Begriffe vorkommen (Prakash, Deelman, & Bagrodia, 2000; Zhang, Leyuan, Meyer, Nazareth, & D’Souzaa, 2009).

In der zweiten Phase der Analyse der gefundenen Beiträge wurden diese auf die genannten Kriterien K1 und K2 hin untersucht, wobei 31 Beiträge als irrelevant identifiziert wurden.

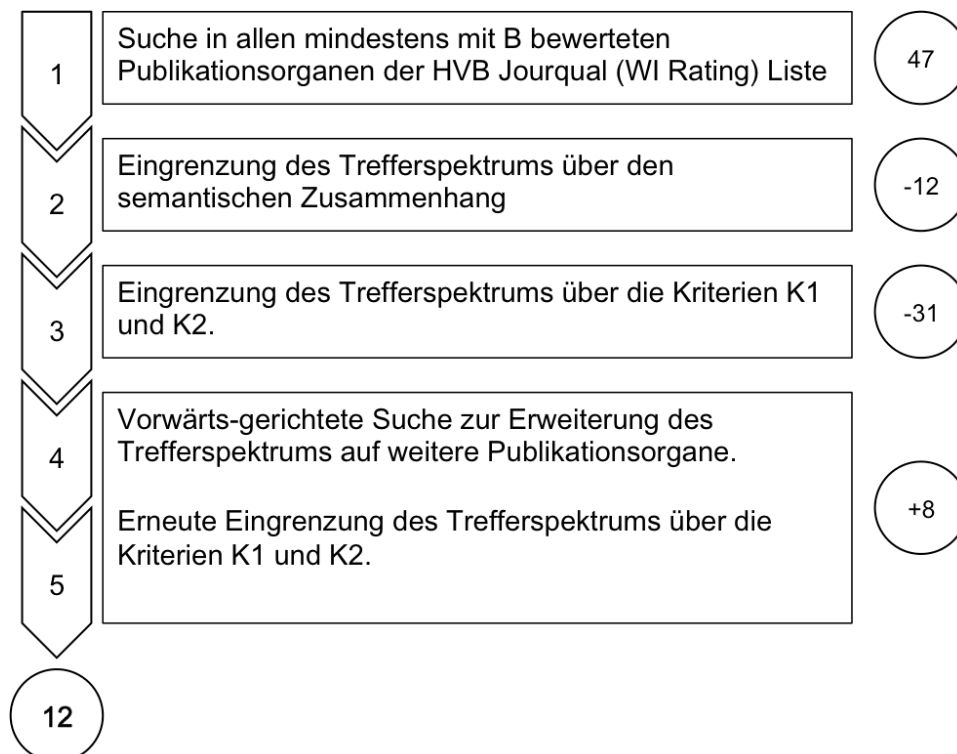


Abbildung 5: Eingrenzung des Trefferspektrums von Beiträgen zur Serviceverbesserung

Durch das Kriterium K1 werden 7 Beiträge ausgeschlossen, da diese keine Artefakte als Forschungsbeitrag liefern, sondern beispielsweise bestehende Methoden evaluieren (Barbara, Philippsen, Tichy, & Unger-Lamprecht, 2002; Córdoba, 2009; Erik, Briand, & Adsholm, 2008).

Die Anwendung des Kriteriums K2 führt zur Reduktion der Trefferliste um 31 weitere Beiträge. Der Großteil dieser Beiträge (20) befasst sich mit technischen Services und deren Verbesserung (beispielsweise (Ashish & Saxena, 2002; Sunita, Li, & Rudraraju, 2004)). Vier weitere Beiträge hingegen behandeln eine Verbesserung der Methoden zur Softwareentwicklung (Kalle & Lyytinen, 2000; Napier, Mathiassen, & Johnson, 2009; Unterkalmsteiner u. a., 2012; Zickert Roman, 2012).

Die vorwärts gerichtete Suche sowie eine Erweiterung des Suchraumes auf Publikationsorgane mit stärkerem Anwendungsbezug führen zu einer erheblich höheren Anzahl an Treffern. Von diesen Beiträgen können 8 als relevant für die Forschungsfrage 2 gesehen werden.

Von den so insgesamt 12 Beiträgen mit hoher Relevanz beziehen sich einige auf eine Kombination aus IT Service Management und übergreifendem Architekturmanagement (Hafner u. a., 2006; Radhakrishnan, 2008; Zarnekow, Hochstein, & Brenner, 2005). Sie befassen sich – mit jeweils unterschiedlichen Schwerpunkten – mit der Fragestellung, inwiefern das Serviceparadigma auf unterschiedlichen Ebenen mithilfe des ITIL-Frameworks in der (IT-) Organisation implementiert werden kann und entwickeln Erweiterungen oder Konkretisierungen des ITIL-Frameworks, jedoch keine konkreten Methoden für die Verbesserung der Services.

Eine weitere Gruppe von Beiträgen befasst sich mit konkreten Methoden für spezifische Einzelfragestellungen im Servicemanagement. Hierbei wird insbesondere die Notwendigkeit des interdisziplinären Vorgehens betont (Bardhan, Demirkan, Kannan, Kauffman, & Sougstad, 2014), die Einbeziehung von nicht-funktionalen Aspekten herausgestellt (Heinrich, Klier, Lewerenz, & Zimmermann, 2015) oder auch die Verknüpfung der Gesamtleistung des Unternehmens mit der Leistung einzelner Dienste und Organisationseinheiten gefordert (G. Kim, Shin, Kim, & Lee, 2011). Zwei Beiträge befassen sich mit der finanziellen Perspektive der Optimierung der Servicegestaltung beziehungsweise mit einer Methode zum Treffen von Investitionsentscheidungen (Braunwarth & Friedl, 2010; Irani, Love, Elliman, Jones, & Themistocleous, 2005). Ein weiterer Beitrag befasst sich mit der Fragestellung der Verfügbarkeit von Services (Franke, Johnson, & König, 2014).

Zwei Beiträge stellen eine Methode vor, die für die Zielsetzung dieser Arbeit sehr relevant ist (Chen, Wu, & Wu, 2006; Vasconcelos, Sousa, & Tribolet, 2007). Zuletzt existiert eine weitere Arbeit, deren Fokus auf der Auswertung der formalen Qualitätskriterien liegt (Lima, de Sousa, Oliveira, Sauve, & Moura, 2010).

3.3 Forschungslücke

In dieser Dissertation werden Methodenartefakte zur Servicegestaltung und –verbesserung entwickelt. Dabei ist es notwendig, von einem fokussierten Servicebegriff auszugehen, da die Beantwortung der Forschungsfragen im Hinblick auf die unterschiedlichen existierenden Servicebegriffe im Rahmen dieser Arbeit nicht möglich ist. Auch die gefundenen Methodenbeiträge beziehen sich nahezu ausschliesslich auf einen nur implizit definierten Servicetyp (wie zum Beispiel SOA-Service, CC-Service). Zur Einordnung der gefundenen Beiträge sowie der Fokussierung der eigenen Methodenbeiträge ist es daher notwendig, zunächst ein umfassendes und detailliertes Modell der unterschiedlichen Servicebegriffe zu entwickeln, um dann eine nachvollziehbare Fokussierung vorzunehmen. In der gesichteten Literatur finden sich einige explizite und detaillierte Definitionen (vergleiche Abschnitt 5.3.1), jedoch wird in keiner dieser Arbeiten ein gesamthafte Modell der unterschiedlichen Servicetypen entwickelt. Teil A der vorliegenden Arbeit soll dies in Form einer Rekonstruktion eines umfassenden Metamodells aus den in der Literatur aufgeführten Servicebegriffen leisten (siehe Kapitel 5.3). Auf diesem Modell aufbauend schliessen die Teilveröffentlichungen dieser Dissertation (Teil B) einzelne Methodenlücken (Beiträge III, IV) und verbessern bestehende Methoden (Beiträge I, II) zu den Themen Servicedesign und Serviceverbesserung. In Tabelle 2 wird der Forschungsstand auf Basis der Literaturanalyse zusammengefasst und um die in dieser Arbeit behandelten Aspekte erweitert.

Die bei der Analyse des Forschungsstandes gefundenen Methoden zur Gestaltung und Verbesserung von Services adressieren nicht direkt die Nutzerperspektive von Services. Im Rahmen dieser Arbeit soll daher untersucht werden, inwiefern die Nutzerperspektive zu berücksichtigen ist, was in den FF 1.2 und FF 2.2 reflektiert ist.

Neben diesem grundlegenden Defizit sind in Bezug auf die Gestaltung und Verbesserung von Services in der Analyse der existierenden Literatur einige weitere relevante Forschungslücken offenbar geworden.

Tabelle 2: Zusammenfassung des Forschungsstandes

	Servicedesign	Servicemanagement
Gefundene Beiträge	Verschiedene Methoden der Servicegestaltung, ausschliesslich aus der intra-organisationalen Produzentensicht und mit einem Fokus auf die funktionalen Aspekte der Servicegestaltung.	Wenige Arbeiten, die sich mit der Verbesserung von Services befassen. Diese binden keine existierenden Methoden ein (z.B. Adaption) und fokussieren auf die technisch messbaren Aspekte der Servicequalität.
Forschungsbeitrag dieser Arbeit	Entwicklung eines Methodenfragments für Servicedesign aus der Produzentensicht, welche geeignet ist unternehmensübergreifend und aus der Nutzerperspektive Services zu gestalten. Entwicklung einer Technik und eines Modells zur Modellierung nicht-funktionaler Eigenschaften von Services.	Methodenbeiträge zu den Themen: <ul style="list-style-type: none"> - Bewertung der Leistung von Service im Kontext der Unternehmensziele - Erfassung von technisch messbaren und qualitativ erfassbaren Zielen von Services

Im Bereich der Servicegestaltung existieren viele unterschiedliche Methoden, die meist einen rein intra-organisationalen und funktionalen Fokus aufweisen. Die Analyse sämtlicher Gestaltungsverfahren für Dienste zeigt drei relevante Bereiche für die weitere Forschung auf (Gu & Lago, 2010; Huergo u. a., 2014):

- Dienstgestaltung wird entweder aus rein fachlicher Sicht begriffen oder die Dienste werden aus der bestehenden IT-Struktur heraus geschnitten. Daher besteht ein Bedarf an Methoden, die sowohl fachlich relevante Dienste definieren als auch in der IT-Struktur möglichst einfach zu verankern sind.
- Dienste, die sowohl interne als auch externe Dienstnutzer haben, müssen mit unterschiedlichen Verfahren gestaltet werden.
- Die nicht-funktionalen Aspekte von Diensten müssen bei der Gestaltung einbezogen werden.

Diese Anforderungen an die Forschung werden teilweise durch die in dieser Arbeit aufgezeigten Methoden behandelt. Während Beitrag I geeignet ist, interne und externe Dienstanutzer abzubilden, beschreibt Beitrag II eine Technik zur Modellierung der nicht-funktionalen Aspekte (siehe auch Abschnitt 0).

Neben der Fragestellung der Servicegestaltung ist der Bereich der Verbesserung von Services ein weiterer Aspekt dieser Arbeit. Für diesen Bereich zeigt – wie bereits erwähnt – die Literaturrecherche wenige relevante Forschungsbeiträge auf und auch in der Praxis sind diese Prozesse noch nicht verankert (Bardhan u. a., 2014; Cartlidge u. a., 2007). Insbesondere fehlt eine ganzheitliche Methode, die eine Einbindung der Aktivitäten der Serviceverbesserung in eine unternehmensübergreifende Methode ermöglicht. Die existierenden Arbeiten erfordern ein eigenständiges Modell des gesamten Unternehmens zur Analyse der Verbesserungspotenziale (Chen u. a., 2006; Vasconcelos u. a., 2007), was die Möglichkeiten der Umsetzung erheblich verringert. Daher besteht der Bedarf, die Methoden zur Verbesserung von Services in bestehende und in der Praxis implementierte Konzepte einzubinden. Dies leisten die in Beitrag III und IV vorgestellten Artefakte. Weiterhin fällt der Fokus auf technisch messbare Aspekte der Serviceverbesserung in der bestehenden Literatur auf (Lima u. a., 2010). Da die Servicequalität durch technische objektiv messbare Aspekte (z.B. die Verfügbarkeit), aber auch durch die vom Nutzer subjektiv wahrgenommenen Eigenschaften bestimmt werden, besteht in diesem Bereich zusätzlicher Forschungsbedarf.

Diese Aspekte der Serviceverbesserung wurden insbesondere in den Beiträgen III und IV adressiert, wobei diese Arbeiten existierende Methoden auswählen und an den spezifischen Servicetyp adaptieren (siehe Abschnitt 5.2).

4 Inhaltliche Grundlagen

In diesem Abschnitt sollen einige inhaltliche Grundlagen dargestellt werden. Neben den Basisdefinitionen der bereits erwähnten Konzepte SOA, CC und IT-Service-Management (ITSM) soll insbesondere das Business Engineering Business Metamodell eingeführt werden, welches im Verlauf der Arbeit als inhaltlicher Bezugsrahmen für die Einordnung des im Dachbeitrag entwickelten Servicemodells dient.

4.1 Basisdefinition Service-orientierte Architekturen

Der Begriff SOA beschreibt einen vielfach genannten Trend im Informationssystem-Management. SOA ist ein Architekturmuster, bei dem einzelne und in sich geschlossene Dienste miteinander verbunden werden und somit auch komplexere Anforderungen umsetzen können.

Eine solche sehr generische Basisdefinition liefert die OASIS Referenz Dokumentation (MacKenzie, Laskey, McCabe, Brown, & Metz, 2006, S. 8):

Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.

Unternehmensarchitekturen, die auf SOA-Prinzipien entwickelt wurden, sollen es den Unternehmen ermöglichen, die Geschäftsanforderungen einfacher in der Unternehmens-IT umzusetzen (Aier & Schönherr, 2006; Schelp & Winter, 2007) und durch Wiederverwendung und Orchestration eine höhere Flexibilität ermöglichen.

SOA wurde zunächst insbesondere auf Softwarearchitekturen bezogen. Viele Autoren früherer SOA-Publikationen setzen den Begriff nahezu gleich mit einer Webservice-orientierten Software-Architektur (z.B. (McCoy & Natis, 2003; Newcomer & Lomow, 2004).

In der weiteren Entwicklung wurde der Begriff jedoch auch auf andere Ebenen der Unternehmensarchitektur bezogen und somit das Ziel einer durchgängig serviceorientierten Unternehmensarchitektur definiert (Schelp & Winter, 2008).

Der Servicebegriff in den Beiträgen zur SOA wird in Abschnitt 5.3.2 detailliert analysiert und dargestellt.

4.2 Basisdefinition Cloud Computing

Es gibt eine Reihe unterschiedlicher Begriffsdefinitionen für den Begriff Cloud Computing, die sich jedoch in wesentlichen Teilen überschneiden (Armbrust u. a., 2009; Mell & Grance, 2011; Youseff u. a., 2008). Die am häufigsten zitierte Definition ist die des National Institute of Standards and Technology (NIST) (Mell & Grance, 2011, S. 2):

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model

is composed of five essential characteristics, three service models, and four deployment models.

Die referenzierten fünf zentralen Charakteristika sowie drei Servicemodelle und vier Deploymentmodelle werden bei der detaillierten Analyse des Servicebegriffes in Abschnitt 5.3.4 ausgeführt.

4.3 Basisdefinition IT-Servicemanagement

Als ITSM werden Massnahmen und Methoden bezeichnet, die auf eine Unterstützung eines Unternehmens durch IT-Services abzielen (Mora & Raisinghani, 2014).

Cartlidge u.a. (2007) definieren den Begriff in Anlehnung an das ITIL Framework folgendermassen (Cartlidge u. a., 2007, S. 6):

Service Management is a set of specialized organizational capabilities for providing value to customers in the form of services.

Viele Unternehmen implementieren ein ITSM, da sie sich eine höhere Effektivität und Effizienz des IT Managements erhoffen und somit auch eine insgesamt erhöhte Leistungsfähigkeit der Organisation (Mora & Raisinghani, 2014). Insbesondere da IT-Organisationen nicht mehr als reine Technologielieferanten gelten, sondern als Dienstleister für die Fachorganisation auftreten, ist die Bedeutung des ITSM enorm (Braun & Winter, 2007; Sallé, 2004).

Zur Umsetzung des ITSM wurde eine Vielzahl von Konzepten entwickelt (Mora & Raisinghani, 2014), die einem Unternehmen helfen sollen, Services zu planen, implementieren, betreiben, überwachen, warten und zu verbessern (ISO, 2005). Viele dieser Konzepte sind in der Praxis entstanden und basieren auf dort bewährten Vorgehensweisen (Cartlidge u. a., 2007). Insbesondere die IT Infrastructure Library (ITIL) gilt heutzutage als de-facto-Standard (Hochstein, 2006).

Bei der Untersuchung der Servicekonzepte des ITSM (Abschnitt 5.3.3) werden aber auch die weiteren Ansätze Control Objectives for Information and Related Technology (Cobit), IBM Tivoli Unified Process (ITUP), Microsoft Operations Framework (MOF) sowie Capability Maturity Model Integration for Services (CMMI-SVC) betrachtet.

4.4 Business Engineering Business Metamodell

Business Engineering (BE) ist ein Ansatz, Unternehmen bei der Transformation durch ein strukturiertes, ingenieurmässiges Vorgehen zu unterstützen und somit Plan- und Reproduzierbarkeit zu erreichen (Österle & Blessing, 2005).

Der St. Galler Ansatz des Business Engineering hat verschiedene Artefakte entwickelt, welche als Grundlage für die vorliegende Arbeit dienen. Zum einen existiert ein einheitliches Business-Metamodell, welches als „[...] konzeptionelles Datenmodell“ für die Beschreibung der Gestaltungsobjekte dient (Österle u. a., 2007, S. 191). Dieses Metamodell wird in einem konkreten Anwendungsfall als Instanz modelliert und gibt dem Business Engineer einen Ordnungsrahmen für die Modellierung.

Zusätzlich existieren zu dem Metamodell definierte Erweiterungsmechanismen, mit deren Hilfe das Kernmodell um spezifische Aspekte ergänzt werden kann. Diese Mechanismen stellen sicher, dass potenzielle Erweiterungen sowohl syntaktisch als auch semantisch konsistent in das Modell eingepasst werden (Österle u. a., 2007). Diese Mechanismen werden im Folgenden verwendet, um das BE Business Metamodell um den Aspekt der SOA zu erweitern.

Weiterhin bildet der St. Galler Ansatz – wie andere Modellierungsansätze auch – verschiedene Gestaltungsebenen ab (Winter, 2011). Diese Gestaltungsebenen ermöglichen eine bessere Beherrschbarkeit komplexer Modelle sowie die Berücksichtigung unterschiedlicher Zielsetzungen der Modellierung (Winter, 2003). Durch ein hierarchisches Ebenenmodell wird sichergestellt, dass diese Ebenen zusammen ein konsistentes Gesamtbild erzeugen (Winter, 2003). Das St. Galler Business Engineering Modell unterscheidet fünf Modellierungsebenen (Winter & Fischer, 2007; Winter, 2011):

1. Strategieebene: Auf dieser Ebene wird das Unternehmen im Wertschöpfungsnetzwerk modelliert. Somit entsteht ein Modell der Strategie und strategischen Positionierung des Unternehmens.
2. Organisationsebene: Auf dieser Ebene werden die Prozesse und die Aufbauorganisation des Unternehmens modelliert.
3. IT/Business-Alignmentebene: Auf dieser Ebene werden die Strukturen der Informationssysteme im Kontext der fachlichen Prozesse modelliert.
4. Softwareebene: Auf dieser Ebene werden die Software Systeme modelliert, beispielsweise Software Systeme und Datenstrukturen.

5. Infrastrukturebene: Auf dieser Ebene werden die physischen Hardware und Netzwerkstrukturen modelliert.

Diese Ebenen werden auf das Business Engineering Metamodell abgebildet. Innerhalb der Ebenen kann zusätzlich eine Komponentenbildung über den Modelltyp abgebildet werden, die geeignet ist, besonders stark gekoppelte Artefakttypen innerhalb einer Ebene als Ausschnitt darzustellen (Winter, 2011). Weiterhin besteht die Möglichkeit, im BE Business-Metamodell Sichten zu bilden, die bestimmte Aspekte auch über Ebenengrenzen hinweg darstellen (Winter & Fischer, 2007).

Zusätzlich zu dem Metamodell benötigt der Business Engineer auch Vorgehensmodelle und –techniken, mit deren Hilfe der Transformationsprozess umgesetzt werden kann (Österle u. a., 2007). Diese werden in der Forschung (Artefaktkonstruktion), so wie zum Beispiel in dieser Arbeit, aber auch in der Praxis entwickelt.

5 Forschungsbeitrag dieser Dissertation

Die vorliegende kumulative Dissertation entwickelt einerseits Methoden- und Modellartefakte für die Gestaltung und die Verbesserung von Services sowie andererseits zu deren Einordnung ein umfassendes Metamodell des Dienstbegriffes als Erweiterung des BE Business-Metamodells. Diese Erweiterung ist insofern notwendig, als die in den zugrundeliegenden Einzelbeiträgen entwickelten Methoden und Modelle jeweils einen spezifischen Servicebegriff voraussetzen (siehe Teil B). Dieser Servicebegriff hat sich jedoch während des Entstehungszeitraumes der Beiträge weiterentwickelt. Daher sind unterschiedlichen Technologietrends und Paradigmen im Zusammenhang mit dem Begriff „Service“ zu sehen. Für eine bessere Einordnung und Nachvollziehbarkeit der einzelnen Beiträge erscheint es daher sinnvoll, einen Bezugsrahmen zu entwickeln (gemäß (Österle u. a., 2007)). Dieser ermöglicht die Einordnung des den Beiträgen unterliegenden Servicebegriffes und erklärt so die Fokussierung der Beiträge auf einen klar definierten Servicetyp.

Neben der Funktion des Servicemodells als Bezugsrahmen zur Einordnung der Beiträge kann dieses Modell auch als unabhängiges Artefakt verstanden werden. In der Literatur existiert kein ganzheitliches Modell, welches die unterschiedlichen Servicebegriffe im Zusammenhang modelliert (siehe Abschnitt 5.3.1). Durch das entstandene Metamodell sind zusätzlich zwei Aspekte der Forschungsfrage motiviert: einerseits die Strukturierung der Forschungsfragen in Konsumenten- und Produzentensicht sowie andererseits die Betonung der nicht-funktionalen Aspekte im

Rahmen der Methodenentwicklung. Für die Entwicklung des Servicemodells werden zunächst Teilmodelle für die Begriffe SOA-Service, ITSM-Service und CC-Service auf Basis einer Literaturanalyse rekonstruiert. Diese Einzelmodelle werden dann zu einem umfassenden Modell vereint. Das so entstandene Servicemodell wird als Spezialisierung in das BE Business-Metamodell eingefügt (Winter, 2008a, 2011).

Die Ergebnisse der vorliegenden Arbeit lassen sich somit in drei Blöcke unterteilen (siehe auch Abbildung 1). Im Dachbeitrag (Kapitel 5.3) wird das genannte Servicemodell entwickelt und in das BE Business-Metamodell eingefügt (Unterabschnitt 5.3.5.3).

Tabelle 3: Übersicht der Forschungsbeiträge

Beitrag	Fokus	Titel	FF 1.1	FF 1.2	FF 2.1	FF 2.2
Beitrag I	SG	Entwurf eines Enterprise Architecture Frameworks für serviceorientierte Architekturen	●	◐	○	○
Beitrag II	SV	Integriertes Performance Measurement als Teil des Enterprise Architecture Management	○	○	◑	○
Beitrag III	SV	Deriving SOA Evaluation Metrics in an Enterprise Architecture Context	○	○	●	◐
Beitrag IV	SV	A Flexible Approach to Service Management-Related Service Description in SOAs	●	◐	◑	○
Beitrag V	SG	Cloud Computing and the Impact on Enterprise IT	○	◐	○	◑

Legende: ● zentraler Forschungsbeitrag, ◑ wesentlicher Forschungsbeitrag,
 ◐ teilweiser Forschungsbeitrag, ◑ geringer Forschungsbeitrag, ○ kein Forschungsbeitrag

SV: Serviceverbesserung, SG: Servicegestaltung

Die Forschungsfragen 1 und 2 werden im Wesentlichen durch die in Teil B dargestellten Publikationen adressiert. Tabelle 3 stellt die einzelnen Publikationen mit ihren jeweiligen Forschungsbeiträgen übersichtlich zusammen. Die folgenden Unterabschnitte 5.1 und 5.2 fassen die Ergebnisse zu den einzelnen Forschungsfragen zusammen.

5.1 Forschungsfrage 1

Die erste Forschungsfrage befasst sich mit der Problemstellung der Servicegestaltung – unterteilt in eine Nutzer- und Providerperspektive. Inhalte zur Beantwortung der Frage liefern die Beiträge I, IV, V sowie der Dachbeitrag.

Die Beantwortung der FF 1.1 erfolgt direkt durch die Beiträge I und IV. Während Beitrag I eine Methode zur Ableitung von funktionalen Servicebeschreibungen aus Prozessmodellen darstellt, stellt Beitrag IV komplementär dazu ein Modell zur Beschreibung der nicht-funktionalen Eigenschaften dar. Unter Bezugnahme auf das im Dachbeitrag entwickelte Servicemodell können mithilfe dieser Artefakte Services gestaltet werden.

Mit der Servicegestaltung beschäftigen sich einige Publikationen, die in der Darstellung der Forschungslücke (Kapitel 3.3) aufgeführt sind. Die vorliegende Dissertation setzt sich jedoch von den aufgeführten Arbeiten in mehrerlei Hinsicht ab.

Die in Beitrag I präsentierte Methode zur Servicegestaltung basiert auf einem Framework, welches für die Gestaltung von Prozessen und Services zwischen Unternehmen erschaffen wurde und dabei explizit eine doppelte Perspektive, nämlich die des Serviceproduzenten und –konsumenten einnimmt (Teil B – Kapitel 1.5). Der Beitrag zeigt, dass dieses Framework verwendet werden kann, um auch unternehmensinterne Services zu gestalten, wobei die grundlegende Struktur des Vorgehens beibehalten wird. Die in dem Beitrag erweiterte Methode ist somit in der Lage, sowohl unternehmensinterne als auch übergreifende Services zu gestalten (Dietrich, 2008). Es gibt zwar einige Beiträge, die auf die Unterstützung unternehmensübergreifender Prozesse durch Services abstellen (Bianchini u. a., 2014; Shiang, Rau, & Lin, 2009). Dabei betrachten sie jedoch nur die inter-organisationalen Prozesse. Das in Beitrag I gezeigte Vorgehen stellt eine Besonderheit dar, da es ausgehend von einer existierenden Methode der inter-organisationalen Servicegestaltung diese um die intra-organisationale Sichtweise ergänzt. Damit kann die Methode nicht nur die Trennung von Nutzer- und Providerperspektive abbilden, sondern ist ebenso in der Lage, zwischen internen und externen Services zu unterscheiden. Diesen Anforder-

rungen werden die existierenden Methoden nicht gerecht (Gu & Lago, 2010; Huergo u. a., 2014).

Als weitere Forschungslücke mit Bezug auf FF 1.1 wird die unzureichende Betrachtung nicht-funktionaler Aspekte bei der Gestaltung von Services genannt (Huergo u. a., 2014). Diesem Aspekt widmet sich Beitrag IV, in welchem ein Modell für die Beschreibung und das Management nicht-funktionaler Aspekte von Services entwickelt wird.

Das rekonstruierte Servicemodell verdeutlicht diese Anforderungen an die Servicegestaltung. Die Anforderungen zur Modellierung nicht-funktionaler Serviceeigenschaften sowie der Adressierung unterschiedlicher Stakeholder lassen sich direkt aus dem Metamodell ableiten.

Somit lässt sich für FF 1.1 feststellen, dass die Beantwortung der gestellten Forschungsfrage im Rahmen dieser Arbeit umfassend erfolgt. Die dabei entwickelten Methoden und Modelle leisten einen eigenständigen Forschungsbeitrag und erfüllen die Anforderungen, die aus dem definierten Servicemodell an sie gestellt werden.

Die Anforderungen der Servicegestaltung aus der Nutzerperspektive (FF 1.2) werden im Rahmen dieser Arbeit nicht so umfassend geklärt wie das in FF 1.1 behandelte Vorgehen aus Providersicht. Der Dachbeitrag leistet jedoch einen zusätzlichen Beitrag zur Beantwortung dieser Forschungsfrage, indem er aufzeigt, inwiefern sich die Sichtweisen von Produzent und Konsument auf die verschiedenen Servicetypen unterscheiden. Wie bereits ausgeführt, ist die Methode aus Beitrag I grundsätzlich geeignet eine spezifische Nutzerperspektive abzubilden, jedoch erfolgt eine Anwendung der Methode in dem Beitrag I nur aus Providersicht.

Bei Beitrag V handelt es sich um eine explorative Fallstudie (zur Forschungsmethode vgl. (Yin, 2003)). Die Publikation nimmt eine reine Servicekonsumentenperspektive ein und stellt die Anforderungen des Servicekonsumenten anhand zweier exemplarischer Anwendungsfälle dar. Die Arbeit zeigt, welche spezifischen Anforderungen funktionaler und nicht-funktionaler Art die Nutzer eines IaaS CC-Services stellen. Diese Fallstudien bestätigen die Analyse des Dachbeitrages und zeigen, dass bei den Servicekonsumenten insbesondere nicht-funktionale Anforderungen im Fokus stehen. Nicht-funktionale Eigenschaften von Services stellen einen wesentlichen Aspekt des „Dienstvertrages“ zwischen Konsument und Produzent dar und bedürfen daher der expliziten Modellierung der Servicebeziehung zwischen diesen beiden Stakeholdern – wie in Beitrag IV ausgeführt. Diese Eigenschaft der nicht-funktionalen

Parameter wird insbesondere in der geschilderten Anwendungsimplementierung (siehe Beitrag IV) deutlich, welche Funktionalitäten enthält, die es dem Servicenutzer erleichtern einen passenden Service zu identifizieren.

Somit lässt sich für die FF 1.2 festhalten, dass zwar kein einzelner Beitrag die aufgeworfene Fragestellung direkt und umfassend behandelt; doch zeigt der Dachbeitrag auf, inwiefern die Anforderungen an die Servicegestaltung zwischen Servicekonsumenten und –providern divergieren. Diese Ableitung wird durch die Fallstudien bestätigt. Weiterhin stellt Beitrag IV ein Modell und eine Beispielimplementierung für die Modellierung der nicht-funktionalen Eigenschaften vor, die sich zur Anwendung aus der Nutzerperspektive eignen. Jedoch kann die aufgezeigte Methode in Beitrag I nicht als vollumfängliche Beantwortung der FF1.2 gewertet werden, da sie zwar auf Methoden der funktionalen Gestaltung basiert, welche die unterschiedlichen Perspektiven beinhalten, aber diese nicht explizit im Beitrag erläutert.

Zusammenfassend ergibt sich zur Beantwortung der FF1.1 und 1.2, dass die in den Beiträgen entwickelten Methoden und Modelle eine Dienstgestaltung ermöglichen, welche dem Serviceverständnis dieser Arbeit gerecht wird. Es handelt sich hierbei um eine Gestaltung von funktionalen und nicht-funktionalen Eigenschaften eines Dienstes, wobei das aufgezeigte Servicemetamodell die notwendige Einbindung in den Unternehmenskontext ermöglicht. Da sich die Arbeit auf Dienste zwischen der technischen und der fachlichen Ebene des Unternehmens bezieht, muss ein Anwender zusätzlich zu den hier aufgeführten Methoden auch Methoden und Techniken zur Gestaltung und Implementierung der technischen Dienste sowie zur Gestaltung der fachlichen Geschäftsprozesse verwenden.

5.2 Forschungsfrage 2

Die FF 2 wird im Wesentlichen durch die Beiträge II und III sowie wiederum durch den Dachbeitrag behandelt.

Der Dachbeitrag stellt mit Hilfe des Servicemodells dar, welche Services existieren und wie diese in das ganzheitliche Unternehmensmodell eingebunden sind. Dabei stellt sich zunächst die Frage, welches grundsätzliche Vorgehen für die Messung der Serviceleistung angemessen ist. Beitrag II entwickelt eine Methode, die mithilfe einer Kombination unterschiedlicher Methoden ein Verfahren ermöglicht, Serviceleistung ganzheitlich im Unternehmenskontext zu messen. Beitrag III stellt komplementär dazu eine Methode dar, mit deren Hilfe konkrete Ziele für ein bestimmtes Unternehmen abgeleitet werden können. Unternehmen können mithilfe der beiden Vorgehenswei-

sen sowohl eine ganzheitliche Messstruktur entwickeln als auch für die eigene Implementierung die konkreten zu messenden Indikatoren ermitteln. Eine Kombination zweier solcher Methoden wird in verschiedenen Beiträgen in analoger Weise durchgeführt (Buglione & Abran, 2000; Goethert & Fisher, 2003), ist jedoch nicht Teil dieser Arbeit. Während bestehende Arbeiten (Chen u. a., 2006; Vasconcelos u. a., 2007), die ebenso wie Beitrag II ein ganzheitliches Verfahren vorschlagen, sämtliche Ebenen von der Strategie bis zur Technologie mit einer eigenen Methode erfassen, stellt Beitrag II auf die Integration bestehender Methoden ab. Da die integrierten Methoden stark verbreitet sind (wie die Balanced Scorecard für die Strategieebene (Henseler, Jonen, & Lingnau, 2006)), ist die Implementierung des Vorgehens aus Beitrag II einfacher anwendbar. Weiterhin stellen die existierenden Methoden stark auf quantitativ direkt messbare Aspekte der Services ab, wie zum Beispiel die Dienstverfügbarkeit. Mit Beitrag III leistet diese Dissertation eine Erweiterung der Serviceverbesserung mit einem Fokus auf qualitative Eigenschaften des Services.

FF 2.1 wird somit von den vorliegenden Beiträgen ausführlich behandelt. Zum einen helfen sie Unternehmen die unterschiedlichen Services im eigenen Unternehmen in die Architektur einzuordnen (Dachbeitrag), zum anderen schlagen sie konkrete Methoden zur Bewertung von Services vor (Beitrag II und III). Hilfreich für die Umsetzung eines solchen Vorgehens kann weiterhin die Berücksichtigung der in Beitrag IV genannten nicht-funktionalen Eigenschaften von Diensten sein. Die im Beitrag vorgestellten konkreten Indikatoren können als Messwerte einbezogen werden.

FF2.2 wird in keiner der Teilveröffentlichungen als zentrale Fragestellung behandelt. Lediglich Beitrag III beinhaltet eine Perspektive aus Sicht der Nutzer von Services. Weiterhin stellt Beitrag V zwar ausschliesslich die Nutzersicht dar, kann jedoch aufgrund der Forschungsmethode keine umfassende Antwort zu der FF 2.2 leisten, sondern lediglich für die gezeigten Anwendungsfälle Verbesserungsnotwendigkeiten aus Nutzersicht aufzeigen. Aufgrund der aufgezeigten zwei Fälle kann jedoch nicht nachgewiesen werden, dass die Nutzerperspektive grundsätzlich zusätzliche Anforderungen definiert.

Eine umfassendere Antwort auf die FF 2.2 kann jedoch aus dem Dachbeitrag abgeleitet werden. Wie aus Abschnitt 5.3.5.4 ersichtlich, ist ein fundamentaler Bestandteil eines Services die duale Perspektive auf den Dienst aus Nutzer- und Providersicht. Beide Sichtweisen haben unterschiedliche Prioritäten, wenn es um die Bewertung und Verbesserung des Dienstes geht. Wenn bei der Umsetzung der Methoden zur

Dienstverbesserung (Beitrag II und III) dieses Servicemodell als Verständnisgrundlage verwendet wird, sind Nutzer- und Providersicht berücksichtigt.

Zusammenfassend lässt sich für FF 2 feststellen, dass ein ganzheitliches Messprogramm für die erfolgreiche Dienstverbesserung elementar ist. Ganzheitlichkeit kann durch die entwickelten Methoden erzielt werden, sofern das unterliegende Serviceverständnis in einem Unternehmensmodell eingebettet ist. Die dargestellten Methoden ermöglichen die Entwicklung eines Messprogramms, welches sowohl die relevanten Ziele des Servicenutzers als auch diejenigen des Serviceproviders erfasst und messbar macht.

5.3 Entwicklung eines umfassenden Servicemodells

Wie in Abschnitt 1.3 dargestellt, wird der Begriff Service in unterschiedlichen Kontexten verwendet. So existieren für die einzelnen Serviceparadigmen (SOA, CC, ITSM) verschiedene Begriffsdefinitionen und auch Metamodelle (siehe Abschnitte 4.2, 4.1). Weiterhin gibt es in der Literatur auch einzelne Beispiele für generische Servicemetamodelle (Jonkers u. a., 2004). Jedoch existiert kein Beitrag, der die unterschiedlichen spezifischen Servicebegriffe in einem gemeinsamen Modell abbildet und somit deren Einordnung ermöglicht. Dieses Ziel wird in den folgenden Unterabschnitten avisiert. Damit wird einerseits ein Bezugsrahmen für die unterschiedlichen Servicebegriffe erzeugt; andererseits ermöglicht ein solches Metamodell die Einbettung in ein ganzheitliches Unternehmensmetamodell, im vorliegenden Fall das BE Business-Metamodell.

Zur Erstellung des Modells werden zunächst die einzelnen Teilmodelle (SOA, CC, ITSM) aus der Literatur rekonstruiert. Dazu werden in den folgenden Abschnitten die vorhandenen Forschungsbeiträge auf ihr jeweiliges Begriffsverständnis hin untersucht und die wesentlichen Komponenten des jeweiligen Metamodells extrahiert. In einem weiteren Schritt werden die Teilmodelle in ein ganzheitliches und detailliertes Metamodell integriert. Dieser Teil der Dissertation befasst sich mit der Erstellung eines Konstrukts, welches das Vokabular der Forschungsdomäne erfasst (March & Smith, 1995).

Die Analyse der unterschiedlichen Begriffe ergab, dass die Komponenten der Modelle sinnvoll mit Hilfe dreier Kategorien strukturiert werden können, da alle Definitionen Elemente dieser Kategorien beinhalten:

1. Kategorie Servicetypen: Enthält Komponenten für die in den jeweiligen Beiträgen genannten Servicetypen sowie Assoziationen für deren Verhältnis zueinander.
2. Kategorie Serviceattribute: Enthält Komponenten für die in den Beiträgen genannten wesentlichen Eigenschaften der Services und Assoziationen für deren Beziehungen zum Service.
3. Kategorie Servicestakeholder: Enthält Komponenten für Stakeholder der Services und deren Beziehungen zu den Servicetypen in Form von Assoziationen.

Aus diesen Komponenten und Assoziationen wird je ein Modell für jeden Servicebegriff rekonstruiert. Als Ausgangsbasis werden die in der Literaturanalyse identifizierten Beiträge (siehe Abschnitt 3) verwendet.

Im weiteren Verlauf dieses Abschnitts werden zunächst die Rekonstruktionen der Servicemodelle aus den SOA-Beiträgen, ITSM-Beiträgen und CC-Beiträgen einzeln dargestellt (Abschnitte 5.3.2, 5.3.3 und 5.3.4). Diese Modelle werden anschliessend in einem Modell konsolidiert und das so entstandene Gesamtmodell in das BE Business-Metamodell (Abschnitt 0) integriert. Vor diesen Inhalten wird im folgenden Unterkapitel ein Überblick über die in der Literatur aufgefundenen Servicemodelle gegeben.

5.3.1 Existierende Definitionen und Modelle des Servicebegriffes

Trotz der sehr umfangreichen Literatur mit Bezug zum Begriff „Service“ gibt es nur wenige Metamodelle des Begriffs, die nicht nur einen spezifischen Servicetypen (z.B. Webservices) modellieren.

Die Open Group hat einen Standard für die Unternehmensmodellierung namens „Archimate“ definiert (Iacob, Jonkers, Lankhorst, & Proper, 2012), der ein umfassendes Servicemodell beinhaltet. Archimate definiert zunächst die drei Ebenen „Business Layer“, „Application Layer“ und „Technology Layer“ und innerhalb dieser wiederum sieben unterschiedliche Domänen die Inhalte von Prozessen bis zur technischen Infrastruktur abdecken (Iacob u. a., 2012; Jonkers u. a., 2004). Für die jeweiligen Ebenen gibt es ein definiertes Metamodell, in dem wiederum ein spezieller Dienstyp definiert wird. Entsprechend werden in dem Beitrag drei Dienstypen genannt: „Organisational Service“, „Application Service“ und „Infrastructure Service“. Somit stellt das Archimate Metamodell ein umfassenden Servicebegriff als Modell dar, hilft jedoch

nicht bei der Einordnung der spezifischen Servicebegriffe wie SOA-Service und CC-Service in dieses Modell.

Arni-Bloch & Ralyté (2010) entwickeln ein umfassendes Metamodell mit der Zielsetzung der Integration unterschiedlicher IT-Services. Das Modell basiert auf der Annahme, dass eine rein funktionale Aussensicht nicht ausreichend sei und beschreibt die Binnenstruktur eines Services anhand von drei Aspekten: der Datenstruktur des Services und ihrer Semantik, des Serviceverhaltens in Form der Serviceoperationen sowie der Regeln, die bei der Erstellung des Services eingehalten wurden (Arni-Bloch & Ralyté, 2010). Das Modell ist formal beschrieben und fokussiert auf technische Services. Explizit wird auch die Relevanz des Modells für SOA-Services benannt (Arni-Bloch & Ralyté, 2010).

Es existieren weitere Beiträge, die Metamodelle für spezifische Dienstypen entwickeln. Diese werden in den folgenden Unterabschnitten mit aufgeführt (Braun & Winter, 2007; Dwivedi & Kulkarni, 2008; Frank, Strecker, Kattenstroth, & Heise, 2011; Goeken u. a., 2009; Weigand & Johannesson, 2009).

5.3.2 Rekonstruktion eines SOA-Servicemodells

Trotz der Vielzahl der Beiträge zu SOA befassen sich nur wenige Beiträge mit der Erstellung eines Metamodells von SOA-Services (Arni-Bloch & Ralyté, 2010; Dwivedi & Kulkarni, 2008; Weigand & Johannesson, 2009). Zusätzlich definieren einige Beiträge Metamodelle technischer Services wie zum Beispiel Webservices (Fokaefs & Stroulia, 2013; Simon, Goldschmidt, & Kondorosi, 2013).

Ein wesentlicher weiterer Anhaltspunkt zum Verständnis des Servicebegriffes der Beiträge zum Thema SOA kann in den in der Literatur behandelten Methoden zur Servicegestaltung gefunden werden. Da jedes vorgeschlagene Vorgehen zur Gestaltung von Services implizit oder explizit einen Servicebegriff beinhaltet, werden die in der Literaturanalyse (Abschnitt 3.1) identifizierten Beiträge auch in diesem Abschnitt herangezogen.

Basierend auf diesen Quellen lassen sich vier unterschiedliche Servicearten unterscheiden (siehe Tabelle 4). Die Auflistung verdeutlicht, dass diese Unterteilung der Servicetypen von einem Großteil der Beiträge so vorgenommen wird.

Tabelle 4: Typen von Services aus der SOA Literatur

Servicetyp	Beschreibung	Definiert in
Composite Service	Ein Composite Service ist ein Service, der sich aus mehreren Business-Services zusammensetzt.	(Beverungen u. a., 2008; Dwivedi & Kulkarni, 2008; Fareghzadeh, 2008; Klose u. a., 2007; Shirazi u. a., 2009)
Business-Service	Ein Business-Service repräsentiert eine in Software implementierte, fachliche Funktion (Aufgabe, Aktivität). Dies kann beispielsweise die Berechnung eines Kreditrisikos für einen Kunden sein.	(Beverungen u. a., 2008; Dwivedi & Kulkarni, 2008; Fareghzadeh, 2008; Klose u. a., 2007; Shirazi u. a., 2009; Weigand & Johannesson, 2009)
IT-Service	Ein IT-Service repräsentiert eine technologisch orientierte Funktionalität, die bei der Umsetzung eines Business-Services mit einbezogen wird. Dies kann beispielsweise das Versenden eines Datensatzes an ein Partnersystem sein.	(Dwivedi & Kulkarni, 2008; Fareghzadeh, 2008; Shirazi u. a., 2009; Weigand & Johannesson, 2009)
Datenservice	Ein Datenservice repräsentiert eine Funktionalität, welche Funktionen für das Management fachlicher Datensätze bereitstellt, beispielsweise das Erzeugen, Modifizieren etc. eines Kundendatensatzes.	(Beverungen u. a., 2008; Dwivedi & Kulkarni, 2008; Fareghzadeh, 2008; Klose u. a., 2007; Shirazi u. a., 2009; Weigand & Johannesson, 2009)

Diese Servicetypen bauen hierarchisch aufeinander auf. So besteht ein Composite Service aus mehreren Business-Services, die wiederum durch IT- und Datenservices unterstützt werden. IT Services können auch Datenservices unterstützen. Einige Beiträge nennen weitere Servicetypen wie beispielsweise Partnerservices oder Webservices (Dwivedi & Kulkarni, 2008; Gu & Lago, 2010). Da diese jedoch nur von weni-

gen Autoren genannt werden und spezifische Unterformen der aufgelisteten Services darstellen, werden diese Servicetypen nicht weiter betrachtet.

Die zweite Kategorie der zu rekonstruierenden Komponenten wird durch sämtliche Konzepte gebildet, die als Attribute der Services genannt werden. Diese Konzepte sind für SOA-Services in Tabelle 5 aufgeführt.

Tabelle 5: Attribute des SOA Servicemodells

Konzepte	Beschreibung	Definiert in
Interface / Schnittstelle	Die Schnittstelle des Services beschreibt die funktionalen Eigenschaften des Services sowie die Zugriffsmöglichkeit auf den Dienst. Jeder Service hat genau eine Schnittstelle, deren Definition vom Dienstyp abhängt. Während für Business-Services häufig semantisch definierte Schnittstellen gefordert werden (Weigand & Johannesson, 2009), sind IT-Services über rein syntaktische Schnittstellen beschrieben.	(Arsanjani u. a., 2008; Beverungen u. a., 2008; Fareghzadeh, 2008; Kohlmann & Alt, 2007)
Nicht-funktionale Beschreibung / Service-Level-Agreement	Die nicht-funktionale Beschreibung definiert das Verhalten des Services neben den funktionalen Eigenschaften. Je nach Servicetyp haben die relevanten nicht-funktionalen Eigenschaften unterschiedliche Ausprägungen. Bei technischen Services (IT-Services, Datenservices) stehen technische Parameter im Vordergrund, wie beispielsweise Skalierungs- und Verfügbarkeitsinformationen. Bei Business-Services hingegen handelt es sich um Dinge wie Kosteninformationen oder Sicherheitsaspekte. Ein einzelner Service kann unterschiedliche nicht-funktionale Spezifikationen unterstützen.	(Ait Ali Slimane, Pinheiro, & Souveyet, 2009; Kritikos & Plexousakis, 2009; McGabe, Estefan, Laskey, & Thornton, 2012; Teka, Condori-Fernandez, & Sapkota, 2012)

Tabelle 5: Attribute des SOA Servicemodells

Konzepte	Beschreibung	Definiert in
Service Repository	Als Service Repository wird eine Infrastruktur bezeichnet, in der Services registriert werden können, um somit einfach auffindbar zu sein. Ein Repository stellt daher Funktionen zur Registrierung und Suche zur Verfügung. Ein einzelner Service kann in einem oder mehreren Repositories registriert werden.	(Dwivedi & Kulkarni, 2008; Klose u. a., 2007; Teka u. a., 2012)
Orchestration Choreographie	Orchestration und Choreographie beschreiben eine Aufrufabfolge mehrerer Services. Ein Composite Service besteht aus mehr als einem Service sowie einer Beschreibung der Orchestration dieser Dienste.	(Arsanjani, 2004; Dwivedi & Kulkarni, 2008; Kohlmann & Alt, 2007)

Bezüglich der relevanten Stakeholder nennt der Großteil der Beiträge lediglich einen „Service-Owner“. Nur wenige Beiträge differenzieren die Stakeholder nach Rollen wie Service-Owner, Service-Consumer, Service-Designer und Service-Provider (beispielsweise (Jamshidi, Khoshnevis, Teimourzadegan, & Nikravesh, 2009)).

Das sich auf der Basis dieser Servicetypen, Attribute und Stakeholder ergebende Metamodell eines SOA-Services ist in Abbildung 6 abgebildet. Diese Rekonstruktion bildet die wesentlichen existierenden Servicetypen ab: Composite Service, Business-Service, Datenservice und IT-Service. Als weitere Komponenten werden die Attribute Schnittstelle/Interface, nicht-funktionale Beschreibung/Service-Level-Agreement und Service Owner dargestellt, die in einem Großteil der Beiträge als wesentlich für die Servicedefinition eingestuft werden. Zusätzlich nennen viele Autoren die Registrierung eines Services in einem Service Repository als relevante Eigenschaft eines SOA-Services.

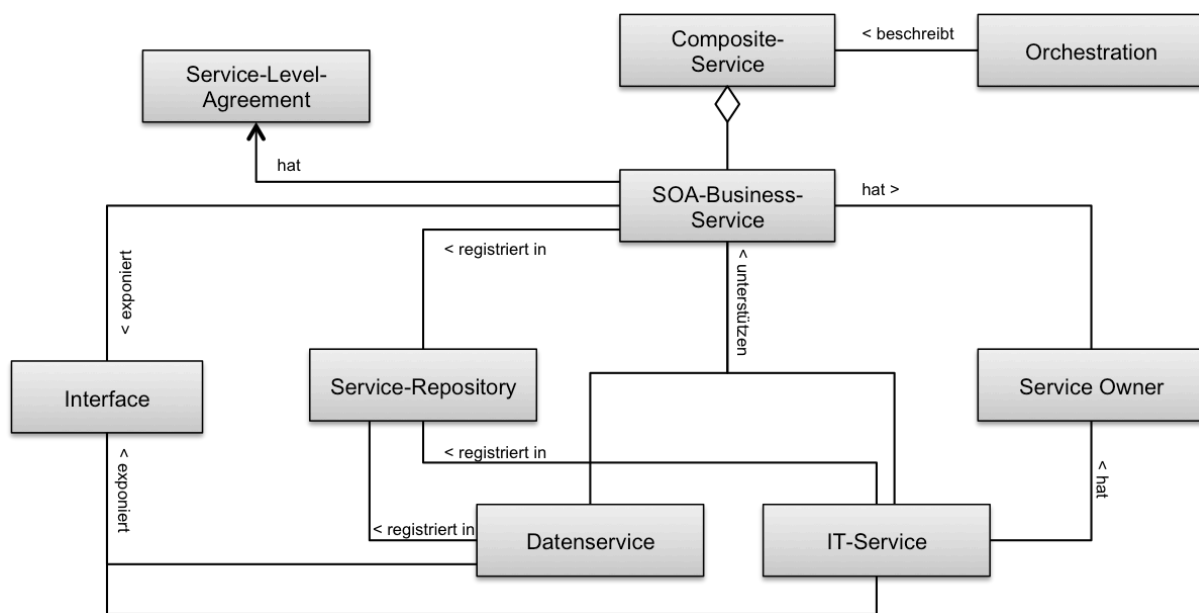


Abbildung 6: Metamodell SOA Service

5.3.3 Rekonstruktion eines ITSM-Service-modells

In den meisten ITSM Frameworks wird eine generische Definition des Servicebegriffes gewählt (siehe Abschnitt 4.3). Es gibt jedoch einige Beiträge, die ITSM Frameworks als Metamodell abbilden (Braun & Winter, 2007; Frank u. a., 2011; Goeken u. a., 2009). Außerdem liegt ein umfassender Vergleich des Serviceverständnisses von sieben unterschiedlichen ITSM Frameworks vor (Mora & Raisinghani, 2014). Fünf dieser Frameworks werden über die betrachteten Arbeiten ebenso in die Rekonstruktion einbezogen:

- ITIL (Braun & Winter, 2007; Cartlidge u. a., 2007; The Stationary Office, 2011)
- CobIT (Control Objectives for Information and Related Technology) (Heschl, Gaulke, & Bitterli, 2005)
- CMMI-SVC (Capability Maturity Model Integration – Service) (CMMI Team, 2010)
- ITUP (IBM Tivoli Unified Process) (Enterprise Management Associates, 2006)
- MOF (Microsoft Operations Framework) (Microsoft Corporation, 2008)

Die zwei verbliebenen Arbeiten sind zum einen unspezifisch und wenig detailliert (ISO, 2005), zum anderen durch die nachfolgende Version bereits vertreten (The Stationary Office, 2011). Auf dieser Basis lassen sich zwei unterschiedliche Servicetypen identifizieren (Tabelle 6).

Tabelle 6: Servicetypen aus der ITSM Literatur

Servicetyp	Beschreibung	Definiert in
ITSM-Service	Ein ITSM-Service wird als nutzbringende Dienstleistung definiert, die immaterielle Güter liefert. Ein Dienst kann auch aus mehreren Diensten bestehen.	(Beims, 2012; Braun & Winter, 2007; CMMI Team, 2010; Goeken u. a., 2009; Heschl u. a., 2005; The Stationary Office, 2011)
IT-Service	Als IT-Service wird derjenige Service bezeichnet, der im Wesentlichen durch IT-Systeme und zusätzliche Unterstützungsprozesse erbracht wird. Ein IT-Service ist daher eine spezifische Form des ITSM-Services.	(Braun & Winter, 2007; Enterprise Management Associates, 2006; Heschl u. a., 2005; The Stationary Office, 2011)

In den Publikationen wird eine Vielzahl an Konzepten beschrieben, die als Attribute der Services modelliert werden können. Diese sind in Tabelle 7 aufgelistet.

Tabelle 7: Entitäten des ITSM Services

Konzepte	Beschreibung	Definiert in
System	Ein System wird generisch als eine Anzahl interagierender Elemente definiert, die gemeinsam eingesetzt werden.	(CMMI Team, 2010; Heschl u. a., 2005; Microsoft Corporation, 2008; The Stationary Office, 2011)
Service-System	Als Service-System wird ein System bezeichnet, das es ermöglicht einen Service auszuführen. Es handelt sich dabei um eine Kombination technischer und organisatorischer Elemente.	(CMMI Team, 2010; Microsoft Corporation, 2008)
Service-Level Agreement (SLA)	Ein SLA beschreibt einen Teil des Vertrages zwischen Provider und Servicenutzer. Er beschreibt die Qualität des zu erbringenden Dienstes.	(Braun & Winter, 2007; Cartlidge u. a., 2007; CMMI Team, 2010; Heschl u. a., 2005; Microsoft Corporation, 2008)
Operations-Level Agreement (OLA)	Ein OLA beschreibt die zu erbringende Dienstqualität innerhalb der Organisation, die den Dienst bereitstellt. Zusätzlich kann ein OLA auch die Leistungen von Lieferanten des Dienstbereitstellers beinhalten.	(Braun & Winter, 2007; Frank u. a., 2011; Heschl u. a., 2005; Microsoft Corporation, 2008)

Tabelle 7: Entitäten des ITSM Services

Konzepte	Beschreibung	Definiert in
IT-Prozesse	Als IT-Prozesse werden jene Geschäftsprozesse bezeichnet, die zur Bereitstellung des IT-Services benötigt werden. Ein IT-Service kann mehrere IT-Prozesse benötigen und ein einzelner IT-Prozess kann mehreren IT-Services dienen.	(Braun & Winter, 2007; Cartlidge u. a., 2007; CMMI Team, 2010; Enterprise Management Associates, 2006; Heschl u. a., 2005)
Servicekatalog	Der Servicekatalog beinhaltet die Services und ermöglicht das vereinfachte Auffinden.	(Braun & Winter, 2007; Cartlidge u. a., 2007; CMMI Team, 2010; Frank u. a., 2011; Heschl u. a., 2005)

In der Kategorie Stakeholder nennen viele Beiträge drei unterschiedliche Stakeholder, die als entsprechende Komponenten zu modellieren wären. Diese sind in Tabelle 8 aufgeführt. Einige Arbeiten differenzieren die Stakeholder weiter aus, wie im Falle des Partners aufgeführt (in Hersteller, Verkäufer, etc.). Anhand dieser drei Kategorien von Komponenten (Servicetypen, Attribute und Stakeholder) lässt sich ein Metamodell des ITSM-Services rekonstruieren, wie es in Abbildung 7 dargestellt ist. Dieses Modell unterscheidet sich von dem SOA-Servicemodell im Wesentlichen durch die weniger differenzierte Unterscheidung der Servicetypen sowie die umfangreichere Gestaltung des Services als Schnittstelle zwischen einem Konsumenten und Produzenten. Dies wird deutlich an den explizit genannten Stakeholdern „Konsument“ und „Produzent“ sowie auch an der Differenzierung zwischen SLA und OLA.

Tabelle 8: Stakeholder von ITSM-Services

Stakeholder	Beschreibung	Definiert in
Provider (Serviceprovider / IT-Serviceprovider)	Als Serviceprovider wird die Organisation bezeichnet, die den Dienst definiert und dem Kunden zur Verfügung stellt. Der Provider ist häufig auch derjenige Stakeholder, der den Dienst produziert, jedoch kann ein Dienst auch teilweise oder ganz durch einen Partner (s.u.) erfolgen.	(Braun & Winter, 2007; CMMI Team, 2010; Heschl u. a., 2005; Microsoft Corporation, 2008)
Servicekonsument	Der Konsument des Services nutzt den Dienst und hat einen entsprechenden Bedarf.	(Cartlidge u. a., 2007; CMMI Team, 2010)
Partner	Der Partner hilft dem Provider bei der Produktion des Dienstes als Dienstleister, Hersteller oder Verkäufer von notwendigen (Teil-)Leistungen. Ein Partner kann sowohl wiederum einen Partner haben als auch die Unterstützungsleistung in Form eines Services erbringen.	(Cartlidge u. a., 2007; CMMI Team, 2010)

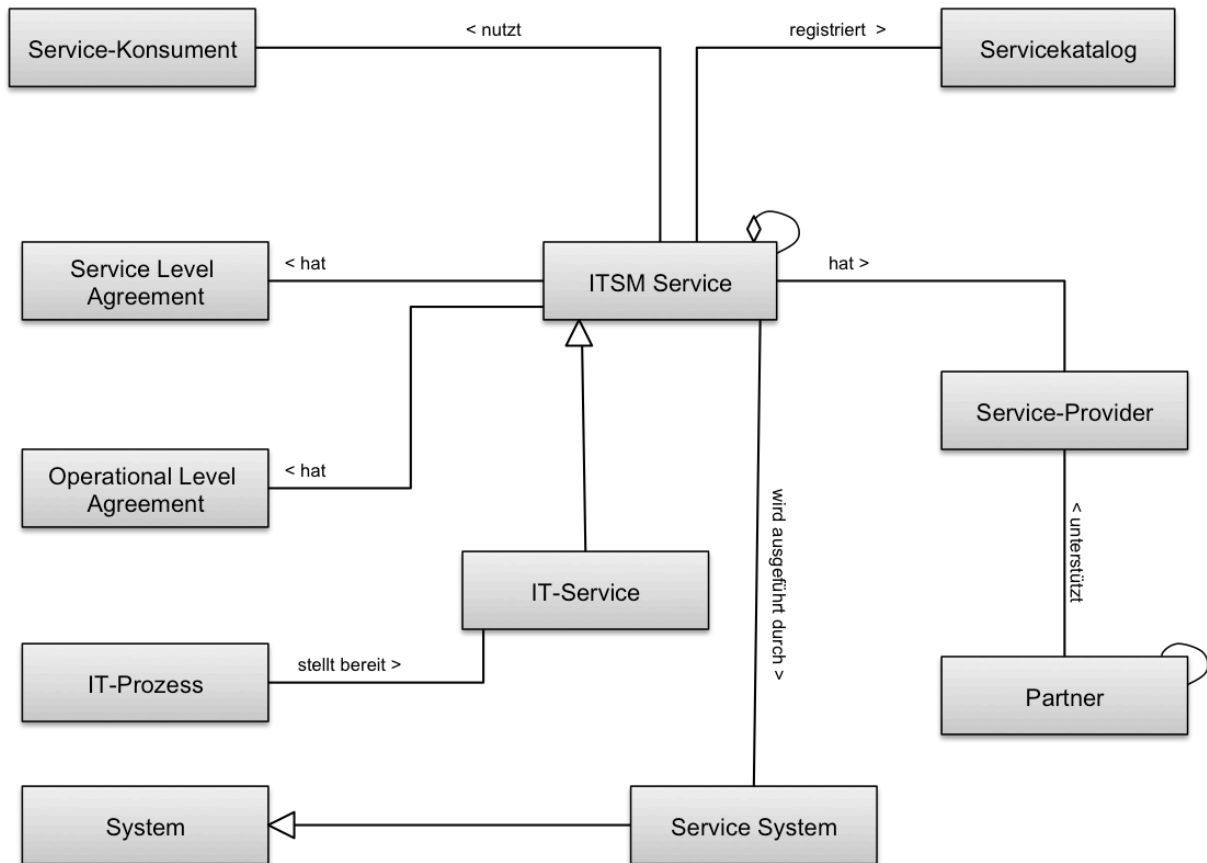


Abbildung 7: Metamodell ITSM Service

5.3.4 Rekonstruktion eines Cloud Computing-Servicemodells

Nur wenige Beiträge zum Thema Cloud Computing befassen sich mit der Erstellung eines Metamodells des Cloud Computing-Servicebegriffes (Chatziprimou, Lano, & Zschaler, 2013; Hamdaq, Livogiannis, & Tahvildari, 2011). Einige weitere Beiträge liefern eine Definition des Cloud-Servicebegriffes (Armbrust u. a., 2009; Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009; Mell & Grance, 2011; Vaquero, Roderomero, Caceres, & Lindner, 2008; L. Wang u. a., 2008; Youseff u. a., 2008). Viele dieser Definitionen basieren auf der Publikation des National Institute of Standardization and Technology (NIST) (Mell & Grance, 2011). Durch diese Fokussierung auf wenige Beiträge entsteht eine weitestgehend homogene Darstellung des Servicebegriffes in der Literatur.

Die CC-Servicedefinitionen basieren auf einem Serviceverständnis, welches mit dem Begriff IT-Service aus der ITSM-Definition korrespondiert. Es handelt sich um IT-fokussierte Services, die durch unterschiedliche Prozesse bereitgestellt werden (Liu

u. a., 2011). Auf diesem Basisbegriff aufbauend folgen die meisten Autoren einer Servicetypologie mit drei Typen von CC-Services, die in Tabelle 9 dargestellt werden (basierend auf (Mell & Grance, 2011)).

Tabelle 9: CC-Servicetypen

Servicetyp	Beschreibung	Definiert in
Software-as-a-Service (SaaS)	SaaS beschreibt eine Serviceart, bei der ein Cloud Provider eine ganze Geschäftsapplikation bereitstellt. Dieser Servicetyp kann auf den folgenden Servicetypen basieren.	(Aboulnaga u. a., 2009; Mell & Grance, 2011; Repschläger & Zarnekow, 2011; Vaquero u. a., 2008; Youseff u. a., 2008)
Platform-as-a-Service (PaaS)	PaaS beschreibt einen Servicetyp, der es dem Nutzer ermöglicht eigene Applikationen zu entwickeln und in der Plattform des Cloudanbieters auszuführen. Dabei werden von dem Cloudanbieter bereitgestellte Programmiersprachen, Softwarekomponenten oder –dienste verwendet. Dieser Servicetyp kann auf einem IaaS basieren.	(Aboulnaga u. a., 2009; Mell & Grance, 2011; Repschläger & Zarnekow, 2011; Vaquero u. a., 2008; Youseff u. a., 2008)
Infrastructure-as-a-Service (IaaS)	Mit IaaS werden Services bezeichnet, auf denen ein Nutzer Rechenleistung wie Prozessorleistung oder Speicherdienste beziehen kann.	(Aboulnaga u. a., 2009; Mell & Grance, 2011; Repschläger & Zarnekow, 2011; Vaquero u. a., 2008; Youseff u. a., 2008)

Einige Autoren nennen weitere Servicetypen wie beispielsweise Datenbanken als Cloud Service (DaaS), Geschäftsprozesse als Cloud Service (BPaaS) oder manuelle Dienstleistung als Cloudservice (Human as a Service) (Klems, Nimis, & Tai, 2009; Motahari-Nezhad, Stephenson, & Singhal, 2009; Repschläger & Zarnekow, 2011). Diese Begriffe sind jedoch nicht weit verbreitet.

Zusätzlich differenzieren sich Cloud Services durch eine Reihe von grundlegenden Eigenschaften von anderen IT-Services (nach (Mell & Grance, 2011)):

1. On-demand Self-service: Der Nutzer ist in der Lage, die Services selbstständig bereitzustellen, ohne dabei Unterstützung des Providers zu benötigen.
2. Broad Network Access: Services sind über Netzwerke verfügbar und können mit Standardtechnologien genutzt werden.
3. Resource Pooling: Die Kapazitäten des Services werden für mehrere Kunden gemeinsam bereitgehalten und den einzelnen Kunden dynamisch zugewiesen. Diese Eigenschaft gilt jedoch nur für „public“ Cloud Services (siehe auch Tabelle 10).
4. Rapid Elasticity: Ressourcen können elastisch nach oben und nach unten skaliert werden, um den Anforderungen des Servicenutzers zu genügen.
5. Measured Service: Die Nutzung der Services wird detailliert und in der Art und Weise dem Service angemessen (z.B. Storage, Processing pro Nutzungstunde oder pro Nutzerzugang) gemessen und abgerechnet, wodurch dem Nutzer eine Möglichkeit der Kontrolle und Nutzungs-Optimierung ermöglicht wird.

Diese spezifischen Eigenschaften werden durch die in Tabelle 10 abgebildeten Komponenten modelliert.

Tabelle 10: Attribute von CC Services

Konzept	Beschreibung	Definiert in
Serviceumgebung	Cloud Services können über verschiedene Betriebsmodelle bereitgestellt werden. Dabei werden allgemein drei Varianten genannt: „private“, „public“ und „hybrid“. Bei einem privaten Betriebsmodell wird die Serviceumgebung ausschliesslich durch einen Konsumenten genutzt. Bei dem „public“-Modell wird vom Konsumenten eine von unterschiedlichen Unternehmen genutzte Umgebung verwendet. Ein hybrides Modell beschreibt eine Kombination aus „public“ und „private“.	(Liu u. a., 2011; Mell & Grance, 2011; Repschläger & Zarnekow, 2011)
Nutzungsberechnung	Gemäss der grundlegenden Anforderung, dass Services nutzungsabhängig abgerechnet werden können, muss für jeden Dienst eine Mess- und Abrechnungslogik für die Dienstnutzung hinterlegt sein. Hinter diesem Attribut ist auch die für eine Nutzungsorientierung notwendige Skalierbarkeit modelliert.	(Armbrust u. a., 2009; Mell & Grance, 2011; Repschläger & Zarnekow, 2011)

Tabelle 10: Attribute von CC Services

Konzept	Beschreibung	Definiert in
API / Schnittstellen	Da Services durch den Nutzer selbstständig angefordert und genutzt werden können, müssen umfassende Schnittstellen zur Dienstnutzung und zum Dienstmanagement vorhanden sein. Weiterhin muss diese Schnittstelle zum Dienst für die Kunden weithin erreichbar sein (siehe oben „broad network access“).	(Buyya, Yeo, u. a., 2009; L. Wang u. a., 2008; Weinhardt u. a., 2009)
Service-Level Agreements (SLAs)	Neben den funktionalen Schnittstellen sollte ein Cloud Service durch nicht-funktionale Eigenschaften definiert sein. Diese sind in den SLAs festgehalten.	(Buyya, Yeo, u. a., 2009; Vaquero u. a., 2008)

Zusätzlich zu den genannten Attributen nennen einige Autoren weitere elementare Eigenschaften wie eine Service Orchestration oder ein Service Inventory (Liu u. a., 2011; Rimal, Jukan, Katsaros, & Goeleven, 2011). Einige Autoren erweitern die möglichen Betriebsmodelle um ein „Community“-Modell, welches als „Public“-Cloud Service definiert ist, an dem jedoch nur ausgewählte Unternehmen teilhaben können (Liu u. a., 2011).

Als dritte Kategorie wurden in den Publikationen die Stakeholder von CC-Services identifiziert. In den Beiträgen werden allgemein drei Stakeholder genannt, die in Tabelle 11 aufgeführt sind. Neben diesen Stakeholdern nennen einige wenige Beiträge weitere wie Cloud Auditor und Cloud Carrier (Liu u. a., 2011).

Tabelle 11: Stakeholder von Cloud Services

Stakeholder	Beschreibung	Definiert in
Cloud-Konsument	Der Cloud-Konsument ist der Nutzer des Cloud-Services.	(Armbrust u. a., 2009; Lenk, Klems, Nimis, Tai, & Sandholm, 2009; Liu u. a., 2011; Mell & Grance, 2011)
Cloud-Provider	Der Cloud-Provider stellt die Cloud-Umgebung als Service bereit und unterhält sämtlich Prozesse, die notwendig sind um die Umgebung den Kunden bereitzustellen.	(Armbrust u. a., 2009; Lenk u. a., 2009; Liu u. a., 2011; Mell & Grance, 2011)
Cloud-Broker	Ein Cloud-Broker vermittelt zwischen dem Cloud-Provider und dem Nutzer. Diese Vermittlung kann als rein kommerzielle Leistung erfolgen (Reselling) oder aber mit einer technischen Komponente. Diese kann durch Integration verschiedener Cloud-Services oder Veredlung eines einzelnen Dienstes erfolgen.	(Armbrust u. a., 2009; Buyya, Ranjan, & Calheiros, 2009; Gottschlich, Hiemer, & Hinz, 2014; Klems u. a., 2009; Liu u. a., 2011)

Auf Basis der dargestellten Typologie der Services sowie der zusätzlichen Attribute und Stakeholder kann ein Metamodell rekonstruiert werden, welches in Abbildung 8 dargestellt ist.

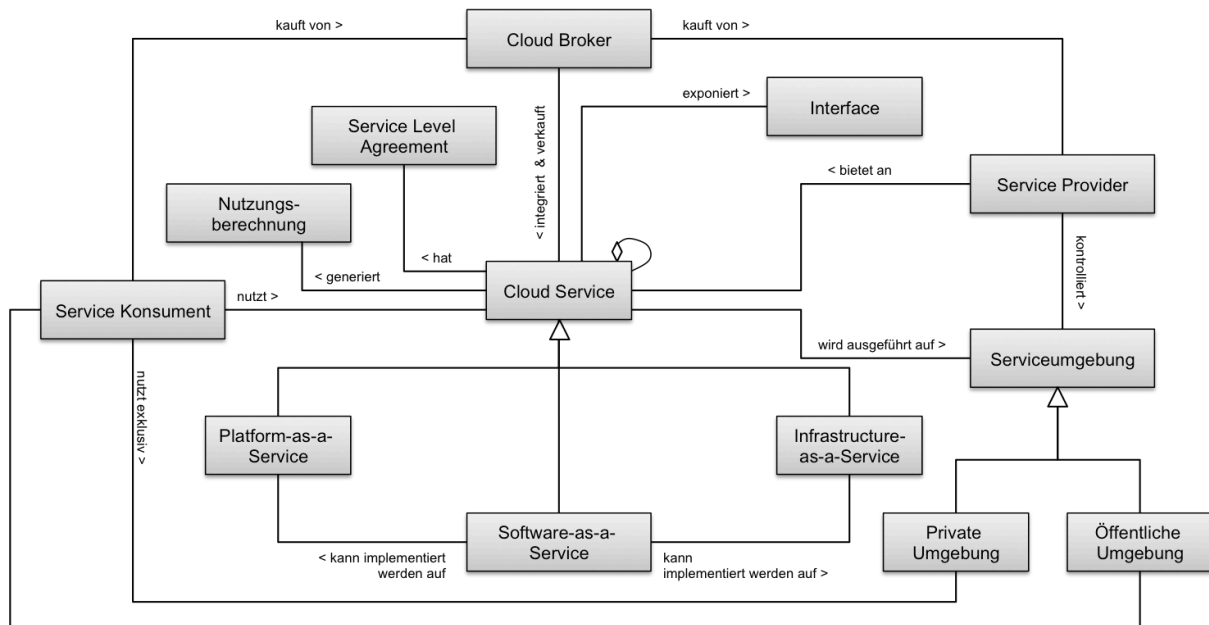


Abbildung 8: Metamodell Cloud Service

Anhand dieses Modells wird im Vergleich zu den bereits vorgestellten deutlich, dass insbesondere die Akteure stärker differenziert werden. Das CC-Servicemodell umfasst drei unterschiedliche Akteure (Konsument, Provider und Broker), während in den anderen Servicedefinitionen lediglich der Service-Provider und -Konsument modelliert wurden. Weiterhin wird ein neuer Aspekt eingeführt, der mit Hilfe der Entität „Serviceumgebung“ modelliert ist. Wie beschrieben, wird bei CC-Services zwischen unterschiedlichen Betriebsmodellen differenziert („private“, „public“, „hybrid“), die sich jeweils dadurch auszeichnen, dass der Service-Konsument entweder exklusiv oder mit anderen Nutzern eine Serviceumgebung nutzt. Zuletzt ist als ein weiterer Aspekt die Nutzungsberechnung zu nennen, die in keinem der anderen Modelle aufgeführt wird.

5.3.5 Konsolidiertes Service-Metamodell

In diesem Abschnitt werden die Modelle der einzelnen Servicebegriffe in einem Gesamtmodell vereinigt. Eine weitere Zielsetzung dieses Abschnittes ist, das so entstandene Servicemodell in dem BE Business-Metamodell zu verankern.

Wie in Abschnitt 4.4 dargestellt, unterscheidet das BE Business Metamodell zwischen fünf unterschiedliche Ebenen (Winter, 2011). Diese Ebenen sind in Abbildung 9 dargestellt. Die hier dargestellten Servicebegriffe beziehen sich auf die Alignmentebene (z.B. Entitäten wie der ITSM-Service) sowie die darunter angeordneten technischen Ebenen (z.B. Daten-Service aus dem SOA-Servicemodell). Die Darstellung

von Teilen des BE Business-Metamodells erfolgt im BE Framework über Modelltypen. Ein Modelltyp ist ein ebenenspezifischer Ausschnitt aus dem BE Business-Metamodell in dem Artefakte mit einer besonders starken Koppelung abgebildet sind (Winter, 2011).

Im Folgenden werden zwei Modelltypen dargestellt: zunächst ein Modelltyp der Services auf der Alignmentebene und anschliessend einer für die technischen Ebenen.

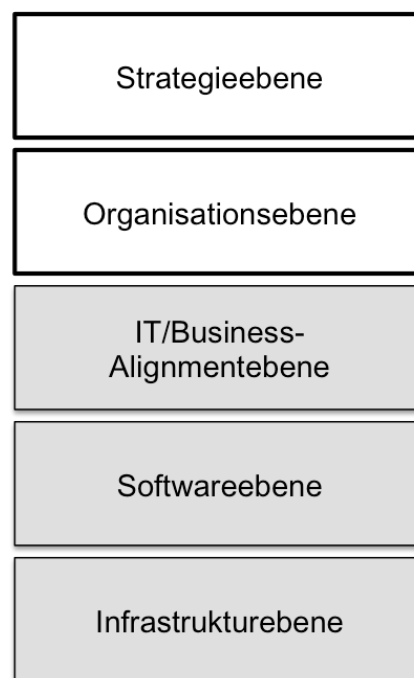


Abbildung 9: BE Business-Metamodell Ebenen

Zur Einordnung in das BE Business-Metamodell ist in Abbildung 10 der relevante Ausschnitt des Modells dargestellt (Winter, 2011). Dabei handelt es sich um die Entitäten der Alignment- sowie der Software- und Infrastrukturebene, ergänzt um Entitäten, die dem besseren Verständnis dienen, aber im Falle einer Instanziierung des Modells nicht einem realen Artefakt entsprechen. Dazu gehören Markt, Unternehmen, Führung, Geschäftsfeld, Informationssystem, Aufbau- und Ablauforganisation (siehe auch (Winter, 2011)).

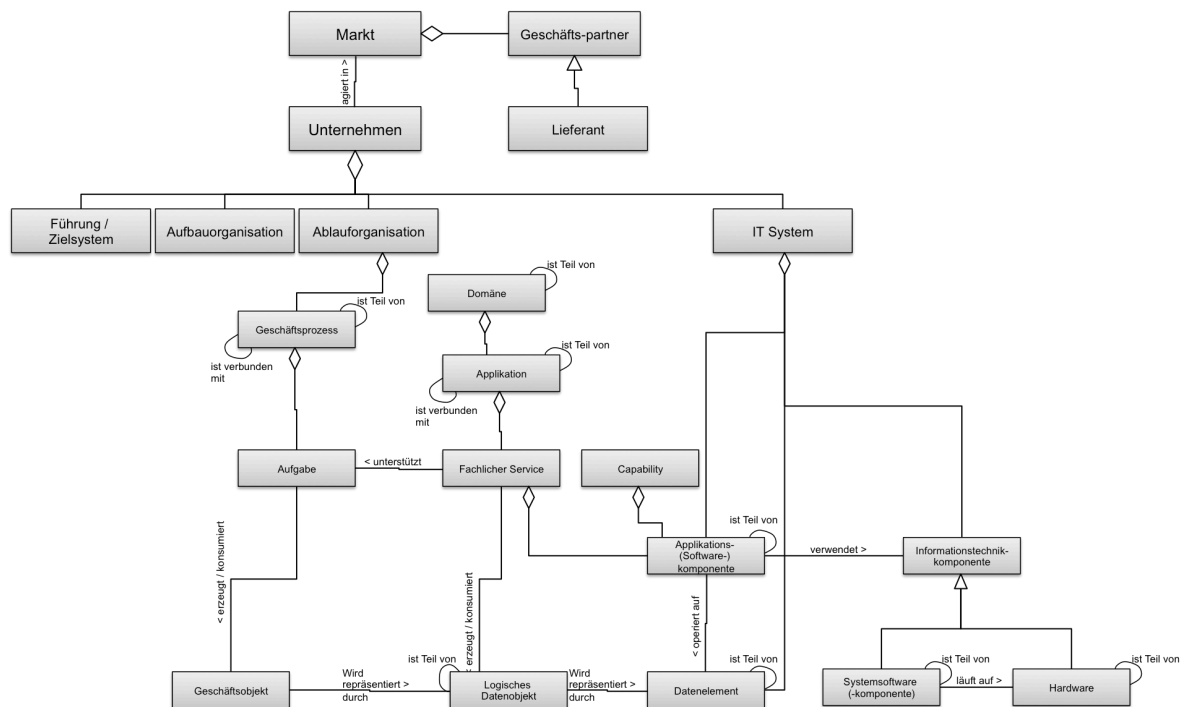


Abbildung 10: BE Business-Metamodell Ausschnitt

In diesem Modell sind bereits Entitäten vorhanden, die für die Modellierung der unterschiedlichen Servicebegriffe relevant sind: Fachlicher Service, Capability, Software Komponente, Informationstechnikkomponente.

Daher besteht die Zielsetzung der folgenden Unterabschnitte einerseits darin, die hier entwickelten Modelle zu integrieren, dabei aber ebenso die existierenden Entitäten des BE Business-Metamodells mit einzubeziehen.

5.3.5.1 Service Metamodelltyp auf der Alignmentebene

Die Alignmentebene dient der Verknüpfung der Entitäten der organisationalen Sicht und der IT-Artefakte (Winter, 2011). Die hier aufgeführten Entitäten des Metamodells sind virtuelle Artefakte, die in einem hierarchischen Verhältnis zueinander stehen und die Komplexität der angrenzenden technischen und organisationalen Ebenen reduzieren sollen (Winter, 2011). Den Anforderungen dieser Ebene entspricht ein Servicetyp, der es ermöglicht, die Stakeholder sowie die funktionalen und nicht-funktionalen Aussenkanten des Services zu beschreiben. Abbildung 11 zeigt diesen Modelltyp für die Alignmentebene. Entlang der Strukturierung nach Servicetypen, Stakeholdern und Attributen soll das Modell im Folgenden erläutert werden.

5.3.5.1.1 Servicetypen auf der Alignmentebene

Es werden drei Servicetypen modelliert: Neben den Spezialisierungen Fachlicher Service und Technischer Service gibt es den generischen Service, der das Äquivalent zu dem ITSM-Service aus dem entsprechenden Metamodell darstellt. Es handelt sich dabei um einen unspezifischen Servicetyp als Generalisierung der Servicetypen Technischer und Fachlicher Service. Der Begriff Fachlicher Service entstammt dem BE Metamodell und wird entlang des fachlichen Geschäftsprozesses gestaltet (Winter, 2011). Der Technische Service hingegen entspricht dem IT-Service Begriff des ITSM-Metamodells. Der Technische Service wird definiert durch die technischen Fähigkeiten und eingesetzten Systeme.

Der Fachliche Service hingegen ist in den entwickelten Metamodellen nicht aufzufinden. Er entspricht einer Generalisierung des SOA-Business-Service, jedoch ohne die notwendige Implementierung des Services (siehe Tabelle 4).

5.3.5.1.2 Attribute der Services auf der Alignmentebene

Für die Services auf der Alignmentebene werden drei Konzepte als Attribute modelliert:

- Entität „Interface“ zur Darstellung der funktionalen Eigenschaften: Alle Services sind durch eine Interface-Beschreibung funktional definiert. Die Schnittstelle definiert sowohl Ein- und Ausgabeparameter als auch das Serviceverhalten. Obwohl für die ITSM-Services in der Literatur keine Schnittstelle definiert wird, erscheint die Modellierung so korrekt, da sämtliche von diesem Servicetyp abgeleitete Services eine Schnittstelle als mit ihr verbundene Entität aufweisen.
- Entität „Service-Level-Agreement“ zur Modellierung der nicht-funktionalen Eigenschaften: Diese Komponente befindet sich in allen Teilmodellen.
- Entität „Servicekomposition“ zur Darstellung komplexer Services: Einzelne Services können miteinander kombiniert und so komplexe Services erzeugt werden. Dies geschieht in der Form von Servicekompositionen. Diese Eigenschaft wird bei den verschiedenen Servicetypen unterschiedlich modelliert, teilweise explizit durch die Definition einer Entität (Composite Services und Orchestration im Falle der SOA Services) oder ohne eigenständige Entität durch Erstellung einer selbstreferenzierenden Aggregationsbeziehung.

5.3.5.1.3 Stakeholder der Services auf Alignmentebene

Das Gesamtmodell enthält drei Stakeholder. Der Service-Konsument bezieht die Serviceleistung, um diese im Wertschöpfungsprozess einzusetzen. Dabei handelt es sich nie um eine technische Gegenstelle, sondern immer um die Organisationseinheit, die den entsprechenden Teil des Wertschöpfungsprozesses verantwortet. Der Service-Produzent hingegen stellt den Service bereit und garantiert funktionale und nicht-funktionale Eigenschaften. Zusätzlich gibt es einen Service-Broker, der verschiedene Dienste aggregiert oder Service-Konsumenten oder Produzenten anderweitig unterstützt. Diese Entität entspricht sowohl dem Partner aus dem ITSM-Servicemetamodell als auch dem Service-Broker aus dem CC-Servicemetamodell. Der spezifischere Begriff des Service-Brokers wurde verwendet, um bei der Einbindung in das generelle BE Business-Metamodell eine Differenzierung zu anderen Partnern des Unternehmens zu erreichen.

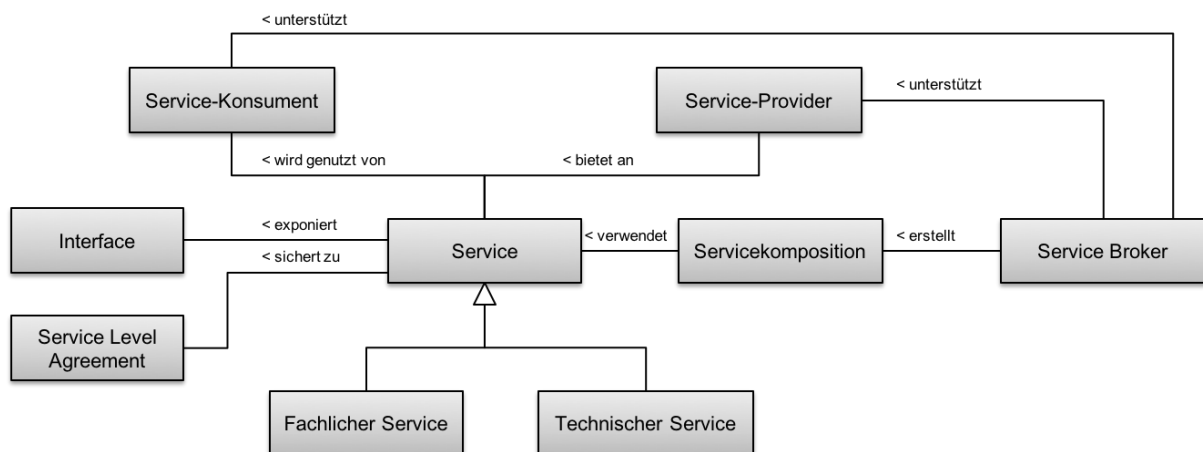


Abbildung 11: Ausschnitt des Metamodells Service auf der Alignmentebene

5.3.5.2 Service Metamodelltyp auf der technischen Ebene

Auf den technischen Ebenen des Business-Metamodells werden die technologischen Aspekte des Unternehmens modelliert. Das St. Galler Modell unterscheidet bei den technischen Artefakten zwischen Softwareartefakten und IT-Infrastrukturen (Winter & Fischer, 2007). Diese Differenzierung ist für die Modellierung der Servicetypen nicht notwendig, da Infrastrukturentitäten nur in geringem Umfang einbezogen werden.

Der in Abbildung 12 dargestellte Ausschnitt des Metamodells beinhaltet sämtliche technische Servicetypen und die mit diesen Entitäten verbundenen Konzepte und Stakeholder.

5.3.5.2.1 Servicetypen auf den technischen Ebenen

Auf den technischen Ebenen des Metamodells werden insgesamt sieben Servicetypen dargestellt. Zusätzlich sind der Fachliche Service und der Technische Service zur Einbindung in das Modell der Alignmentebene dargestellt, auch wenn diese Servicetypen selbst nicht auf den technischen Ebenen zu verorten sind. Aus dem BE Metamodell entstammt der Servicetyp Capability, der eine Gruppierung zusammengehöriger Softwarefunktionalitäten darstellt (Winter, 2011). Die Capability definiert, mit den zugehörigen Unterstützungsprozessen, den Technischen Service. Weiterhin wurde zur Verbesserung der Übersichtlichkeit des Modells eine Entität „Service-Implementierung“ eingeführt. Während die technischen SOA-Servicetypen Datenservice und IT-Service als Spezialisierungen der Capability modelliert sind, entspricht der SOA-Business-Service einer Spezialisierung des Fachlichen Services. Alle drei Servicetypen werden durch eine Service-Implementierung technisch realisiert. Die Cloud-Services stellen wiederum eine Spezialisierung dieser Implementierung mit den spezifischen Eigenschaften eines Cloud-Services dar.

5.3.5.2.2 Attribute der Services auf den technischen Ebenen

Neben den Servicetypen werden in dem Metamodellausschnitt für die technischen Ebenen sechs weitere Entitäten aufgeführt. Mit der Entität „Laufzeitumgebung“ wird die Einbindung in die Infrastrukturebene des BE Business-Metamodells realisiert. Diese Entität entspricht auch dem Service System des ITSM Services. Zu Steigerung der Übersichtlichkeit wurden die Spezialisierungen „Private“ und „Öffentliche Umgebung“ nicht dargestellt. Aus dem Metamodell des ITSM-Services wurde das Operational-Level-Agreement ergänzt und mit der Service-Implementierung verbunden, da sämtliche technischen Services ein solches Agreement aufweisen sollten. Aus dem Cloud-Service-Metamodell wurde die Nutzungsberechnung hinzugefügt, die so auch für SOA-Services darstellbar ist.

5.3.5.2.3 Stakeholder der Services auf den technischen Ebenen

In dem Modellausschnitt der technischen Ebenen werden keine Stakeholder modelliert, da sämtliche Services Spezialisierungen der Services der Alignmentebene sind und somit die Stakeholderbeziehungen von diesen Servicetypen vererbt bekommen.

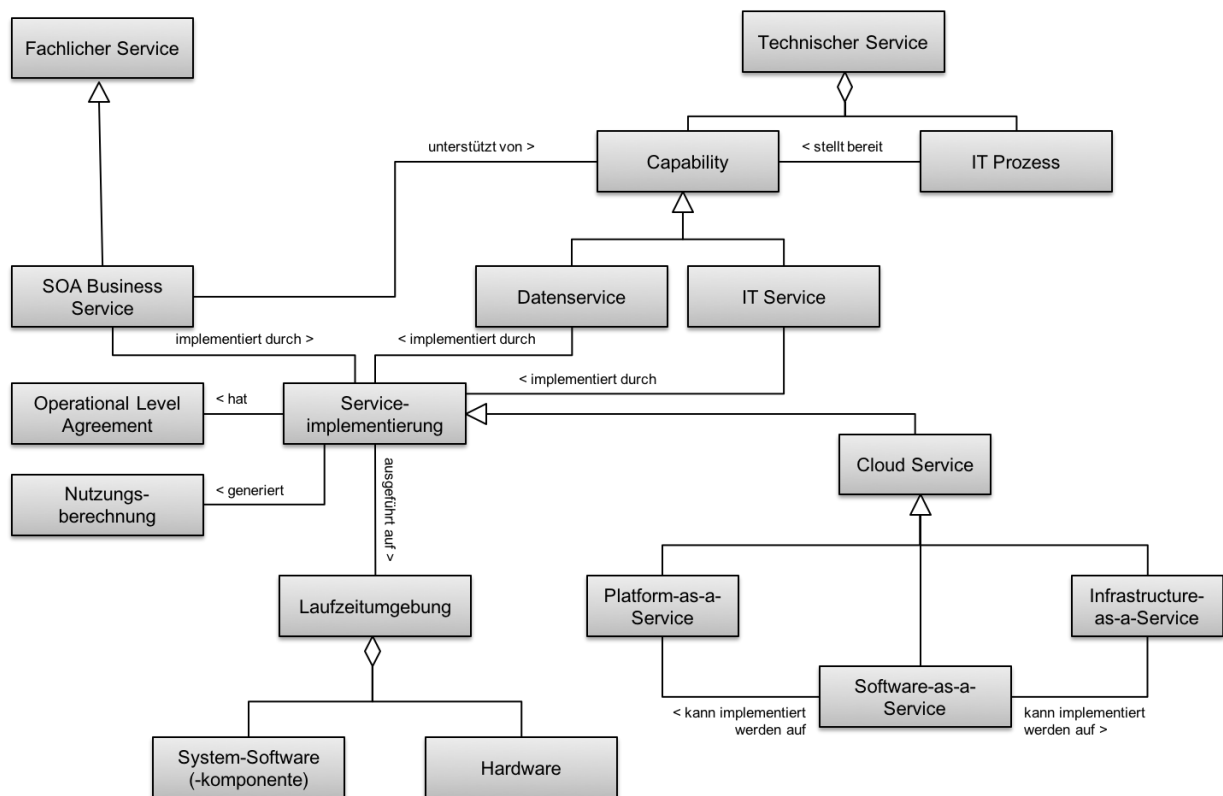


Abbildung 12: Ausschnitt des Metamodells Services auf den Technologieebenen

Die beiden Modelltypen werden im nächsten Schritt in das BE Business-Metamodell eingefügt, was im Ergebnis die angestrebte Servicespezialisierung des BE Business-Metamodells darstellt.

5.3.5.3 Service Spezialisierung des BE Business-Metamodell

Das integrierte Metamodell ist in Abbildung 13 dargestellt (in Appendix A befindet sich zusätzlich eine ganzseitige Version).

In diesem Modell sind die Elemente des existierenden BE Business-Metamodells weiß dargestellt und die für die Spezialisierung zusätzlich modellierten Entitäten in grauer Färbung eingefügt. Zur Vereinfachung der Darstellung wurde die Entität „Servicekomposition“ lediglich anhand einer selbstreferenzierenden Aggregationsbeziehung dargestellt. Zusätzlich wurde die Entität „interner Serviceprovider“ als Spezialisierung des Serviceproviders eingefügt, um so beide Varianten, die der internen Leistungserbringung sowie die der externen Beziehung darstellen zu können.

Zusammenfassend zeigt das Modell, in welcher Relation die unterschiedlichen Servicebegriffe zueinander stehen und wie sie in den Unternehmenskontext einzuordnen sind. Das entstandene Servicemodell zeigt in ganzheitlicher und detaillierter Weise

die unterschiedlichen Servicedefinitionen auf und eignet sich daher für die angestrebte Eingrenzung des Servicebegriffes für die in dieser Arbeit adressierten Servicetypen.

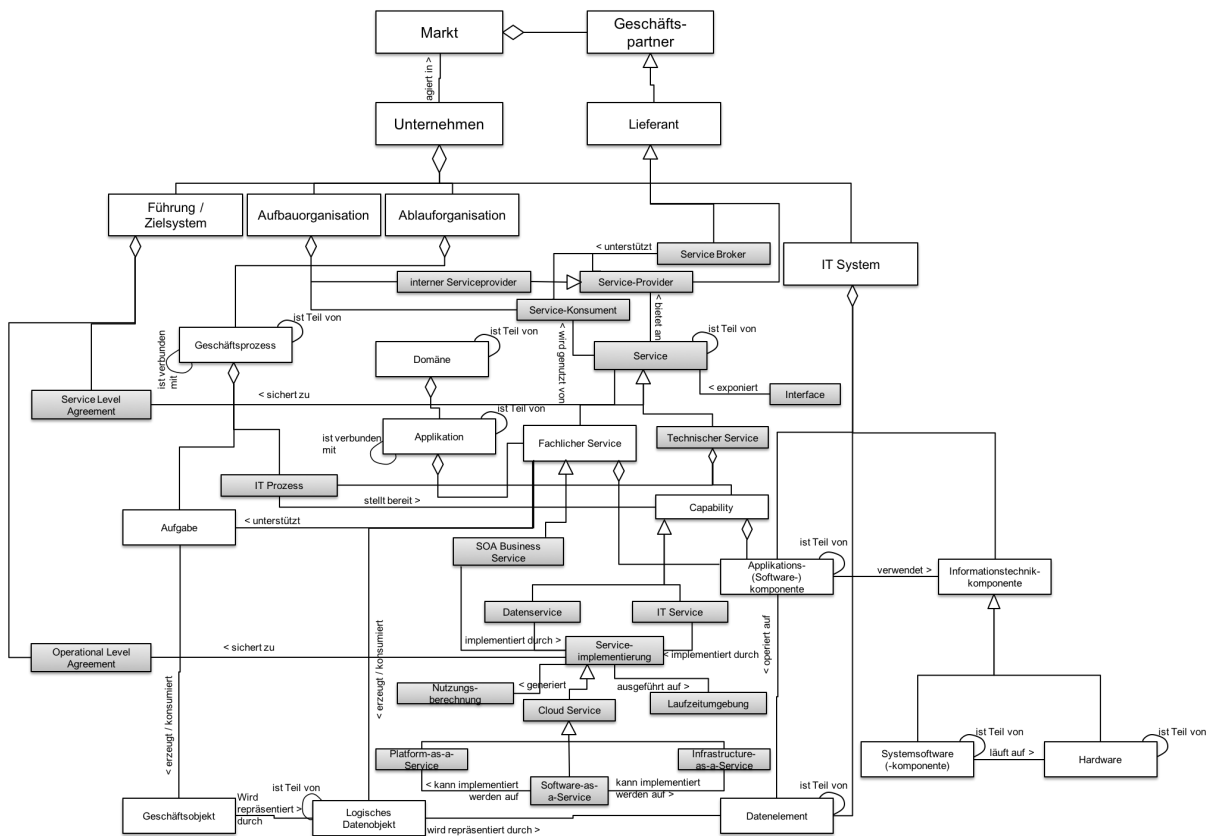


Abbildung 13: BE Business Metamodell Service Spezialisierung

5.3.5.4 Vereinfachte Darstellung des Servicebegriffes

Die Einbindung der unterschiedlichen Servicetypen in das BE Business-Metamodell ermöglicht ein ganzheitliches Verständnis der Servicebegriffe. Aufgrund der Komplexität der Darstellung des Gesamtmodells gibt Abbildung 14 eine vereinfachte Darstellung der wesentlichen Aspekte.

Das Modell zeigt, dass für alle Servicetypen zwei fundamentale Stakeholder existieren: der Servicekonsument und der -produzent. Für alle Servicetypen gilt, dass sie funktional durch Interfaces und nicht-funktional durch Service-Level-Agreements beziehungsweise Operational-Level-Agreements definiert sind. Die Prozesse, in welche die Services eingebunden sind, sind jedoch fundamental unterschiedlich. Der Konsument bindet die Services in seine Geschäftsprozesse ein, während der Produzent

auf die IT-Prozesse zur Bereitstellung der Services fokussiert ist. Dies ist auch eine wesentliche Ursache der unterschiedlichen Zielsetzungen der beiden Stakeholder. Während für die IT-Prozesse Ziele wie Effizienz durch Wiederverwendung relevant sind, steht bei den Geschäftsprozessen die Effektivität und Agilität im Vordergrund (Winter & Fischer, 2007).

In der vereinfachten Darstellung wird deutlich, wie die identifizierten Servicetypen gestaltet werden. Jene auf der linken Seite sind vorwiegend ausgehend von fachlichen Aspekten gestaltet, während diejenigen auf der rechten Seite aufgrund von Erfordernissen der IT gestaltet werden.

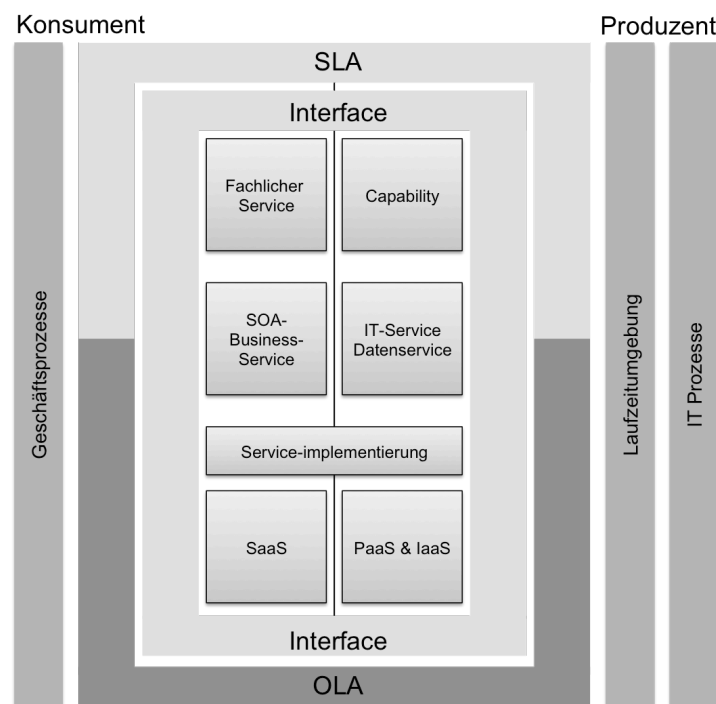


Abbildung 14: Einfache Darstellung des Servicebegriffs

5.3.5.5 Zusammenfassung

Zusammenfassend lässt sich feststellen, dass in dieser Arbeit sieben unterschiedliche Servicetypen aus der Literatur identifiziert wurden. Diese Services und ihre Attribute lassen sich direkt mit den Entitäten des BE Business-Metamodells verknüpfen und erweitern dieses um eine umfassende Servicespezialisierung.

Für alle drei Servicetypen existieren zwei Sichtweisen – eine Providersicht und eine Nutzersicht. Sie werden aber nur in sehr wenigen Arbeiten explizit adressiert. Diese Perspektiven beinhalten jeweils andere Verantwortungen und Zielstellungen. Weiterhin zeigt sich, dass alle Dienste auf der technischen Ebene des BE Business-

Metamodells durch drei wesentliche Elemente beschrieben werden sollten: funktional über eine Interfacebeschreibung und nicht-funktional über SLA und OLA.

Aufgrund der gemeinsamen essentiellen Eigenschaften von Diensten auf unterschiedlichen Ebenen lässt sich ableiten, dass auch für die auf der technischen Ebene des BE Business-Metamodells existierenden Servicetypen (z.B. SOA-Business-Service, Cloud-Service in Abbildung 13) Methoden notwendig sind, die funktionale und nicht-funktionale Anforderungen der Dienste spezifizieren, und zwar idealerweise aus Provider- wie aus Konsumentensicht.

Die Methoden im weiteren Fortgang der Arbeit befassen sich mit SOA-Business-Services sowie Cloud-Services als Spezialisierungen der Serviceimplementierungen. Somit bezieht sich der Fokus der Arbeit auf nunmehr klar definierte Servicetypen.

5.4 Zusammenfassung der Beiträge aus Teil B

Im weiteren Verlauf dieses Abschnitts werden die wesentlichen Inhalte der Beiträge aus dem Teil B dieser Dissertation dargestellt. Der jeweilige Titel und die Forschungsmethode sind tabellarisch in Tabelle 12 dargestellt.

Tabelle 12: Titel und Forschungsmethode der Beiträge

Beitrag	Titel	Forschungsmethode
Beitrag I	Entwurf eines Enterprise Architecture Frameworks für serviceorientierte Architekturen	Design Research (Artefakt-konstruktion) / Experimental
Beitrag II	Integriertes Performance Measurement als Teil des Enterprise Architecture Management	Design Research (Artefakt-konstruktion) / Testing
Beitrag III	Deriving SOA Evaluation Metrics in an Enterprise Architecture Context	Design Research (Artefakt-konstruktion) / Testing
Beitrag IV	A Flexible Approach to Service Management-Related Service Description in SOAs	Design Research (Artefakt-konstruktion) / Experimental
Beitrag V	Cloud Computing and the Impact on Enterprise IT	Case Study / Descriptive

5.4.1 Beitrag I – Entwurf eines Enterprise Architecture Frameworks für serviceorientierte Architekturen

Zielsetzung:

Beitrag I beschreibt eine Methode zur Gestaltung von Services. Ziel ist es eine Methode zu entwickeln, die in eine Unternehmensarchitektur eingebettet ist und weiterhin auf kollaborative Szenarien über Unternehmensgrenzen hinweg erweitert werden kann.

Vorgehen:

Basierend auf dem Meta-Framework für Unternehmens-Architekturmanagement Generalised Enterprise Reference Architecture and Methodology (GERAM) wird ein auf SOA adaptiertes Vorgehen entwickelt. Die GERAM bildet einen Rahmen, in dem verschiedene Modelle (Enterprise Models – EM) und Methoden (Enterprise Engineering Tools – EET) eingeordnet werden können. In Beitrag I wird somit ein EET für eine SOA innerhalb eines Unternehmens entwickelt, um damit ein EM der SOA zu erhalten. Dafür werden verschiedene Notationen (als EET) analysiert und eine Me-

thode (UN/CEFACT Modeling Methodology - UMM) verwendet, mit deren Hilfe aus unternehmensinternen als auch -übergreifenden Prozessen Dienste abgeleitet werden können.

Forschungsbeitrag:

Beitrag I adaptiert ein bestehendes Vorgehen zur Modellierung von Prozessen auf die SOA und ermöglicht damit die Servicegestaltung. Die Auswahl der GERAM als Rahmen und UMM als Methode sowie deren Erweiterung um spezifische Aspekte für die SOA stellt den zentralen Forschungsbeitrag dar.

Das Vorgehen wurde deduktiv entwickelt und durch ein Experiment evaluiert.

Publikation:

- Offermann, Philipp, Christian Schröpfer, Marten Schönherr und Maximilian Ahrens. Entwurf eines Enterprise Architecture Frameworks für serviceorientierte Architekturen - Betrachtung am Beispiel einer erweiterten UN/CEFACT Modeling Methodology. In *Wirtschaftsinformatik (1)*. 2007. S. 549-566

5.4.2 Beitrag II – Integriertes Performance Measurement als Teil des Enterprise Architecture Management**Zielsetzung:**

Beitrag II beschreibt eine Methode eines durchgängigen und konsistenten Performancemanagements im Falle einer serviceorientierten Informationsarchitekturstrategie. Ziel des Beitrages ist es, eine Methode für einen Teilbereich des Performance Management, dem Performance Measurement, zu entwickeln. Diese Methode ermöglicht die Bewertung von Organisations- und Informationsarchitektur in integrierter Weise.

Vorgehen:

Der Beitrag arbeitet mit dem Verfahren der Methodenkonstruktion. In dem Beitrag werden verschiedene vorhandene Methoden erläutert und bewertet wobei deutlich wird, dass keine von ihnen dem integrierten Anspruch gerecht wird. Jedoch decken die existierenden Vorgehensweisen Teilbereiche umfassend ab.

Aufbauend auf dieser Erkenntnis entwickelt der Beitrag eine Vorgehensweise, die verschiedene existierende Methoden integriert und dabei adaptiert.

Forschungsbeitrag:

Der Beitrag II entwickelt ein Vorgehen, welches auf unterschiedlichen Methoden basiert. Das Artefakt löst das bestehende Problem einer integrierten Performance Measurement Methode – insbesondere unter Einbeziehung des Serviceparadigmas auf technischer Ebene. Die Methode wurde deduktiv entwickelt und in einem Anwendungsfall getestet.

Publikation:

- Maximilian Ahrens, Udo Bub und Marten Schönherr. Integriertes Performance Measurement als Teil des Enterprise Architecture Managment. In *MDD, SOA und IT- Management (MSI 2007)*, Eds. Ulrike Steffens, Jan Stefan Addicks, and Nils Streekmann, 23-25. Oldenburg: Gito Verlag, 2007.

5.4.3 Beitrag III – Deriving SOA Evaluation Metrics in an Enterprise Architecture Context**Zielsetzung:**

Dieser Beitrag beschreibt eine Methode, mit deren Hilfe konkrete Metriken für die Bewertung von Services entwickelt werden können. Ziel ist, im Falle einer serviceorientierten Informationsarchitektur, einem Anwender die Möglichkeit zu geben Services im Kontext der gesamten Unternehmensarchitektur zu evaluieren.

Vorgehen:

Ähnlich wie in Beitrag II wird basierend auf einem bestehenden Verfahren (hier Goal Question Metric (GQM)) eine Erweiterung entwickelt. Mit deren Hilfe kann analysiert werden, welche Ziele mit der bestehenden Servicearchitektur in welcher Güte erreicht werden. Insbesondere wird ein Prozess zur formalen Ableitung des Messprogrammes aus der Zielhierarchie in die Methode eingefügt. Damit ist es möglich, eine Vielzahl von Zielen strukturiert und konsistent in ein Verhältnis zueinander zu bringen.

Forschungsbeitrag:

Beitrag III liefert ein Methodenartefakt, welches dem Anwender ermöglicht, ein konsistentes Messprogramm für Services integriert in eine Unternehmensarchitektur zu entwickeln. Die Methode wurde deduktiv entwickelt und im Rahmen einer Fallstudie evaluiert.

Publikation:

- Stephan Aier, Maximilian Ahrens, Matthias Stutz und Udo Bub. "Deriving SOA evaluation metrics in an enterprise architecture context". In *Service-Oriented Computing-ICSOC 2007 Workshops*. Springer Berlin Heidelberg, 2009.

5.4.4 Beitrag IV – A Flexible Approach to Service Management-Related Service Description in SOAs**Zielsetzung:**

Dieser Beitrag liefert einen Ansatz zur semantischen Beschreibung von nicht-funktionalen Serviceinformationen. Ziel des Beitrages ist es, einem Anwender die Beschreibung der nicht-funktionalen Eigenschaften von Diensten zu ermöglichen. Dies soll mithilfe eines Modells für die Beschreibung und einer prototypischen Implementierung eines Managementsystems ermöglicht werden.

Vorgehen:

In der Publikation wird die service-spezifische semantische Beschreibungssprache OWL-S (Web Ontology Language for Services) (D. Martin, Burstein, Hobbs, & Lassila, 2004) als Basis genommen und diese um die Modellierung nicht-funktionaler Eigenschaften erweitert. Diese Erweiterung wird detailliert modelliert und exemplarisch dargestellt. Weiterhin wird in dem Beitrag die Implementierung einer Softwareplattform vorgestellt, die einen Teil des Lebenszyklusmanagements von Diensten unterstützt. Insbesondere soll die Plattform die Modellierung der genannten nicht-funktionalen Parameter sowie das Auffinden von Diensten anhand von Such- und Navigationsfunktionen ermöglichen.

Forschungsbeitrag:

Die Arbeit liefert ein Metamodell als Artefakt, mit dessen Hilfe nicht-funktionale Eigenschaften von Diensten modelliert werden können. Das Artefakt wurde in Form eines Prototypen implementiert und somit durch ein Experiment verifiziert.

Publikation:

- Schröpfer, Christian, Marten Schönherr, Philipp Offermann und Maximilian Ahrens. A Flexible Approach to Service Management-Related Service Description in SOAs. In *ECOWS06 Workshop on Emerging Web Services Technology*. Zürich. 2006.

5.4.5 Beitrag V - Cloud Computing and the Impact on Enterprise IT

Zielsetzung:

Beitrag V verfolgt die Zielsetzung, die Herausforderungen der Einbindung von CC in die technische Unternehmensarchitektur anhand praktischer Beispiele aufzuzeigen. Weiterhin soll aus diesen praktischen Anwendungsfällen der allgemeine Charakter der Anforderungen von Unternehmen an CC Dienste abgeleitet werden.

Vorgehen:

Der Beitrag wendet die Methode der Fallstudienforschung an (Yin, 2003). Es werden zwei Fälle detailliert dargestellt und die spezifischen Herausforderungen dieser Fälle beschrieben. Ausgehend von den Fällen wird deduktiv abgeleitet, welche der identifizierten Anforderungen allgemeinen Charakter haben könnten.

Forschungsbeitrag:

Die Publikation zeigt an praktischen Beispielen, inwiefern funktionale und nicht-funktionale Anforderungen an CC-Dienste aus der Nutzerperspektive bestehen. Dabei zeigt sich, dass Unternehmen Anforderungen stellen, welche die untersuchten Dienste nicht erfüllen. Somit verdeutlicht der Beitrag den Bedarf an methodischer Unterstützung aus Nutzer- und Providerperspektive bei Gestaltung und Verbesserung der Dienste.

Publikation:

- Maximilian Ahrens. 2010. Cloud Computing and the Impact on Enterprise IT. In A.J. Berre et al. (Eds.): *FIS 2010*, LNCS 6369, pp. 148–155, 2010.

6 Diskussion und weiterer Forschungsbedarf

Dieses Kapitel dient der kritischen Reflektion der in dieser Dissertation beschriebenen Inhalte und Methoden sowie der Darstellung möglicher weiterer Forschungsaktivitäten, die sich anhand der Ergebnisse dieser Arbeit entwickeln lassen.

6.1 Anwendbarkeit der Forschungsergebnisse

Ein wesentliches Ziel der gestaltungsorientierten Forschung ist die Relevanz und Anwendbarkeit in der Praxis (Winter, 2008b). Die vorliegende Arbeit stellt methodische Bausteine, jedoch keine ganzheitliche Methode dar. Aus diesem Grund bedarf es für die praktische Anwendung der Artefakte ein Vorgehen zur Einordnung der zu lösenden Fragestellung in Bezug auf die methodischen Bausteine.

Sämtliche Artefakte befassen sich mit der Gestaltung und dem Management eines spezifischen Servicetyps. Es existiert in der Literatur jedoch kein gesamthaftes Modell der unterschiedlichen Servicedefinitionen (siehe Abschnitt 3.3), welches der Einordnung dienen könnte. Aus diesem Grund wurde zunächst in Abschnitt 5.3 ein umfassendes Servicemodell entwickelt.

Die Anwendung der methodischen Bausteine in der Praxis kann somit in zwei Schritten erfolgen. Zunächst sollte ein Anwender die zu lösende Problemstellung konkretisieren. Die in dieser Arbeit entwickelten Artefakte dienen der Servicegestaltung und der –verbesserung (siehe Abschnitt 2.1). Soll eine dieser beiden Themenbereiche adressiert werden, sollte weiterhin eingegrenzt werden auf welcher Ebene der Unternehmensarchitektur was für Servicetypen gestaltet werden sollen. Diese Eingrenzung kann mithilfe des in Abschnitt 5.3 entwickelten Metamodells adressiert werden. Der Anwender sollte dazu mithilfe des in Abbildung 13 dargestellten Metamodells, das Modell der eigenen Unternehmensarchitektur abgleichen und somit eingrenzen welche Servicetypen für die Problemstellung relevant sind. Handelt es sich um SOA-Business-Services oder um Cloud-Services sind - wie in Abschnitt 5.3.5.5 dargestellt - die Artefakte dieser Arbeit anwendbar.

Die Einordnung dient dabei nicht nur der Fragestellung, ob die Methodenbausteine relevant für die Problemstellung sind, sondern kann auch helfen die Ergebnisse der Methodenbausteine im Gesamtkontext der Unternehmensarchitektur zu verordnen.

6.2 Kritische Würdigung

Das für diese Arbeit formulierte Forschungsziel – beschrieben durch die FF1 und FF2 – wurde im Wesentlichen durch die präsentierten Ergebnisse erreicht, wie in Kapitel 5 dargestellt.

Die Beiträge I-IV dieser Dissertation folgen dem konstruktionswissenschaftlichen Forschungsparadigma (siehe Abschnitt 2.4). In ihnen werden unterschiedliche Artefakte erstellt. Beitrag V hingegen dient der fallstudienbasierten Validierung des Forschungsbedarfs. Für wissenschaftliche Arbeiten aus dem Bereich der Konstruktionswissenschaft existieren konkrete Vorgehensmodelle für die Konstruktion und Evaluation der im Prozess erstellten Artefakte (Peffer, Tuunanen, Rothenberger, & Chatterjee, 2007; Sonnenberg & vom Brocke, 2012). Insbesondere die Evaluation der Ergebnisse erfolgt in der vorliegenden Arbeit sehr heterogen und bedarf daher

der genaueren Betrachtung. Abbildung 15 stellt ein Vorgehen zur Evaluation in vier Schritten dar (Sonnenberg & vom Brocke, 2012).

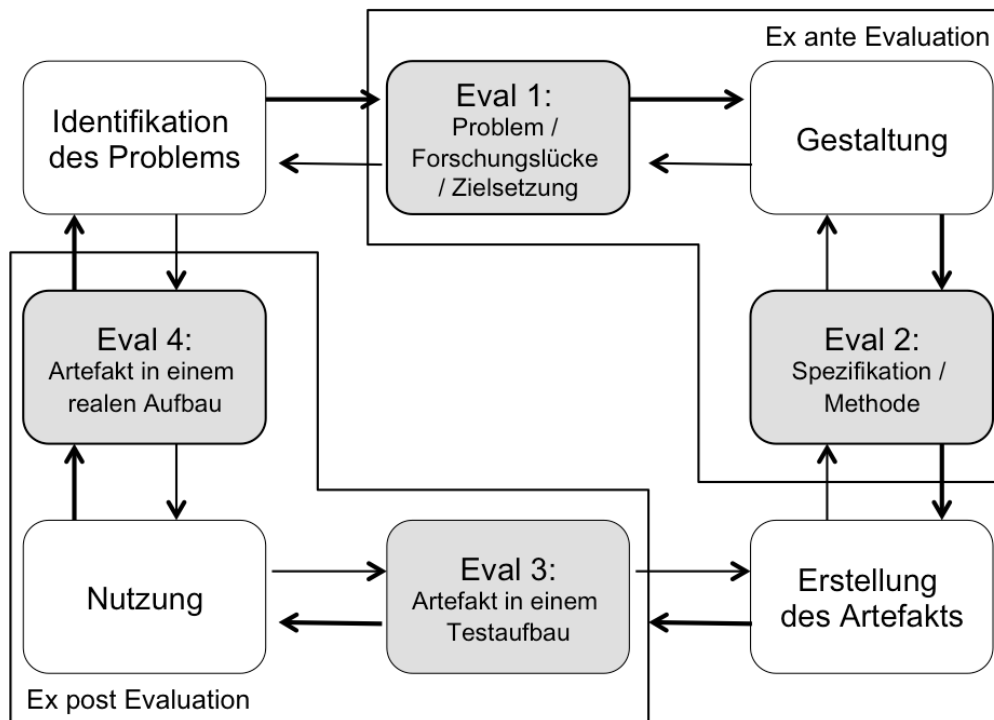


Abbildung 15: Evaluationsschritte (nach (Sonnenberg & vom Brocke, 2012))

6.2.1 Evaluation der Forschungslücke (Eval 1)

Die Evaluation der Forschungslücke kann mit Hilfe unterschiedlicher Methoden erfolgen, beispielsweise durch Literaturrecherche, Analyse von Anwendungsfällen aus der Praxis, Experteninterviews oder Befragungen (Sonnenberg & vom Brocke, 2012). Der Dachbeitrag dieser Dissertation führt zunächst Praxisbeispiele zur Darstellung der Problemrelevanz auf (vgl. Abschnitt 1.1, 1.2) und leistet dann eine umfangreiche strukturierte Literaturrecherche (Abschnitt 3) zum Nachweis des Methodendefizits. Auch die Praxisbeispiele aus Beitrag V verdeutlichen die Forschungsmotivation. Somit stellt auch dieser Beitrag einen Teil der Evaluation der Forschungslücke dar.

Zusätzlich erfolgt in den Beiträgen I-IV eine Evaluation durch eine jeweilige Literaturrecherche. Insgesamt erscheinen die Evaluation des Problems und der Nachweis des Forschungsbedarfes ausreichend geleistet.

6.2.2 Evaluation des Gestaltungsentwurfes (Eval 2)

Ziel der Evaluation des Gestaltungsentwurfes ist es, vor der Instanziierung des Artefakts das Design zu validieren – dies kann durch eine mathematisch-logische Kon-

trolle erfolgen oder durch stärker qualitativ orientierte Verfahren wie Demonstration, Simulation, Befragung oder Interviews (Sonnenberg & vom Brocke, 2012).

Diese Form der Evaluation ist hinsichtlich der Forschungsfrage 1.1 insbesondere für die Beiträge I und IV relevant. In Beitrag I wird die Methode zur Gestaltung von Diensten in Form einer Simulation getestet, während das Modell zur Modellierung nicht-funktionaler Eigenschaften aus Beitrag IV durch Beispiele simuliert wird.

Die Artefakte zur Beantwortung der Forschungsfrage 2.1 sind in den Beiträgen II und III dargestellt. Beide Artefakte sind Kombinationen von bereits existierenden und validierten Methoden. Daher wurden in den Beiträgen lediglich Ergänzungen logisch nachvollzogen.

Die Forschungsfragen 1.2 und 2.2 werden jeweils teilweise durch den Dachbeitrag und die Beiträge I und IV, beziehungsweise III erläutert. Da die Evaluation der Artefakte zwar in den Beiträgen erfolgt, aber diese nicht ausreichend explizit auf die Nutzerperspektive eingehen, wurde eine deduktive Ableitung der Validität der Artefakte bezüglich der FF 1.2 und 2.2 im Dachbeitrag durchgeführt (siehe Abschnitte 5.1, 5.2).

6.2.3 Evaluation des Artefakts in einem Testaufbau (Eval 3)

Die dritte Phase der Evaluation erfolgt nach der Instanziierung des Artefakts. In dieser Phase soll eine erste Demonstration des Artefakts erreicht werden, was mit Hilfe von prototypischen Implementierungen, Experimentaufbauten oder auch wiederum durch Experteninterviews oder Befragungen erreicht werden kann (Sonnenberg & vom Brocke, 2012). Diese Phase der Evaluation wurde lediglich für die FF 1.1 und 2.1 erreicht. Die Methoden- und Modellartefakte wurden durch prototypische Testimplementierungen (FF1.1 – Beiträge I und IV) beziehungsweise eine Implementierung in einem begrenzten realen Anwendungsfall (FF 2.1 – Beiträge II und III) evaluiert.

Die FF 1.2 und 2.2 hingegen wurden aufgrund der im vorigen Abschnitt dargestellten Problematik keiner direkten Evaluation des Artefakts aus Nutzersicht unterzogen.

6.2.4 Evaluation des Artefakts in einer realen Implementierung (Eval 4)

In dieser Phase der Evaluation soll das Artefakt in einer realen und produktiven Umgebung eingesetzt werden, was durch Fallstudien, Feldstudien und Interviews dargestellt werden kann.

Keines der in dieser Arbeit erstellten Artefakte wurde vollumfänglich in einer realen Umgebung implementiert. Die in Beitrag III vorgestellte Methode wurde zwar in einer realen Umgebung getestet und somit durch ein praktisches Experiment evaluiert, jedoch erfolgte keine nachhaltige Implementierung der Methode in dem Unternehmen, so dass nach dem Experimentdurchlauf keine Anwendungen der Methode stattfanden.

Diese weiteren Evaluationen sollten Bestandteil weiterer Forschung sein (siehe Abschnitt 6.3).

6.2.5 Zusammenfassende kritische Würdigung

Durch die dargestellten Massnahmen zur Evaluation der Artefakte ist die Validierung der Artefakte teilweise erfolgt. Eine Einschränkung der Evaluation der Ergebnisse dieser Arbeit liegt insbesondere darin, dass die FF 1.2 und 2.2 nicht auf identische Art und Weise wie FF 1.1 und 2.1 durch direkte Evaluation erfolgen konnte, sondern lediglich durch Deduktion in diesem Dachbeitrag. Weiterhin wurde die Evaluation durch eine reale Praxisimplementierung nicht umfassend, sondern nur einzelne Beiträge betreffend durchgeführt.

Zusätzlich ist zu beachten, dass die Ergebnisse aus Teil B der Arbeit über einen längeren Zeitraum in unterschiedlichen Kontexten generiert wurden. Daher ist eine begriffliche und methodische Konsistenz nicht gewährleistet, was ein einfaches Gesamtverständnis der Zusammenhänge der einzelnen Beiträge erheblich erschwert. Der Dachbeitrag adressiert diese Problematik, indem ein umfassendes Metamodell der unterschiedlichen Servicetypen entwickelt wird, um so eine Einordnung der Beiträge zu ermöglichen. Dennoch bleibt für den Leser der vorliegenden Arbeit die Herausforderung bestehen, die Inhalte des Dachbeitrages mit denen der jeweiligen Einzelbeiträge in Beziehung zu setzen.

6.3 Weiterer Forschungsbedarf

Bereits aus den im vorigen Abschnitt dargestellten Inhalten ergibt sich zusätzlicher Forschungsbedarf.

6.3.1 Methodenartefakte aus der Nutzersicht darstellen und dezidiert evaluieren

Die hier dargelegten Methoden und Modelle für die Servicegestaltung und die Verbesserung der Dienste stellen nur implizit die Nutzersicht dar. Nur mithilfe des Dachbeitrages werden die spezifischen Nutzeraspekte deutlich. Aus diesem Grund sollten

sich weitere Arbeiten mit der Darstellung eines Artefakts dezidiert aus der Nutzersicht befassen und dieses Artefakt auch in allen Phasen evaluieren (Eval 1-4).

6.3.2 Evaluation sämtlicher Artefakte in einer realen Implementierung (Eval 4)

Sämtliche Artefakte müssen in einer realen Implementierung evaluiert werden. Da dies noch nicht erfolgt ist, sollte es in einem Anwendungsfall mit dedizierte Nutzer- und Providerperspektive geschehen.

6.3.3 Integration in ein etabliertes Framework

Die Methoden und Modelle dieser Arbeit adressieren sehr spezifische Aspekte, die jedoch in der Praxis in einem ganzheitlichen Kontext implementiert werden. Diese Dissertation leistet eine solche Einbindung bezüglich des Modellaspektes (durch die Einbindung in das BE Business-Metamodell), nicht jedoch bezüglich eines ganzheitlichen Vorgehensmodells. Dieser Aspekt kann in einer weiteren Arbeit adressiert werden. Im Rahmen dieses Dachbeitrages ist eine grundlegende Einordnung in das ITIL Framework erfolgt (siehe Kapitel 2.2). Eine Einordnung dieser Art sollte jedoch weiter fundiert werden, zum Beispiel durch strukturierte Auswahl eines geeigneten Frameworks. Weiterhin sollte ein so entstandenes Artefakt auch vollständig evaluiert werden.

Teil B – Einzelbeiträge

1 Beitrag I: Entwurf eines Enterprise Architecture Frameworks für Serviceorientierte Architekturen

1.1 Abstract

Damit serviceorientierte Architekturen (SOAs) ihr Potential voll entfalten können, müssen neben technischen auch methodische und organisatorische Aspekte betrachtet werden. Enterprise Architecture Frameworks (EAFs) ermöglichen die dazu notwendige ganzheitliche Sicht auf das Unternehmen. Einzelne EAFs sind meist zu spezialisiert, um als Framework für die SOA angewendet werden zu können. Das Metaframework Generalised Enterprise Reference Architecture and Methodology (GERAM) kann jedoch genutzt werden, um die für eine SOA notwendigen Elemente zu identifizieren. Zusätzlich haben wir nachgewiesen, dass das Framework UN/CEFACT Modeling Methodology (UMM) für die Integration von innerbetrieblichen Prozessen geeignet ist. Daher wird die UMM unter Verwendung der GERAM in Hinblick auf die SOA analysiert, um Schwachstellen aufzudecken. Darauf aufbauend werden Erweiterungen der UMM für die SOA vorgeschlagen.

1.2 Einleitung

Während Technologien für serviceorientierte Architekturen (SOA) bereits standardisiert sind und eine gewisse Verbreitung gefunden haben, befinden sich Methoden zur Einführung einer SOA noch im Entwicklungsstadium.

1.2.1 Ziel des Artikels

Der Artikel hat zum Ziel, einen Beitrag zu methodischen Grundlagen für den Entwurf von SOAs zu leisten. Zur Einführung einer SOA gehört eine ganzheitliche Sicht auf das Unternehmen, wie sie gewöhnlich von Enterprise Architecture Frameworks (EAFs) eingenommen wird. Jedoch haben einzelne EAFs jeweils ihren eigenen Kontext und sind meist zu spezialisiert, um ohne weiteres für das Architekturkonzept der SOA eingesetzt werden zu können.

Die Generalised Enterprise Reference Architecture and Methodology (GERAM) stellt ein Metaframework für EAFs da, welches zum Vergleich von existierenden und zur

Entwicklung von neuen Frameworks als Referenz benutzt werden kann. Ausgehend von GERAM wird in diesem Artikel die UN/CEFACT Modeling Methodology (UMM) daraufhin untersucht, inwieweit sie sich für den Entwurf einer SOA eignet. Die Untersuchung der UMM unter Bezugnahme auf GERAM soll aufzeigen, welche Probleme sich in Hinblick auf SOAs ergeben, welche zusätzlichen Artefakte benötigt werden und wie die UMM erweitert werden kann, um neben der Geschäftsprozessintegration auch SOAs zu unterstützen.

1.2.2 Methode

Als wissenschaftliche Methode wird die Deduktion verwendet (Braun u. a., 2004). Ausgehend vom Metaframework GERAM wird abgeleitet, welche Anforderungen an ein Framework für SOAs zu stellen sind. Diese abgeleiteten Anforderungen werden dann mit dem existierenden Framework UMM verglichen, um daraus Erkenntnisse über die Anwendbarkeit der UMM für SOAs und mögliche Erweiterungen abzuleiten. Für die UMM wurde an unserem Fachgebiet bereits unter Anwendung der Methode der Aktionsforschung (Braun u. a., 2004) nachgewiesen, dass sie für die Integration unternehmensinterner Geschäftsprozesse anwendbar ist (Dietrich, 2008).

1.2.3 Gliederung

Zunächst werden EAFs eingeführt, um dann das Metaframework GERAM vorzustellen. Zusätzlich wird das Konzept der SOA vorgestellt und eingeordnet, welche Anforderungen an ein EAF für die SOA sich durch GERAM ergeben. Danach wird das konkrete Framework UMM vorgestellt und aufgezeigt, welche Erweiterungen für die innerbetriebliche Service-Orchestrierung notwendig sind. Schließlich wird die UMM in GERAM eingeordnet.

1.3 Enterprise Architecture Frameworks

Als erstes werden Enterprise Architecture Frameworks (EAFs) und die GERAM vorgestellt.

1.3.1 Das Ziel von Enterprise Architecture Frameworks

Unternehmen sind komplexe Gebilde, in denen Menschen in formalen und informellen Strukturen zusammenarbeiten, um ihre jeweiligen Ziele zu erreichen. Heutzutage erfolgt die Unterstützung dieser Strukturen vielfach durch EDV. EAFs helfen dabei, einen Überblick über das komplexe Zusammenspiel von Menschen, Organisations-

struktur und EDV zu erlangen. Für Schallert & Rosemann (2003) ist die Beherrschbarkeit von Komplexität und Management von Unternehmensintegration das oberste Ziel einer Architekturbetrachtung (Schallert & Rosemann, 2003). Die International Organization for Standardization (ISO) definiert die Ziele einer Enterprise Architecture allgemein mit der Befähigung eines Teams, umfassend alle Ressourcen eines Unternehmens zu integrieren (ISO, 2000).

1.3.2 Existierende Frameworks

Seit den 80er Jahren wurden viele EAFs entwickelt. Eine Übersicht findet sich in (Scheckermann, 2006) und (Schönherr, 2006). Zu den bekanntesten Frameworks zählen CIMOSA, GRAI-GIM, PERA, Zachman, ARIS und DoDAF. Jedes dieser Frameworks hat auf Grund seiner Entstehungsgeschichte einen anderen Fokus.

Angesichts dieser Vielzahl von EAFs hat die *International Organization for Standardization* (ISO) einen Standard publiziert, in dem Anforderungen an EAFs definiert sind (ISO, 2000). Im Annex des ISO-Standards findet sich die *Generalised Enterprise Reference Architecture and Methodology* (GERAM) der IFAC/IFIP (IFAC/IFIP, 1999), ein generalisiertes Framework, um konkrete Frameworks vergleichen und bewerten zu können (Bernus & Nemes, 1997).

1.3.3 GERAM

GERAM besteht aus neun Elementen. Diese sind in Abbildung 16 dargestellt.

Das zentrale Element der GERAM ist die *Generalised Enterprise Reference Architecture* (GERA). In ihr werden allgemeine Konzepte zur Beschreibung eines Unternehmens empfohlen. Die Konzepte werden in drei Klassen unterteilt: mitarbeiterorientierte, prozessorientierte und technologieorientierte Konzepte. Die prozessorientierten Konzepte beinhalten ein Lebenszykluskonzept, welches acht Phasen unterscheidet. Die Phasen sind in Abbildung 17 zu sehen. GERA unterscheidet fünf Entity-Typen, welche Lebenszyklen besitzen können (IFAC/IFIP, 1999):

- Strategic Enterprise Management Entity,
- Enterprise Engineering/Integration Entity,
- Enterprise Entity,
- Product Entity und
- Methodology Entity.

für Serviceorientierte Architekturen

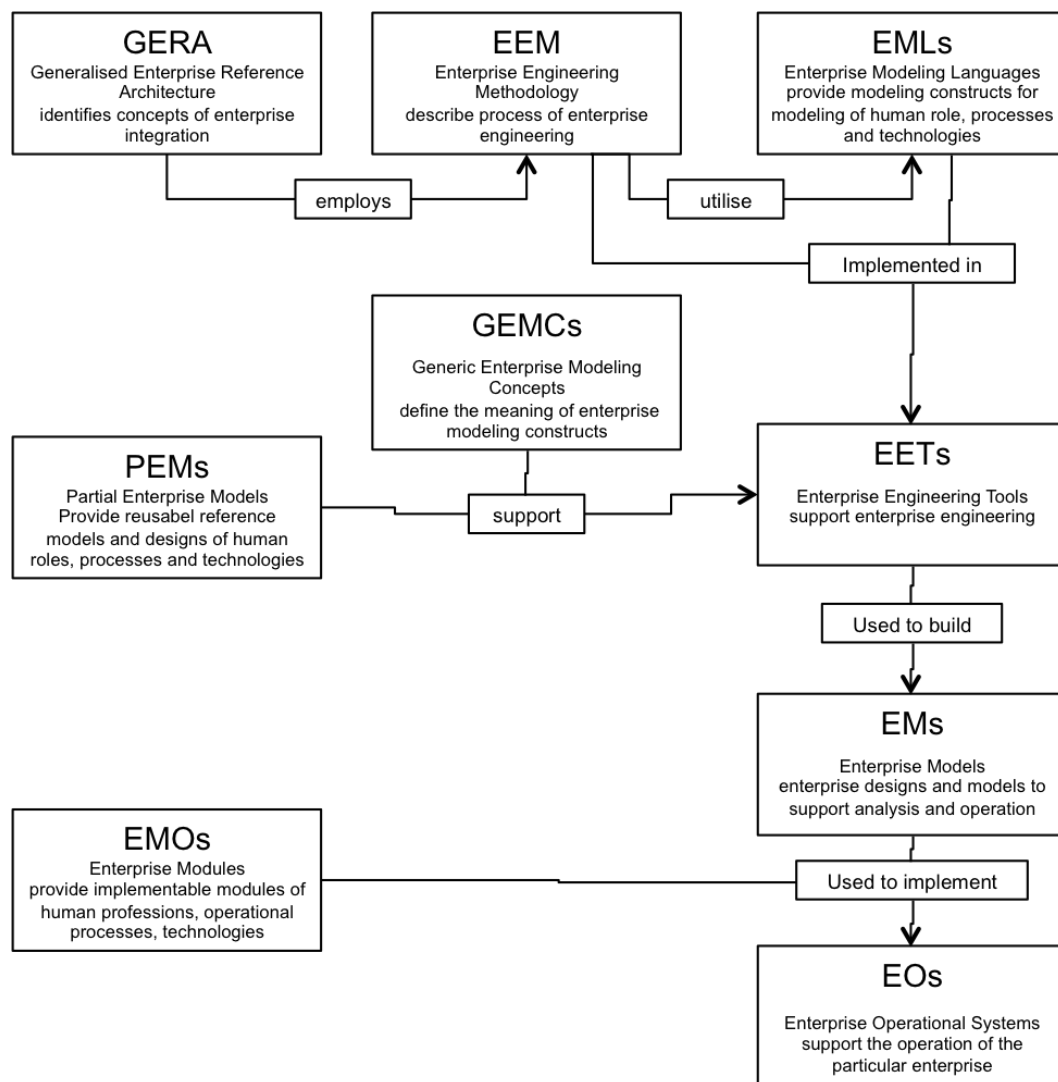


Abbildung 16: GERAM Elemente

Die *Enterprise Engineering Methodology* (EEM) verwendet die abstrakten Konzepte der GERA und beschreibt allgemeine Methoden, die zur Erstellung von konkreten Unternehmensmodellen und zum Management von Unternehmenseinheiten entlang ihrer Lebenszyklen verwendet werden können.

Die Unternehmensmodelle, die durch Anwendung der EEM erstellt werden, müssen festgehalten werden. Hierzu sieht GERAM *Enterprise Modeling Languages* (EMLs) vor. Um die verschiedenen Aspekte eines Unternehmens modellieren zu können, sollte es mehrere unterschiedliche Modellierungssprachen geben. Durch die Integration der verschiedenen Sprachen wird es möglich, ein durchgängiges Unternehmensbild zu schaffen (IFAC/IFIP, 1999).

Zum Erstellen und Verwalten von Unternehmensmodellen gibt es *Enterprise Engineering Tools* (EETs). Diese ermöglichen die Anwendung der EMLs und unterstützen damit die EEM. Weitere Anforderungen an EETs sind u. a. die Unterstützung von Kollaboration und ein Speicher für wiederverwendbare Modellteile.

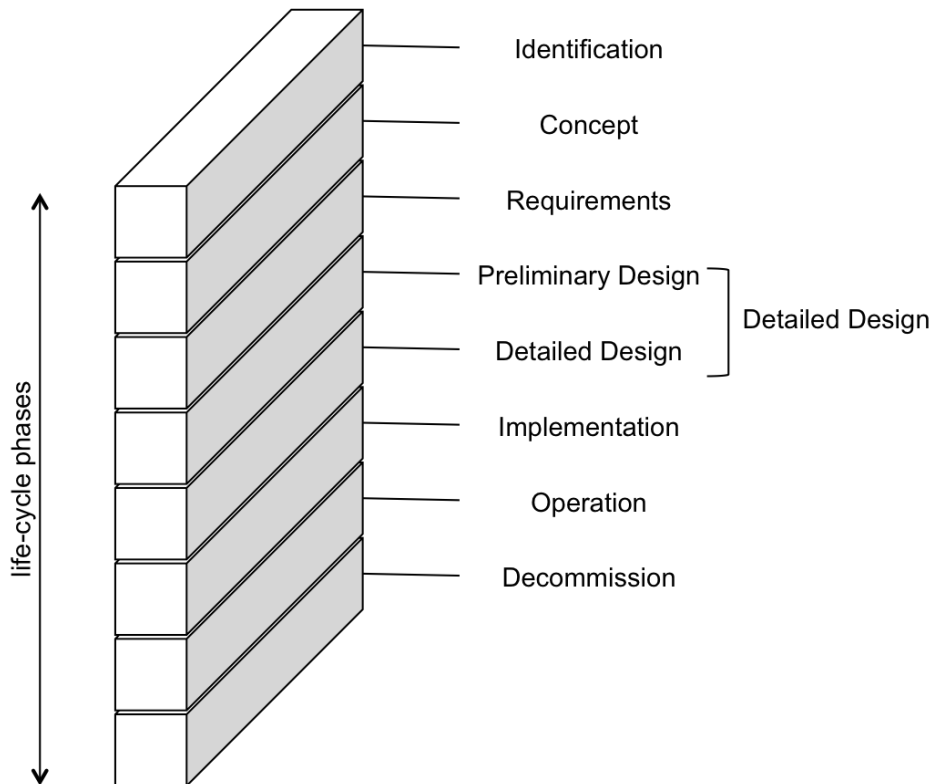


Abbildung 17: GERA Lebenszyklusphasen

Zur Unterstützung der Modellierung mit einem EET führt GERAM *Partial Enterprise Models* (PEMs) ein. Dabei handelt es sich um Referenzmodelle, die in vielen Unternehmen angewendet werden können. Der Vorteil der Benutzung von Referenzmodellen ist, dass die Modellierung beschleunigt wird und unter Umständen die Referenzmodelle effizienter sind als die tatsächlich im Unternehmen eingesetzten Strukturen.

Für die Modellierung können weiterhin *Generic Enterprise Modeling Concepts* (GEMCs) eingesetzt werden. Hierbei handelt es sich um Glossare, Metamodelle oder Ontologien, welche die Bedeutung der verwendeten Begriffe definieren. Nur durch diese einheitliche Festlegung von Begriffen ist es möglich, konsistente Modelle aufzubauen.

Die EETs werden eingesetzt, um *Enterprise Models* (EMs) zu erstellen. Diese Modelle können als Entscheidungshilfe, gemeinsame Verständigungsgrundlage und Trai-

ningsmaterial sowie zum Betrieb und zur Kontrolle verwendet werden. Auf Grundlage der EMs kann schließlich die EDV, die *Enterprise Operational Systems* (EOS), implementiert werden. Zur Implementierung können *Enterprise Modules* (EMOs) eingesetzt werden, welche im Unternehmen oder am Markt verfügbare Module zur Umsetzung der EOS sind. Generell sind EMOs Implementierungen von PEMs. Wenn ein Teilunternehmensmodell (PEM) in einem Unternehmensmodell (EM) verwendet wird, kann das entsprechende Unternehmensmodul (EMO) für die Umsetzung des Unternehmensmodells (EMs) in den Unternehmenssystemen (EOS) benutzt werden.

1.3.4 GERAM und serviceorientierte Architekturen

Mit seinen neun Elementen spannt GERAM einen allgemeinen Rahmen auf, der von konkreten EAF gefüllt wird. Ein konkretes EAF fokussiert sich dabei oft auf bestimmte Aspekte eines Unternehmens, um diese beherrschbar zu machen und ggf. optimieren zu können. Ein solcher Aspekt sind Organisations- und EDV-Strukturen, die das Konzept der serviceorientierten Architektur (SOA) umsetzen.

Es wird zunächst das Konzept der SOA erläutert, um dann an einem konkreten Beispiel aufzuzeigen, welche Anforderungen an ein EAF für SOA im von GERAM aufgespannten Rahmen gestellt werden.

1.4 Serviceorientierte Architekturen

SOA sind ein aktueller Trend in der EDV-Industrie (Gomes, 2006). Viele große Firmen wie IBM, Microsoft, BEA und SAP unterstützen Standards im Bereich von SOA oder stellen ihre Produkte auf eine SOA um (BEA Systems, 2006; IBM Corporation, 2006a; Microsoft Corporation, 2006; SAP AG, 2006). Auch Organisationen wie das World Wide Web Consortium (W3C), OASIS und die Object Management Group (OMG) entwickeln und publizieren Standards im Zusammenhang mit SOA (OASIS, 2006; Object Management Group, 2006; World Wide Web Consortium, 2004).

1.4.1 Ziel von serviceorientierten Architekturen

Die Gartner Group definiert eine SOA wie folgt: „SOA is a software architecture that builds a topology of interfaces, interface implementations and interface calls. SOA is a relationship of services and service consumers, both software modules large enough to represent a complete business function. So, SOA is about reuse, encapsulation, interfaces, and ultimately, agility.“ (McCoy & Natis, 2003)

Die am häufigsten angewendete Technologie zur Implementierung einer SOA sind Webservices. Ein Webservice ist ein Programm, welches seine Funktionen über definierte Schnittstellen und offene Protokolle anbietet (World Wide Web Consortium, 2004). Der Unterschied zur klassischen Modularisierung von Programmlogik ist, dass die Funktionalität, welche von den Webservices angeboten wird, aus den Aktivitäten des Geschäftsprozesses und nicht aus den EDV-Systemen abgeleitet wird (Foody, 2005; Schmelzer, 2005; Szyperski, Bosch, & Weck, 1999).

Aus diesem Grund beschränkt sich das Konzept der SOA nicht auf die technische Seite, sondern reicht bis in das Geschäftsprozessmanagement hinein (Leymann, Roller, & Schmidt, 2002). Indem Webservices so entwickelt werden, dass sie Geschäftsprozessfunktionalität abbilden, ist es möglich, die Struktur der Geschäftsprozesse in Übereinstimmung mit der die Geschäftsprozesse unterstützenden EDV zu bringen.

Um zu ermöglichen, die EDV-Struktur an den Geschäftsprozessen auszurichten, ist es notwendig, die Reihenfolge, in der die Webservices aufgerufen werden, zu beschreiben. Bei dieser Beschreibung muss sichergestellt werden, dass die Reihenfolge der Serviceaufrufe strukturell gleich ist zur Reihenfolge der Aktivitäten im Geschäftsprozess. Eine solche ausführbare Beschreibung heißt Orchestrierung, wenn die Aufrufe zentral koordiniert werden, und Choreographie, wenn die Koordination dezentral erfolgt (Newcomer & Lomow, 2004).

1.4.2 Anforderungen einer SOA an Elemente der GERAM

Um sowohl die Organisationsstruktur mit den Geschäftsprozessen als auch die EDV mit den Webservices ganzheitlich modellieren zu können, eignen sich EAF sehr gut. Eine SOA stellt jedoch bestimmte Anforderungen an ein EAF, die am Rahmen von GERAM erläutert werden.

Mit dem Ziel, die EOS auf Basis von Services zu bauen, können Implementierungen von einzelnen Webservices als EMOs angesehen werden. Das EAF sollte also als EMOs Webservices vorsehen. Die Webservices können, da sie Geschäftsprozessaktivitäten unterstützen, auf abstrakter Ebene auch als PEMs betrachtet werden.

Webservices sollten strukturell in der gleichen Reihenfolge aufgerufen werden, in der auch die Geschäftsprozessaktivitäten ausgeführt werden. Deshalb müssen bei den EMs insbesondere die Geschäftsprozessmodelle beachtet werden. Diese sollten es

ermöglichen, unter Verwendung der PEMs und ihrer Implementierung in EMOs automatisch eine Orchestrierung oder Choreographie von Webservices zu erzeugen.

Die EETs sollten es entsprechend unterstützen, die Geschäftsprozessmodelle so zu beschreiben, dass die Prozessmodelle in eine ausführbare Orchestrierung oder Choreographie übersetzt und Webservices an die Aktivitäten gebunden werden können. Hierfür müssen wie bereits erwähnt die PEMs Webservices und wenn möglich bereits vororchestrierte Prozesse enthalten. Weiterhin sollten, da für eine Bindung der Services an Aktivitäten eine semantische Beschreibung der funktionalen und nicht-funktionalen Eigenschaften der Aktivitäten sowie der Services nötig ist, die GEMCs eine solche Beschreibung unterstützen.

Für die EMLs ist neben den erwähnten Anforderungen an die Geschäftsprozessmodelle insbesondere vorzusehen, dass Rollen im Unternehmen für das Servicemanagement modelliert werden können. Entsprechend müssen in GERA die Konzepte ausgebaut werden, um die Modellierung des Servicemanagements zu ermöglichen.

Wir haben die UN/CEFACT Modeling Methodology (UMM) erweitert und werden an diesem Beispiel zeigen, wie eine konkrete Ausprägung der einzelnen Elemente für SOAs aussehen kann. Hierbei wird die GERAM als Rahmen verwendet, um die Elemente der UMM einordnen und damit die Vollständigkeit des Frameworks bewerten zu können.

1.5 UN/CEFACT Modeling Methodology für SOA

Um das Ziel der SOA, das heißt die Anpassung der EDV-Strukturen an die Geschäftsprozesse, zu erreichen, müssen die Prozesse aufgenommen und aus ihnen die Services abgeleitet werden. Die UN/CEFACT Modeling Methodology (UMM) stellt eine Methode und Modelle bereit, um Prozesse aufzunehmen, die für die Prozesse notwendige Kollaboration zwischen verschiedenen Geschäftspartnern zu identifizieren und eine Choreographie für Services zwischen den Geschäftspartnern abzuleiten (UN / CEFACT, 2006b). Der Ursprung der UMM ist die Koordination von kollaborativen Geschäftsprozessen. Wie erwähnt wurde aber bereits nachgewiesen, dass sich die UMM auch zur Integration von unternehmensinternen Geschäftsprozessen eignet (Dietrich, 2008).

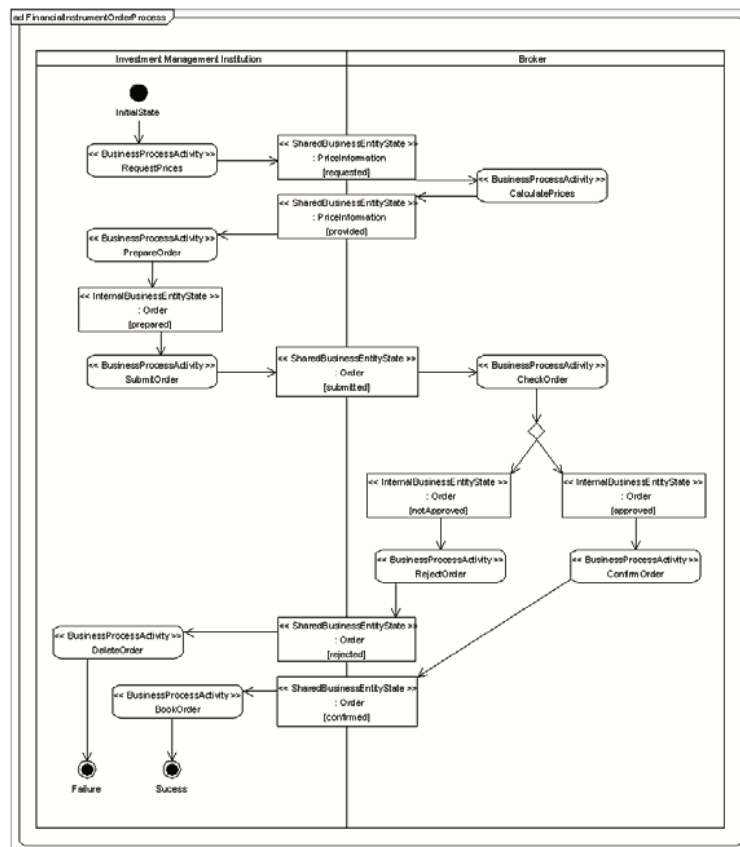


Abbildung 18: Beispiel des UMM Business-Process-View

1.5.1 Die Methodologie

Die UMM basiert auf drei Sichten: Business Domain View, Business Requirements View und Business Transaction View. Die *Business Domain View* wird verwendet, um Geschäfts- und Prozessfelder zu identifizieren. Für jedes Feld werden dann Geschäftsprozesse und in diese involvierte Interessenten modelliert. Zum Einsatz kommen hierbei auf UML Anwendungsfalldiagrammen basierende Diagramme. Im *Business Requirements View* werden im nächsten Schritt die im Business Domain View identifizierte Geschäftsprozesse modelliert. Dazu definiert die UMM den *Business Process View*, der auf UML Aktivitätsdiagrammen basiert. Ein Beispiel ist in Abbildung 18 zu sehen. Für die im Business Process View verwendeten Business Entities werden zusätzlich im *Business Entity View* Lebenszyklen festgelegt. Aus dem Business Process View kann der Bedarf an Kollaboration zwischen Unternehmen abgelesen werden. Dieser Bedarf wird im dritten und letzten View des Business Domain Views, dem *Partnership Requirements View*, genauer modelliert. Nachdem der Busi-

ness Requirements View modelliert worden ist, ist der *Business Transaction View* der letzte Schritt der Methodologie. Er besteht aus drei Sichten: Business Choreography View, Business Interaction View und Business Information View. Im *Business Choreography View* wird der Ablauf von für eine Kollaboration zwischen Geschäftspartnern nötigen Aktivitäten festgehalten. Einzelne Kollaborationen werden dann im *Business Interaction View* modelliert. Die Informationseinheiten, die bei einer Kollaboration ausgetauscht werden, werden im *Business Information View* spezifiziert.

Vorteilhaft ist es, wenn die Informationseinheiten, die im Business Interaction View ausgetauscht werden, einheitlich verwendet und semantisch beschrieben werden. Hierzu können Core Components wie auch UMM, ein UN/CEFACT Standard, verwendet werden. Konkret können Business Information Entities, die von Core Components abgeleitet sind, die in UMM verwendeten Information Entities realisieren (UN / CEFACT, 2006a).

Auf Basis der Business Interaction View und der Business Choreographie View in Verbindung mit den durch Core Components unterstützten Information Entities ist es möglich, eine Choreographie zu erstellen. Mit Hilfe des UMM Plug-Ins (Research Studio Austria, 2006) für den Sparx System Enterprise Architect (Sparx Systems Pty Ltd, 2006) ist es zum Beispiel möglich, aus den Modellen eine Choreographiebeschreibung auf Basis von BPSS (UN / CEFACT, 2003) zu erzeugen.

1.5.2 Kritik

Auf Grund ihres Ursprungs ist die UMM für die Integration kollaborativer Geschäftsprozesse gut geeignet. Wie dargestellt ist es sogar möglich, bei entsprechender Werkzeugunterstützung ausführbare Orchestrierungen zu generieren. Schwachstellen ergeben sich jedoch, wenn innerbetriebliche Prozesse betrachtet werden sollen. Diese sind entscheidend, um in einer SOA eine Prozessorchestrierung generieren zu können. Außerdem sind weiterreichende Beschreibungen der Prozessaktivitäten wünschenswert, um aus diesen Anforderungen an Services ableiten zu können. Lösungsvorschläge für diese Schwachstellen haben wir in Form von Erweiterungen der UMM vorgeschlagen.

1.5.3 Erweiterungen der UMM

Wir haben drei Erweiterungen der UMM vorgeschlagen, um neben der Erstellung von Choreographien auch die Erstellung von Orchestrierungen möglich zu machen. Der

Vorteil hiervon ist, dass die Umsetzung einer SOA sowohl im Unternehmen als auch zwischen Unternehmen mit einer einzigen Methodologie und einem kohärenten Satz von Modellen möglich ist.

Die erste Erweiterung bezieht sich auf die Verwendung des Business Process View. Wir schlagen vor, die Aktivitäten auf der ersten Ebene nur mit jeweils einer Aktivität pro Geschäftspartner zu modellieren. Erst in den Verfeinerungen sollen die Details der internen Geschäftsprozesse modelliert werden. Hierdurch lässt sich aus der ersten Ebene der Bedarf an Kollaborationen deutlicher ablesen, während weitere Ebenen zur Erstellung von innerbetrieblichen Orchestrierungen verwendet werden können. Ein Beispiel hierfür ist in Abbildung 19 zu sehen.

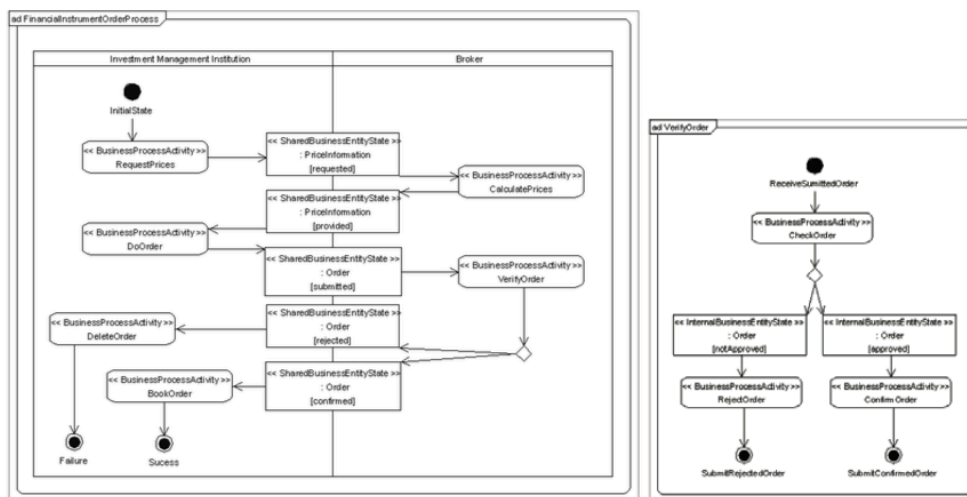


Abbildung 19: Geschäftsprozessmodell mit Verfeinerung

Die zweite Erweiterung bezieht sich auf die semantische Hinterlegung der Business Entities, die im Business Process View benutzt werden. Wie beschrieben können die Information Entities, die im Business Transaction View benutzt werden, durch die Core Components mit einer einheitlichen Semantik hinterlegt werden. Für die Business Entities im Business Process View ist dies nicht vorgesehen. Für eine Anbindung von Aktivitäten an Services ist jedoch hauptsächlich der Informationsfluss interessant. Wir schlagen daher vor, eine Verbindung zwischen Information Entities und Business Entities zu schaffen. Da Information Entities an Business Information Entities gebunden werden können, wären damit auch die Business Entities semantisch hinterlegt und der Informationsfluss in der Business Process View semantisch beschrieben.

Als dritte Erweiterung schlagen wir vor, die Diagramme der Business Process View durch Diagramme in der Business Process Modeling Notation (BPMN) (Object Management Group, 2006) zu ersetzen. Ein Beispiel hierfür ist in Abbildung 20 zu sehen. Diese Variante hat folgende Vorteile:

- Während viele Konstrukte in BPMN Äquivalente in UML Aktivitätsdiagrammen haben, ist die BPMN, insbesondere was die Fehlerbehandlung und die Transaktionsverwaltung angeht, expliziter.
- BPMN wurde entwickelt, um Geschäftsprozessmodelle in die ausführbare Orchestrierungssprache BPEL (Andrews, Curbera, Dholakia, & Goland, 2003) übersetzen zu können. Entsprechend gibt es eine im BPMN-Standard definierte Abbildung von BPMN auf BPEL.
- Während UML Aktivitätsdiagramme der OMG-Standard für die Modellierung von objektorientierter Software sind, ist die BPMN der OMG-Standard für die Geschäftsprozessmodellierung.
- Durch die speziellen Pfeile für den Nachrichtenfluss und die Markierung von Gruppen können notwendige Kollaborationen leichter identifiziert werden als im auf UML basierenden Business Process View.

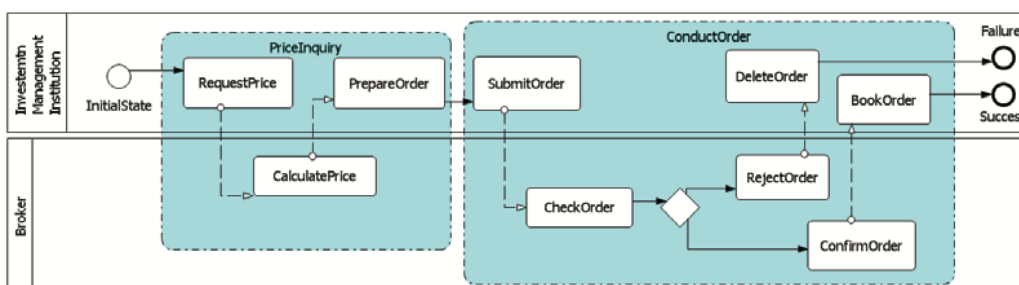


Abbildung 20: Geschäftsprozessmodell in der BPMN

1.5.4 Einordnung der Erweiterung in GERAM

Es stellt sich die Frage, wie die erweiterte UMM die für eine SOA notwendigen Anforderungen der Elemente von GERAM umsetzt. Für die EMOs ist es erforderlich, dass Webservices als Module verwaltet werden können. Hierfür wird ein Webservice-Verzeichnis benötigt, das die Webservices für die Benutzung in den EOS bereitstellt. Während ein solches Verzeichnis nicht Teil der beschriebenen Methodologie ist, gibt es Ansätze wie den UDDI-Standard (Newcomer, 2002), solche Verzeichnisse zur Verfügung zu stellen. Auch Verzeichnisse mit semantischen Erweiterungen wur-

den z. B. in (Aggarwal & Verma, 2004) und (Grønmo, Jaeger, & Hoff, 2005) bereits vorgeschlagen.

EMLs sollten insbesondere Geschäftsprozessmodelle vorsehen, die für eine automatische Orchestrierung und Choreographie verwendet werden können. Während mit der UMM die Erstellung einer in BPSS beschriebenen Choreographie möglich ist, gestattet die von uns vorgeschlagene Erweiterung auch die Erstellung von Orchestrierungen. In diesem Bereich werden die Anforderungen an EMLs für eine SOA gut erfüllt.

Im Bereich der EETs haben wir mit Hilfe eines Metamodellierungstools ein Modellierungstool entwickelt, welches alle für die erweiterte UMM notwendigen Modelle abbilden kann. Im Weiteren werden wir eine Core Components Bibliothek im Modellierungstool als GEMC anbinden, so dass der in den Geschäftsprozessmodellen dargestellte Informationsfluss semantisch hinterlegt werden kann. Dies ist notwendig, um eine automatische Bindung von Services an Aktivitäten zu ermöglichen.

Im Bereich der PEMs ist es möglich, Referenzprozesse zu hinterlegen, die bereits durch EMOs unterstützt werden können. Genauso wie für das Webservice-Verzeichnis müsste auch für Referenzprozesse ein Verzeichnis geschaffen werden. Die EEM wird von der UMM übernommen und ist bereits gut für SOA geeignet, da ein starker Fokus auf den Geschäftsprozessen und ihrer Unterstützung durch EDV liegt. Eine Erweiterung ist hier nicht notwendig.

Die Einordnung der Erweiterung in GERAM ist in [Tabelle 13](#) zusammengefasst.

Tabelle 13: Erweiterte UMM im Rahmen von GERAM

GERAM-Elemente	Bestandteil	Umsetzung in erweiterter UMM
GERA ..	Mitarbeiterorientierte Konzepte	In Arbeit
	Prozessorientierte Konzepte	Fokus von erweiterter UMM
	Technologieorientierte Konzepte	Nicht unterstützt
EEM		Methodologie zur Erstellung der Modelle von UMM übernommen
EMLs	Jeder Bereich ist modellierbar	Wird durch UMM Modelle gewährleistet
	Modelle sind miteinander integrierbar	Wird durch UMM Modelle gewährleistet
EETs		Prototyp in Metamodellierungstool entwickelt
PEMs	Partial Human Role Models	Konzepte sind in Arbeit
	Partial Process Models	Referenzmodell-Verzeichnis wird erstellt
	Partial Technology Models	Nicht betrachtet
	Partial Models of IT systems	Insbesondere Webservice-Verzeichnis
GEMCs		Core Components
EMs		In der Methodologie erstellte Modelle
EOS		Das modellierte Unternehmen
EMOs		Webservices

1.6 Schlussfolgerungen

Die Schlussfolgerungen beziehen sich zum einen auf die Erweiterung der UMM, zum anderen auf die Verwendung von GERAM als Referenzmodell für ein SOA Framework.

1.6.1 Erweiterung der UMM

Der Beitrag zeigt, dass die erweiterte UMM in vielen Bereichen der GERAM die durch eine SOA gestellten Anforderungen schon adressiert. Im Bereich von GERA und EML sollten zusätzlich für SOA notwendige Rollen und Lebenszykluskonzepte mit betrachtet werden. Wir arbeiten zurzeit an weiterreichenden Modellen für Rollen und Technologien, die den gesamten Servicelebenszyklus abdecken. Im Bereich der PEMs und EMOs erarbeiten wir im Zusammenhang mit einem Forschungsprojekt geeignete Lösungen in Form von Verzeichnissen, die auch in unser Modellierungstool eingebunden werden.

Durch die Einordnung der Methodologie in die GERAM ist es möglich, Schwachstellen und Stärken der Methodologie besser zu erkennen und somit gezielter zu einer ganzheitlichen Unternehmensmodellierung für eine SOA zu kommen.

1.6.2 Enterprise Architecture Frameworks für serviceorientierte Architekturen

GERAM stellt einen guten Rahmen zur Verfügung, in dem ein EAF für SOA entwickelt werden kann. Denn die für eine SOA notwendige ganzheitliche Sicht auf ein Unternehmen wird durch die von GERAM spezifizierten Elemente abgedeckt. Ein in diesem Rahmen entwickeltes Framework hat gute Voraussetzungen, die Einführung einer SOA und ihre Verwaltung zu erleichtern.

Wir werden im Rahmen der GERAM weiter an einem Framework arbeiten, dass die Belange einer SOA berücksichtigt. Die hier vorgestellte Erweiterung der UMM ist dabei nur ein erster Schritt zur Modellierung einer auf Services aufbauenden Unternehmensstruktur.

2 Beitrag II: Integriertes Performance Management als Teil des Enterprise Architecture Management

2.1 Abstract

Dieser Artikel befasst sich mit einer integrierten Performance Measurement Methode im Rahmen des Enterprise Architecture Management (EAM). Das Ziel des EAM ist es, sowohl organisatorische als auch informationstechnische Belange integriert zu steuern. Grundlegend für ein erfolgreiches Management ist es, ein detailliertes, quantifizierbares Bild des Systems zu entwickeln, wofür ein integriertes Performance Measurement Konzept notwendig ist. Heutige Konzepte fokussieren auf Teilbereiche des EAM, nicht jedoch auf ein integriertes Gesamtkonzept. In diesem Artikel wird ein neuartiger, integrierter Ansatz vorgestellt, der sowohl organisatorische als auch technische Belange aufnimmt.

2.2 Einführung

Enterprise Architecture Management enthält insbesondere im Kontext serviceorientierter Architekturen (SOA) eine erhöhte Bedeutung. Durch die strukturelle Anpassung von Informationssystemarchitektur und Organisationsarchitektur wird eine integrierte Steuerung im Rahmen des EAM möglich. Trotz dieser Perspektive sind SOA ein häufig technisch diskutiertes Thema. Dieser Artikel hingegen fokussiert auf die Performancebewertung im Rahmen des EAM unter besonderer Beachtung von SOA.

2.2.1 Ziel des Artikels

Der Artikel soll einen Beitrag zu integrierter Steuerung von Informations- und Organisationsarchitektur leisten. In heutigen Unternehmen werden nahezu sämtliche Tätigkeiten durch Informationstechnologie (IT) unterstützt, wodurch eine starke Abhängigkeit zwischen Informations- und Organisationsarchitektur entsteht. Diese Abhängigkeit erfordert eine integrierte Steuerung von Organisation und IT im Rahmen des EAM wobei in diesem Kontext insbesondere folgende Ziele für die Unternehmen relevant sind:

- Umsetzung von unternehmensstrategischen Vorgaben in Organisations- und Informationsarchitektur
- Gestaltung von IT mit dem Ziel effektiver Geschäftsprozessunterstützung sowie effizienter IT-Entwicklung und IT-Betrieb

- Gestaltung der Geschäftsprozesse mit dem Ziel effektiver Umsetzung der operativen Leistungsebene sowie effektiver und effizienter Prozessunterstützung durch IT
- Flexible Anpassung der EA auf neue Anforderungen

Damit diese Aspekte in Unternehmen umgesetzt werden können, sind unterschiedliche Konzepte wie Enterprise-Architecture-Frameworks und Technologien (z.B. Integrationstechnologien) entwickelt worden. Bei den Technologien wird zurzeit insbesondere das Konzept der Serviceorientierung in diesem Kontext diskutiert. SOA zielen auf eine Strukturierung der IT Architektur nach Prozess- und Organisationsaspekten ab. Somit ermöglicht eine SOA im Unternehmen den Brückenschlag zwischen Organisations- und Informationsarchitektur.

Um jedoch ein integriertes Management von Unternehmensarchitekturen zu ermöglichen, ist es notwendig, auch Bewertungen der Leistungsfähigkeit beider Architekturen konsistent zu integrieren. Eine solche Performance Measurement Methode zur integrierten Bewertung von Organisations- und Informationsarchitektur wird in diesem Artikel vorgestellt.

Als wissenschaftliche Methode wird die Methodenkonstruktion verwendet (Braun u. a., 2004). Die Methodenkonstruktion stellt ein planmäßig-systematisches Verfahren dar, welches zur Lösung eines theoretischen oder praktischen Problems führt. Die an der Praxis orientierte Methodenkonstruktion lässt sich wissenschaftstheoretisch zu den induktiven Vorgehensweisen zuordnen, wohingegen die theoretische Methodenkonstruktion deduktiv orientiert ist.

Im Folgenden werden zunächst die Begriffe EAM sowie Performance Management vorgestellt. Anschliessend werden einige Performance-Measurement-Konzepte (Abschnitt 2.3.1) präsentiert, die als Basis für das in dem Abschnitt 2.4 präsentierte integrierte Modell dienen.

2.2.2 EA Management

Der Begriff Enterprise Architecture (EA) umfasst sowohl den organisatorischen Aufbau (Organisations-Architektur) als auch die technologischen Einrichtungen (Informationssystem-Architektur) eines Unternehmens (Krcmar, 1990). Im Rahmen der Enterprise Architecture werden die Leitlinien der Unternehmensstrategie in operative Richtlinien und Strukturen umgesetzt. Diese Strukturen lassen sich in folgende Be-

reiche unterteilen: Aufbauorganisation und Governance, Ablauforganisation und Geschäftsprozesse sowie die Informationssystem-Architektur.

Obwohl sämtliche Teile der Architektur voneinander abhängig sind und somit beispielsweise in vielen Unternehmen die Aufbau- und Ablauforganisation stark an die Strategie gekoppelt sind, gibt es häufig eine Zäsur zwischen der technischen Architektur und der organisatorischen. Durch den unterschiedlichen Charakter beider Teilarchitekturen werden die Abhängigkeiten unterschiedlich bewertet.

Einige Autoren sehen einen einseitigen Effekt der Technologie auf die Organisation (technologischer Imperativ) (Markus & Robey, 1988), andere Autoren betonen die notwendige Dominanz der organisatorischen Aspekte (organisatorischer Imperativ). Beide Ansichten werden in der heute vielfach vertretenden Enabler-Perspektive vereint, die komplexe Wirkungszusammenhänge zwischen der Organisation und der Informationstechnologie annimmt (Frese, 2000).

2.2.3 SOA

SOA sind ein aktueller Trend in der Informations-Technologie (IT) Industrie (Gomes, 2006), auch Organisationen wie das World Wide Web Consortium (W3C), OASIS und die Object Management Group (OMG) entwickeln und publizieren Standards im Zusammenhang mit SOA.

Die Implementierung einer SOA soll es Unternehmen ermöglichen, die betrieblichen Anforderungen schneller und effektiver in der Informationstechnologie abzubilden, indem die Geschäftsprozessanforderungen als oberste Modularisierungsebene in Form von Services definiert werden (Orriens & Yang, 2005). Diese Services in einer SOA bilden fachliche Aktivitäten in den Systemen ab, welche analog zu den Geschäftsprozessen zu komplexen Applikationen orchestriert werden. Aus diesem Grund beschränkt sich das Konzept der SOA nicht auf die technische Seite, sondern reicht bis in das Geschäftsprozessmanagement hinein (Leymann u. a., 2002).

2.3 Performance Management und Performance Measurement

Der Begriff Performance Management umfasst sowohl strategische als auch operative Aspekte der Unternehmensführung. Mit ihm verknüpft werden alle Methoden und Prozesse, die im Rahmen des Managementsystems zur Planung, Steuerung und Kontrolle von Zielen und Zielerreichung eingesetzt werden (Cokins, 2004).

In Abgrenzung zum Performance Management befasst sich das Performance Measurement mit der Erfassung der Leistung mittels Kennzahlen. Insofern ist das Performance Measurement ein Bestandteil des Performance Management, welches zusätzlich aus dem Ableiten der Maßnahmen und deren Implementierung besteht. Performance Measurement wird dabei als Instrument für den Aufbau und den Einsatz von quantifizierbaren und mehrdimensionalen Messgrößen verstanden, welches die Effektivität und Effizienz eines Unternehmens auf unterschiedlichen Leistungsebenen bewerten kann.

Performance Measurement Konzepte haben ihren Ursprung in den vorwiegend auf finanzielle Kennzahlen ausgelegten Kennzahlensystemen. Ausgehend von diesen vorwiegend zur nachlaufenden Erfolgskontrolle ausgelegten Systemen wurden komplexere und auf unterschiedliche Operationalisierungsbereiche spezialisierte Konzepte entwickelt, deren Indikatoren auch Prognosequalitäten haben. Im Folgenden werden kurz einige Systeme erläutert, die jeweils für unterschiedliche Bereiche entworfen wurden und auch eine Basis für das abschliessend vorgestellte integrierte Konzept darstellen.

2.3.1 Strategische Performance Measurement Konzepte

Strategische Performance Measurement Systeme enthalten die Kennzahlen, aus denen sich strategische Aussagen ableiten lassen, weshalb primär Managementbereiche Adressaten dieser Konzepte sind. Beispiele für solche strategisch orientierten Systemen sind die Performance Pyramid sowie die Balanced Scorecard (BSC) (Kaplan & Norton, 1992). Insbesondere die BSC hat eine hohe Bedeutung erlangt, da sie an die jeweiligen Gegebenheiten anpassbar und ausgewogen komponiert ist. Sie enthält finanzielle und nicht-finanzielle Vorlauf- und Ergebnisindikatoren. Ausgangspunkte des Ansatzes sind die Vision und die Strategie des Unternehmens, die in unterschiedlichen Perspektiven aufgliedert werden. Unter Beachtung der sichtenübergreifenden Unternehmensstrategie gilt es sichten-spezifische Massgrößenbündel in Form von Zielen, Kennzahlen für diese Ziele, Vorgaben für die Werte der Kennzahlen sowie Maßnahmen zur Realisierung der Ziele zu entwickeln. Die BSC betrachtet vier solcher Perspektiven: Kundenperspektive, Finanzperspektive, Interne Geschäftsperspektive und die Lern- und Entwicklungsperspektive. Im Fokus des integrierten Konzeptes liegt dabei die Interne Geschäftsperspektive, die auch die operativen Geschäftsprozesse enthält.

Die BSC eignet sich insbesondere im Rahmen des integrierten Konzeptes, da sie aus strategischer Sicht eine Anbindung an die operative Leistungsebene ermöglicht. Dabei wird eine Prozessstruktur der obersten Gliederungsebene entwickelt, die im Rahmen der operativen Performance Measurement Systeme weiter detailliert werden kann.

2.3.2 Operative Performance Measurement Konzepte

Als Vertreter der Operativen Performance Measurement Systeme wird im Folgenden das Prozessorientierte Performance Measurement System (PPM) nach Gleich & Schimpf vorgestellt (Gleich & Schimpf, 1999). Operative Performance Measurement Systeme haben zum Ziel, die Prozessebene der Leistungserbringung zu bewerten. Essentielle Grundlage für das PPM und die anvisierte Prozessleistungstransparenz ist die Prozessstrukturtransparenz. Dafür muss ein unternehmensspezifisches Prozessmodell erstellt werden, das die Unternehmensabläufe hinsichtlich ihrer Leistungsebenen mit allen relevanten Prozessleistungsparametern integriert. Gleich & Schimpf (1999) verfeinern dazu die Leistungsebene Prozess zu Geschäftsprozessen, Hauptprozessen, Teilprozessen und Tätigkeiten. Geschäftsprozesse bilden in aggregierter Form die wesentlichen Unternehmensabläufe ab. Hauptprozesse sind bereichsübergreifende Prozesse mit einem einheitlichen Kostentreiber. Teilprozesse repräsentieren Tätigkeitsbündel, die innerhalb eines Bereichs beziehungsweise einer Kostenstelle auftreten. Tätigkeiten stellen schliesslich mit einzelnen Aufgaben als inhaltliche Konkretisierung der Teilprozesse die unterste Leistungsebene dar. Abbildung 21 zeigt, wie nach Gleich & Schimpf (1999) die unterschiedlichen Prozessebenen analysiert werden sollen, um eine konsistente Analyse aller Prozessebenen zu erhalten. Das PPM besteht zunächst nur aus dem Konzept, welche durch die Vorgabe konkreter Kennzahlen komplettiert wird.

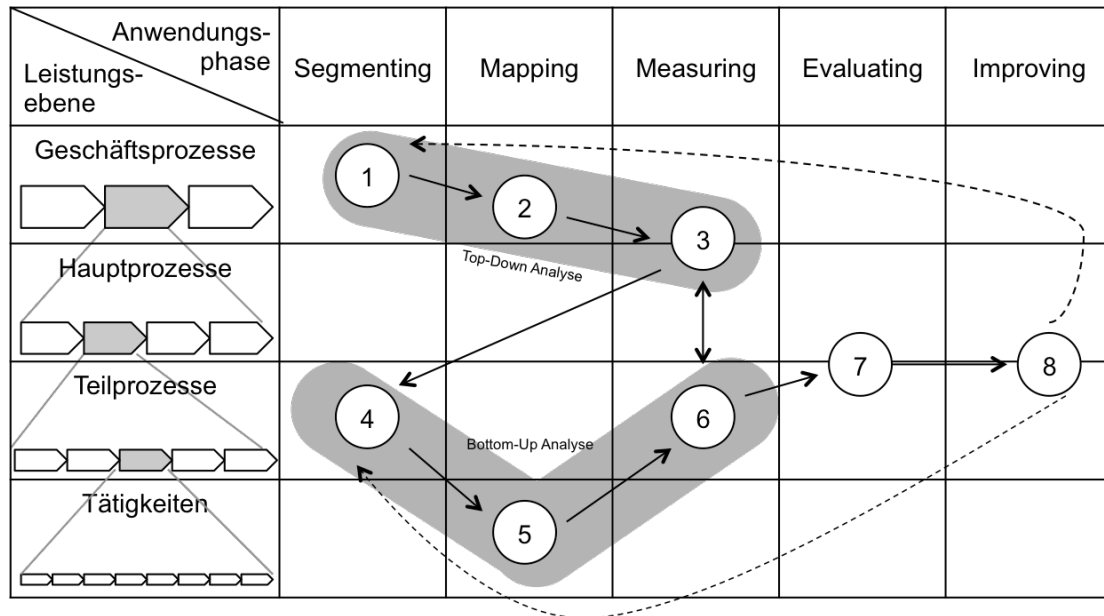


Abbildung 21: Kombiniertes top-down und bottom-up Vorgehen im Rahmen des PPM

Dies ist insofern sinnvoll, als dass Kennzahlen industrie- und prozessspezifisch sind. Es gibt Ansätze, solche Indikatoren insbesondere für das Benchmarking zu definieren. Beispielsweise wurden für produzierende Unternehmen durch die European Network of Advanced Performance Studies (ENAPS) 95 Indikatoren für sechs Prozessstypen identifiziert.

2.4 Integriertes Measurement Model

Das vorige Kapitel befasste sich mit Systemen und Ansätzen zum Performance Measurement für die Strategie und die organisatorische Architektur. Dabei wurden Aspekte der Planung, Steuerung und Kontrolle der Leistung und Leistungsfähigkeit bis hin zur operativen Leistungserstellung im Rahmen von Geschäftsprozessen betrachtet.

Dieses Kapitel soll nun die Möglichkeiten aufzeigen, wie das Business-Performance Measurement um eine zusätzliche Ebene, die des IT-Performance-Measurement, erweitert wird und in einem integrierten Modell dargestellt werden kann.

Dazu ist es notwendig, zunächst den IT-Performance Begriff im Kontext Service Architekturen zu definieren. Anschliessend wird die Gestaltung von Services und der IT-Performance Ebene dargelegt, die abschliessend in ein integriertes Modell eingliedert werden.

2.4.1 IT Performance

Die Funktion der IT bei der Leistungserbringung eines Unternehmens ist es, eine effiziente Durchführung zu ermöglichen. Daher lässt sich der Begriff IT-Performance im weiteren Sinne auf das Bereitstellen und Betreiben der gesamten IT-Infrastruktur beziehen (vgl. (Barton, 2002)). In dieser Definition fallen auch Aktivitäten wie beispielsweise Mitarbeiter-Support unter die Aktivitäten, die im Rahmen der IT-Performance bewertet werden müssen. Im Kontext des integrierten Modells zur Performance-Bewertung geht es jedoch nicht um eine isolierte Betrachtung der IT-Performance, sondern um die Bewertung der IT-Leistung ausgehend von der Geschäftsprozessebene. Dadurch lässt sich die Definition der IT-Performance wesentlich enger als Leistung von Softwaremodulen, deren Leistung wiederum auch von der unterliegenden Hardware- und Netzwerkebene abhängt, definieren.

Neben der Anwendungsdomäne ist auch noch die Art der zu bewertenden Leistung zu definieren. Dabei wird häufig der vornehmlich auf die Effizienz abgestellt. Jedoch ist insbesondere auch die Effektivität, die zum Beispiel die Flexibilität und Reagibilität eines Systems bewertet, mit in Betracht zu ziehen (Bub, 2003).

2.4.2 Gestaltung der IT-Performance Ebene

Erster Schritt bei der Gestaltung der IT-Performance Ebene ist die Gestaltung der Bezugsebene. Die Definition dieser Einheit wird an der Art des Zugriffs von Geschäftsprozessaktivitäten auf die IT festgemacht.

Aus der Sicht einer Tätigkeit innerhalb eines Geschäftsprozesses werden verschiedene Funktionalitäten von Informationssystemen nachgefragt, die vom Versenden einer E-Mail über die Bonitätsprüfung eines Kunden bis zu Videokonferenzen oder komplexen standortübergreifenden Produktionsplanungsoptimierungen reichen können. Alle diese Funktionalitäten werden im Rahmen einer SOA als Service definiert.

Ein solcher Service bildet somit die logische Einheit, der die IT-Performance zugeordnet werden kann. Um eine konsistente Anbindung der relevanten IT-Performance-Parameter an die Prozess-Performance Größen sicherzustellen, werden nachfolgend jene Eigenschaften der IT-Service identifiziert, die zur Erreichen der Performance-Ziele auf der Geschäftsprozess-Ebene beitragen. Prinzipiell kann dabei zunächst nach funktionalen und nicht-funktionalen Eigenschaften differenziert werden.

2.4.2.1 Funktionale Performance Parameter: IT-Service-Quality Modell

Die funktionalen Parameter werden bestimmt durch die konkreten geschäftsspezifischen Anforderungen der Prozesse. Dabei spielen die Branche, in der das Unternehmen tätig ist sowie der Kontext des Geschäftsprozesses innerhalb des Unternehmens eine entscheidende Rolle, weshalb keine generischen Eigenschaften definiert werden können. Im Rahmen eines IT-Performance Measurement sollte ein kontinuierlicher Abgleich zwischen den benötigten und tatsächlichen Funktionalitäten stattfinden, um erforderliche Anpassungen der IT an die Bedürfnisse der Geschäftsseite zu gewährleisten.

Die qualitative Natur funktionaler Eigenschaften impliziert eine Methode, welche analog zu den Leistungsparametern der Prozesse die Zufriedenheit der Service-Benutzer betrachtet. Ein Ansatz, der in der Lage ist, funktionale Merkmale von IT-Services abzubilden, sieht die so definierte Service-Qualität als adäquate Messgröße für die Performance von IT-Services an. Dieses Konzept basiert auf dem ursprünglichen im Marketingumfeld entwickelten SERVQUAL-Instrument von (Parasuraman, Zeithaml, & Berry, 1985), das mit Hilfe von Fragebögen auf das Verstehen und Messen von Kundenerwartungen abzielt und von Services überprüft, modifiziert und angewandt wurde (vgl. (Kang & Bradley, 1999)). Das Ergebnis der Analyse ist die Identifikation von Differenzen (Gaps) zwischen funktionaler Erwartung der Servicenutzer und Fähigkeiten des IT-Systems erzeugt (Kang & Bradley, 1999).

Das IT-Service-Quality-Modell von (Kang & Bradley, 1999) ist ein Instrument, um die schwer quantifizierbare funktionale IT-Performance anhand der Differenzen zwischen der Erwartung im Sinne der Anforderungen durch die Geschäftsprozesse, und der Wahrnehmung als Bezugsgröße für die tatsächliche erbrachte Leistung messen zu können.

2.4.2.2 Nicht-funktionale Performance-Parameter: Service Level Agreements

Um die nicht-funktionalen Anforderungen abbilden zu können, muss der rein fachlich beschriebene Service zunächst in technische Komponenten zerlegt werden. Diese technischen Komponenten müssen dann mit nicht-funktionalen Eigenschaften annotiert werden.

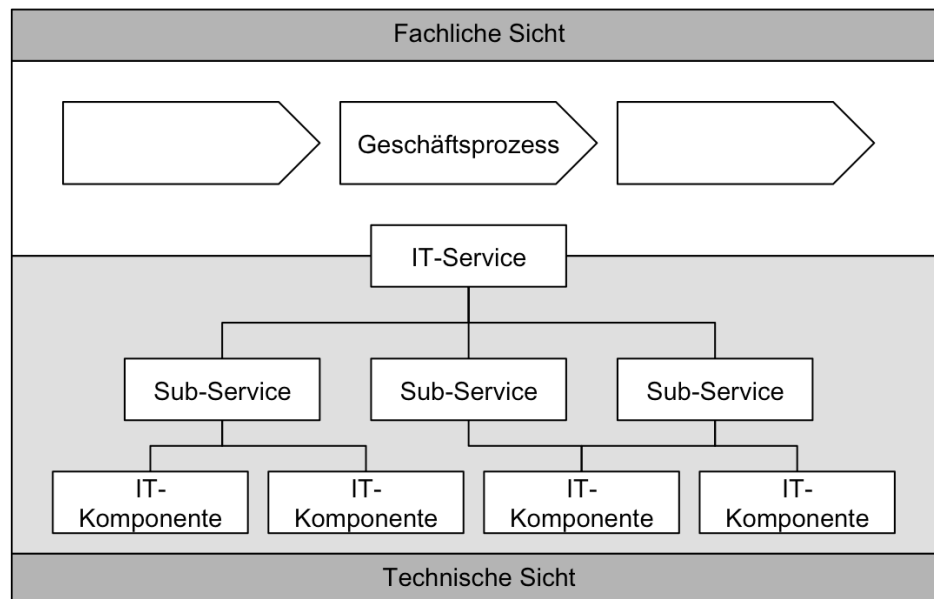


Abbildung 22: Dekomposition eines IT-Services (Zarnekow u. a., 2005)

Abbildung 22 zeigt eine schematische Dekomposition eines IT-Services. Dabei kann ein Service zunächst in weitere Sub-Services gegliedert werden, die jeweils einen Teil der Gesamtfunktionalität des Services zur Verfügung stellen. Schliesslich werden auch die IT-Komponenten betrachtet, die durch die erforderliche Infrastruktur für den Betrieb der Sub-Services und Services gegeben sind. Im Allgemeinen können zwischen Sub-Services und IT-Komponenten n:m-Beziehungen bestehen, dies heisst, dass eine IT-Komponente von mehreren Services genutzt wird und ein Service jeweils auf mehrere IT-Komponenten zugreifen kann. Die technische Sicht umfasst weiterhin die Implementierung der Services.

Die Anforderungen an die Leistungsgrößen eines IT-Services werden ausgehend von der fachlichen Sicht für die einzelnen Bestandteile des Services aufgebrochen. Für die einzelnen IT-Komponenten werden Kennzahlen festgelegt, die dann in den Sub-Services und IT-Services aggregiert werden können. Kennzahlen für ein solches System schlägt Berger vor (Berger, 2005).

2.4.3 Integriertes Modell

Ausgehend von Ansätzen zum Performance Measurement, die jeweils ein begrenztes Anwendungsgebiet in den Mittelpunkt ihrer Betrachtung stellen, sollen hier die im Vorangegangenen vorgestellten und entwickelten Teilmodelle zusammengefügt werden. Dieses integrierte Modell verbindet die einzelnen Wirkungsfelder und ermöglicht

so eine systematische Komposition von Instrumenten für ein ganzheitliches Performance Measurement. Dieses in Abbildung 23 illustrierte Modell umfasst im Wesentlichen drei Leistungsebenen.

Die strategische Leistungsebene bildet den Überbau des Modells. Ein geeignetes Instrument, das Performance-Ziele auf strategischer Ebene strukturiert und in einem ersten Schritt operationalisiert, ist die BSC. Dazu werden für die vier enthaltenden Perspektiven Zielvorgaben aus der Unternehmensvision abgeleitet und an die nachrangigen Managementebenen kommuniziert. Diese nächste Leistungsebene widmet sich der Prozess-Performance. Die Prozessperspektive aus der BSC, die für den Einsatz auf operativen Level nicht präzise genug gestaltet ist, wird hierbei durch das Vorgehensmodell des PPM konkretisiert. Von den strategischen werden operative Zielvorgaben abgeleitet, die durch Kennzahlen messbar gemacht werden, welche relevanten Ursache-Wirkungs-Zusammenhänge für die Erreichung der übergeordneten Ziele abbilden. Anknüpfungspunkte zwischen BSC und PPM ergeben sich auf der höchsten Aggregationsebene in Form der Geschäfts- oder Hauptprozesse des PPM.

Die IT-Leistungsebene wird nun als Erweiterung der bis dahin reinen Betrachtung der Business Performance hinzugefügt. Im Zentrum steht hier die Sicht der operativen Leistungsebene auf die IT-Services, die sich in funktionale und nicht-funktionale Anforderungen an die IT gliedert. Wiederum werden die Zielvorgaben durch die Geschäftsprozesse, nun aber in deren feinsten Granularität in Form von Tätigkeiten, in Leistungsziele für die IT-Systeme übersetzt. Diese können bezüglich der funktionalen Parameter durch das Konzept der IT-Service-Quality und im Hinblick auf die nicht-funktionalen Aspekte mittels SLAs spezifiziert, gemessen und kommuniziert werden. Entscheidend für die erfolgreiche Implementierung ist die Fokussierung auf relevante Kennzahlen. Je nach Industrien und Prozess gibt es, wie im Vorangegangenen ausgeführt, spezifische Parameter.

Im Rahmen des EAM kann nun das Performance Measurement auf allen Leistungsebenen durchgeführt werden. Dabei werden die erfassten Informationen über den Zielerreichungsgrad sowohl auf der jeweiligen Ebene mit den geplanten Zielen in Beziehung gesetzt als auch in Form von Ergebnissen der Leistungsmessung in die darüber liegenden Performance Management Prozesse eingesteuert. Im Rahmen des Performance Management Prozesses können so die jeweiligen Kennzahlen in

einem kombinierten top-down / bottom-up Ansatz zu einem Kennzahlensystem zusammengefügt werden.

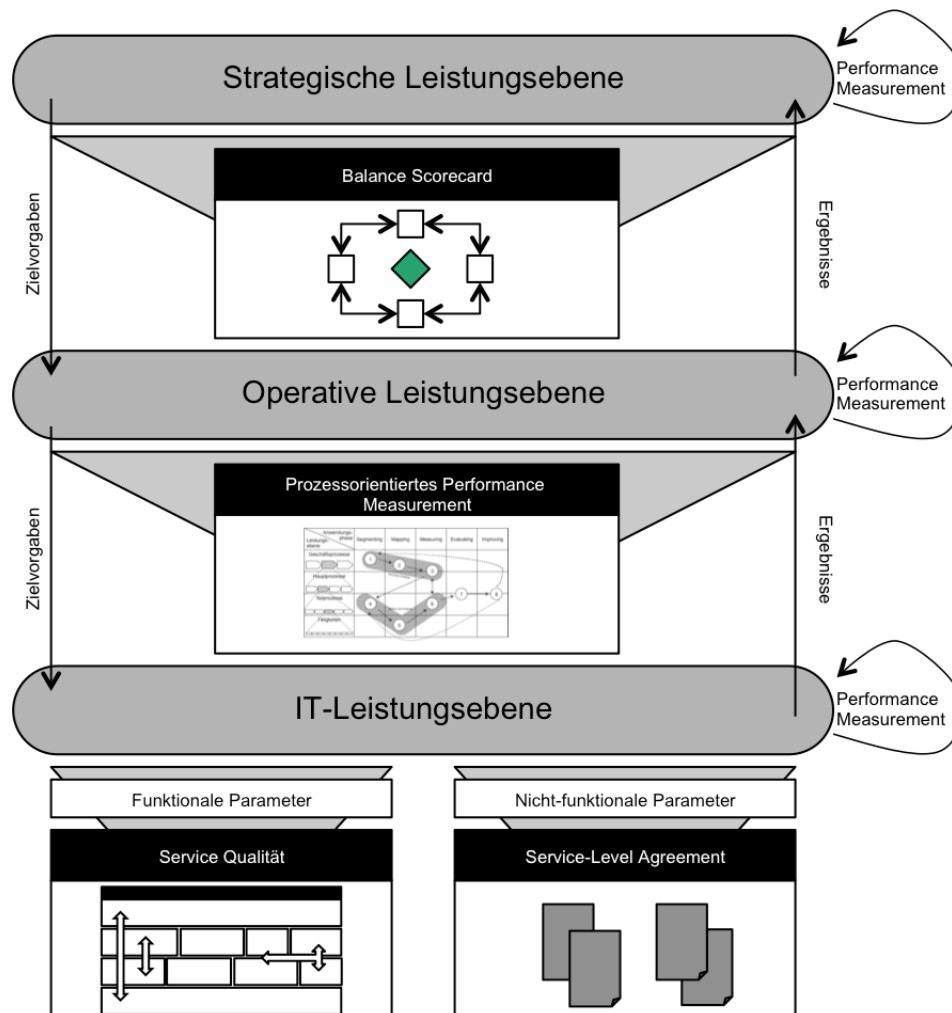


Abbildung 23: Integriertes Modell

2.5 Fazit und Ausblick

Die vorgestellte Methode erlaubt es Unternehmen im Rahmen des EAM eine integrierte Performance Measurement Methode zu implementieren. Die Methode wurde an einem Beispiel basierend auf den zitierten Kennzahlensystemen positiv evaluiert. Für ein Unternehmen hat die konsistente Ableitung der Kennzahlen insbesondere den Vorteil, dass die im Sinne der Enabler-Perspektiven komplexen Abhängigkeiten von IT und Organisation korrekter in dem Modell abgebildet werden und somit fundierte Managemententscheidungen erlauben.

Im weiteren Verlauf der Forschung ergeben sich einige interessante Zielsetzungen. Das nächste Ziel ist die Implementierung des Ansatzes in einem Unternehmen zur Evaluation der Methode. Weiterhin sind die gegenseitigen Beeinflussungen der Kennzahlen auf den unterschiedlichen Ebenen genauer zu erforschen. Insbesondere existieren viele indirekte Abhängigkeiten, die noch nicht in dem Modell integriert sind. Weiterhin ist es sinnvoll, Performance Management Methoden im Rahmen des EAM zu entwickeln, die aus der hier vorgestellten Bewertungsmethode Massnahmen ableiten und implementieren.

3 Beitrag III: Deriving SOA Evaluation Metrics in an Enterprise Architecture Context

3.1 Abstract

Service oriented architectures (SOA) are becoming reality in a corporate environment. Rather than pure technology improvements SOA intends to increase manageability and sustainability of IT systems and to better align business requirements and technology implementations. Following these intentions current SOA research is focused on the management of SOA. In this paper we present a method to identify metrics for an evaluation of SOA. Therefore we adopted the goal/question/metrics method to SOA specifics. Since SOA is not limited to questions of technology we will show where SOA might be located in an enterprise architecture (EA) context. Based on the assumption that SOA follows different goals on different levels of EA abstraction, the paper shows how these goals can be developed to metrics which can be consolidated in a measurement program.

3.2 Introduction

Service Oriented Architecture (SOA) is one of the recent approaches addressing the complexity of today's IT architectures. While SOA has been discussed in the scientific community, practitioners have also started to implement SOA in their organizations. However, organizations implementing SOA face different challenges, e.g. which are the goals to aim at, how to evaluate whether the implemented SOA achieved the defined goals, and – if not – which countermeasures should be taken? All of these questions are related to SOA management approach. Two essential elements of every management process are the measurement and evaluation of the current state.

In this paper we present a method to design a set of metrics to measure the success of SOA. With these metrics the architects have a set of indicators showing the impact of each of their decisions during the process of building and maintaining SOA. Thus architects will be better equipped to align the overall architecture with the set of desired qualities (Vasconcelos u. a., 2007).

Our underlying research method is based on a design science approach (Hevner u. a., 2004): After summarizing conceptual foundations we will review other approaches in the related work section and develop our method for the evaluation of

SOA. We will also show how the method proposed can be applied to design a SOA measurement system. On the basis of several case studies, we will present our findings from applying the method in the final section.

3.3 Conceptual Foundations

The main idea of SOA is to encapsulate business functionality into small loosely coupled services which – once they are implemented on a software level – may be flexibly orchestrated to match the particular and ever changing business requirements. While SOA is often discussed from a technology perspective (Gold, Mohan, Knight, & Munro, 2004; Haas & Brown, 2004), its actual potential is to align business requirements and processes with technical implementations (Rabhi, Yu, Dabous, & Wu, 2007). This issue is regularly discussed as Business/IT alignment (Luftman & McLean, 2004). Therefore it is reasonable to link SOA to the concept of enterprise architecture (EA) in order to systematically address non-technical issues too (Lankhorst, 2013). Usually there is no single enterprise architecture but several partial architectures which may be organized in layers (CIO-Council, 1999; Harrison, Blevins, & Homan, 2007). Based on an analysis of architecture meta-models Winter and Fischer differentiate five layers with their respective architectures (Winter & Fischer, 2007): Strategy layer (business architecture), organizational layer (process architecture), integration layer (integration architecture), and the system layers (software architecture, infrastructure architecture) (Figure 24).

The integration layer is of particular importance. It couples artefacts of the software layer and the organizational layer. In the notion of SOA these couplings represent enterprise services (Schelp & Winter, 2007). These considerations outline the scope of the SOA management and measurement system. On the organizational layer the primary goals are effectiveness and efficiency (Österle, 1995), on the software layer the primary goal is reuse, and on the integration layer the goals are alignment, adequate IT support for business processes, and ultimately agility (Winter & Fischer, 2007; Yusuf, Sarhadi, & Gunasekaran, 1999).

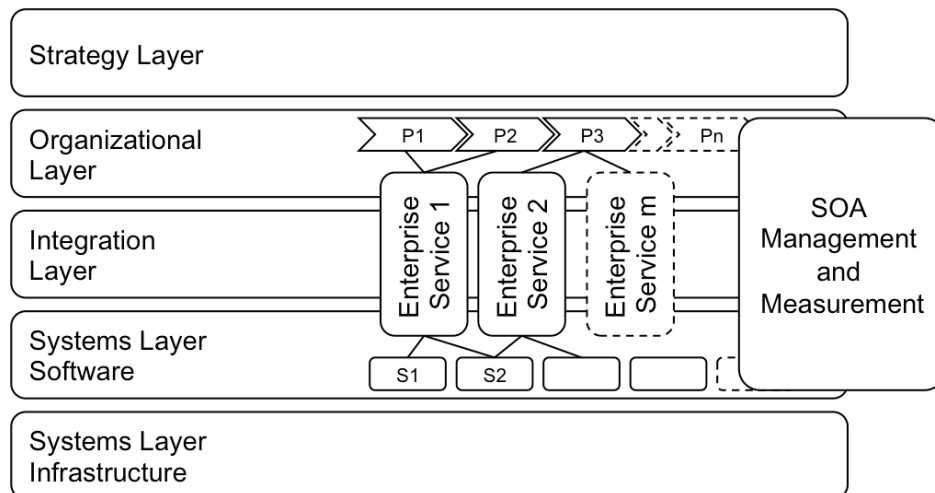


Figure 24: Enterprise Architecture, layers and services (Winter & Fischer, 2006)

After we have analyzed the related work in the following section, we will present a method that will facilitate these top level goals to a measurement program.

3.4 Related Work

Related work to the evaluation of SOA can be clustered in three different groups. The first group is formed by approaches that enable CIOs to argue in favour of the introduction of a SOA. These approaches are primarily sales-driven top down methodologies that allow rough estimations of financial benefits. These approaches are mostly presented by consulting companies (e.g. (Barnes, 2006; DiMare, 2006)) and often lack scientific methodology.

The second group of approaches primarily focuses on IT aspects of SOA evaluation. Many authors present approaches like the IT balanced scorecard to evaluate existing IT assets (Son, Weitzel, & Laurent, 2005). Other authors define and evaluate quality attributes which may refer to single applications (Johnson & Lagerstrom, 2006) or to the entire IT architecture (Khaddaj & Horgan, 2004; Vasconcelos, da Silva, Fernandes, & Tribolet, 2004).

The third group of related work can be identified in IT related extensions of process performance concepts, e.g. Gammelgård & Närman (2007) propose to use the performance indicators of the CobIT framework (Gammelgård, Närman, Ekstedt, & Norström, 2006). Mooney et al. present a framework for process oriented business

value measurement of IT combining existing approaches of the current literature (Mooney, Gurbaxani, & Kraemer, 1996).

Eventually, within the existing literature no methods for SOA evaluation in the context of EA exist. Either the approaches are driven from a technology perspective lacking a process and management view or they are driven from a business perspective not including technical aspects.

3.5 Method Construction

Our method to design a set of metrics to evaluate SOA focuses on the different goals SOA may pursue and takes the individual situation of the respective organization into account. Hence we propose to use a tailored version of the Goal/Question/ Metric (GQM) method (Basili & Weiss, 1984).

GQM follows a top-down approach by first defining goals, then deriving questions and eventually deriving metrics. In our case of evaluating SOA the organizational layer, the integration layer and the software layer are affected. Although, as rooted in the nature of these layers, each layer has its separate goals, there is a hierarchy of goals according to the hierarchy of the layers. Strategic goals do affect the goals on the layers below. After the definition of goals, questions have to be derived which take the different aspects of a goal into account. Finally, based on these questions, metrics can be defined which are the basis for answering the questions. The questions in turn can be understood as guidelines of how to interpret the measured metrics (Figure 25).

Following the GQM method a set of metrics has to be tailored to fit the individual situation. In order to raise the quality and acceptance of the measurements this will be done in structured interviews with the respective stakeholders. In accordance with van Latum et al. (1998) we introduce so called abstraction sheets to facilitate these interviews (Figure 26). These abstraction sheets are the basis for deriving questions and metrics (Latum u. a., 1998).

Architecture Context

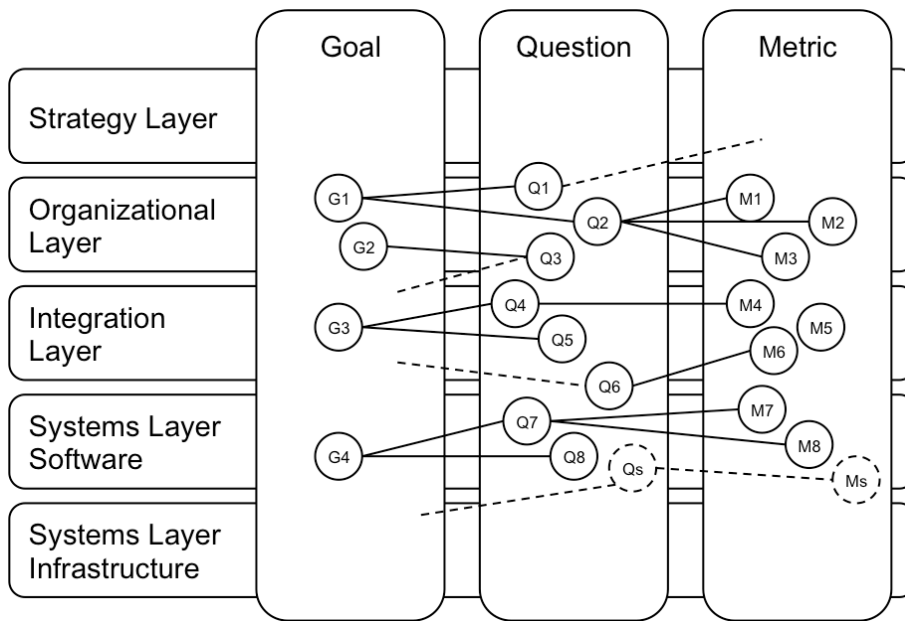


Figure 25: Example GQM abstraction sheet (Latum u. a., 1998)

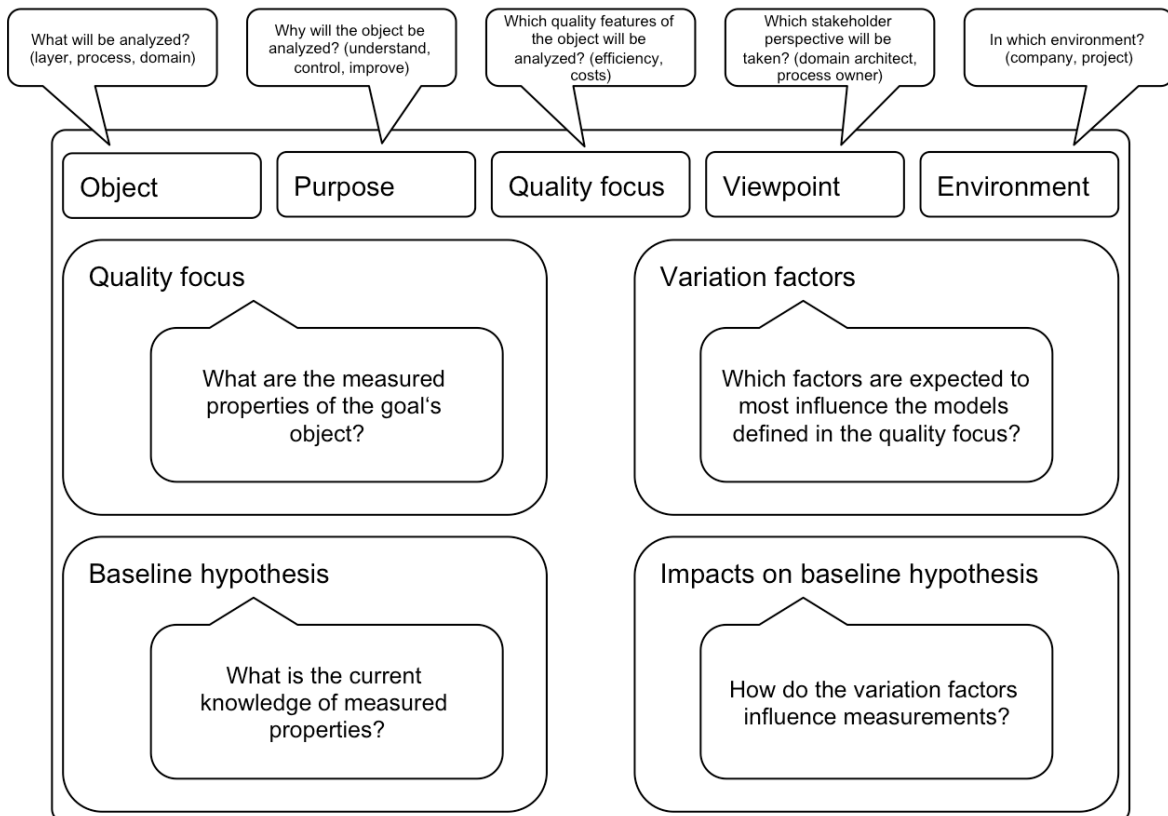


Figure 26: Application of the Goal/Question/Metric method in the context of SOA

After defining goals to questions to metrics, these metrics have to be consolidated in a measurement program. The aim of the consolidation is to reduce the number of metrics by isolating independent metrics and focusing on metrics which are applicable for answering several questions. In the EA environment this step requires a more formal approach than the one being used in the standard GQM method. This is caused by the highly complex environment as well as the involvement of many different stakeholders. We propose to use Extended Influence Diagrams (EID) for this formalization step (Figure 30). The goal of the GQM method is depicted as a utility node within the influence diagram. Nodes within an EID can be connected by two different relationship types: causal and definitional.

Within the EID uncertainty of the chance, nodes can be addressed by adding probability information to the nodes. This allows to reflect the quality of the metrics and to differentiate between directly and indirectly measurable metrics. For example if the goal is to have a very much quantifiable analysis, the influence of the indirectly measurable metrics could be reduced. It is important to analyze, whether all of the consolidated metrics are actually measurable, i.e. the metrics have to be available with a reasonable effort.

Eventually a measurement program has to define how the set of metrics will be embodied in the organization, i.e. who will when measure which metrics.

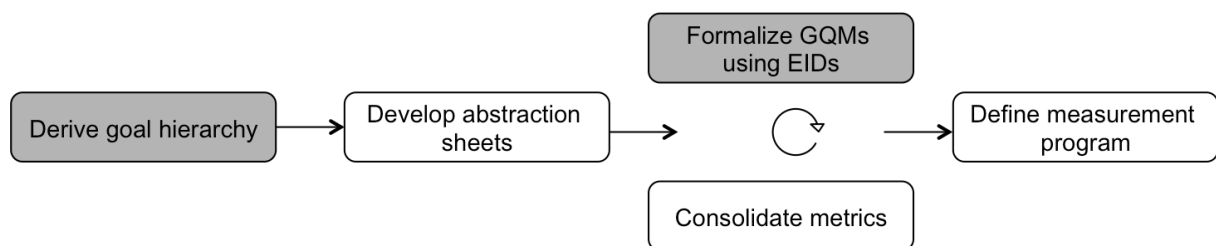


Figure 27: Adopted GQM method, additional steps in shaded grey

3.6 Application of the Method

This section describes the evaluation results and exemplarily shows a number of resulting artifacts of the method application. We primarily refer to a case study that has been conducted in Deutsche Telekom, which is the largest German telecommunications provider. The method was used within a project that aimed at understanding the

current state of enterprise services and on identifying potentials for improvement. The actual project used a traditional portfolio approach without employing the adopted GQM method. To evaluate our method we ex post used the project data with the original project results as a benchmark to our method. According to the presented method in the first step a goal hierarchy has been developed. A simplified example is shown in Figure 28.

The implementation of the adapted GQM method has been supported by the abstraction sheet shown in Figure 29.

As the first step within the evaluation method, object and purpose need to be defined. We defined the Enterprise Service 1 including related software objects (S_i) and process segments (P_j) of the layered business model as depicted in Figure 24.

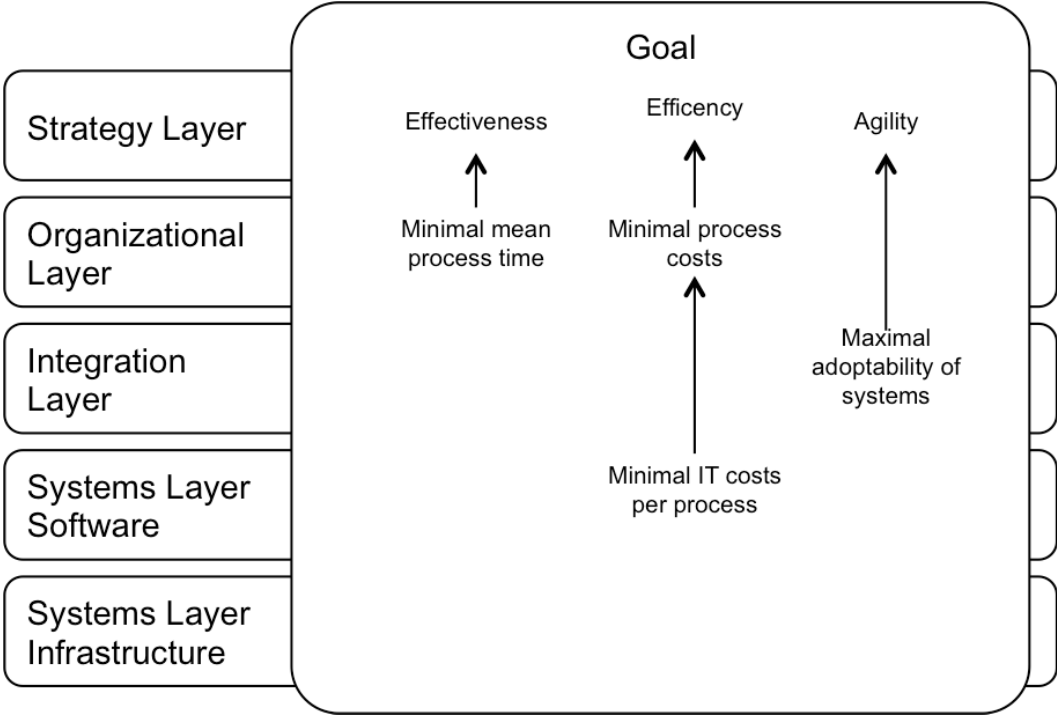


Figure 28: Case study goal hierarchy

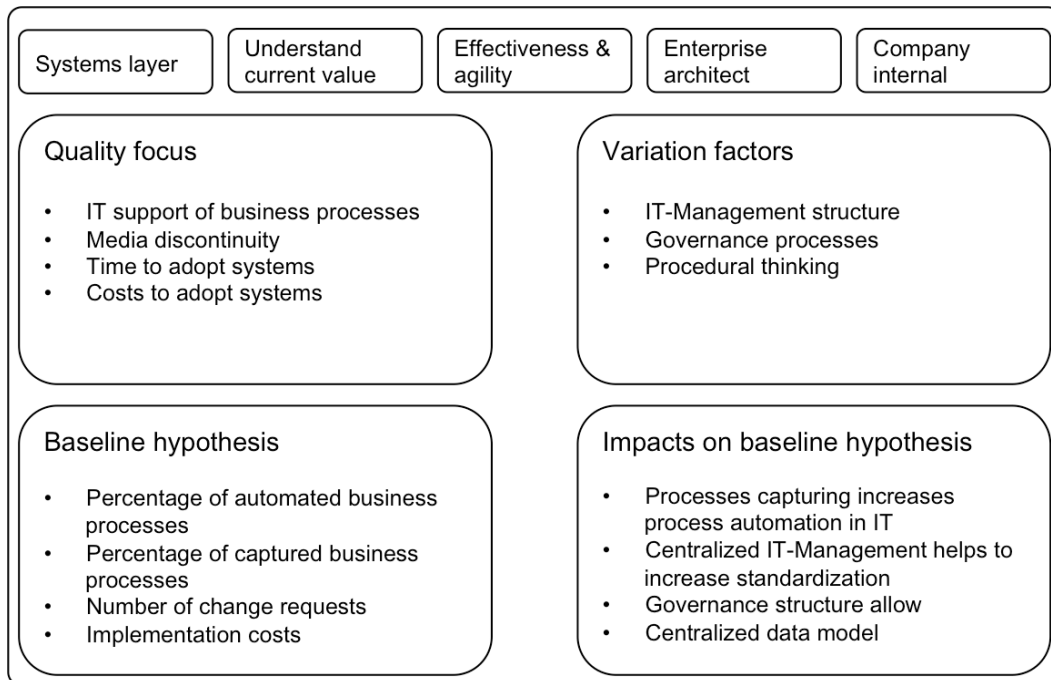


Figure 29: Simplified abstraction sheet

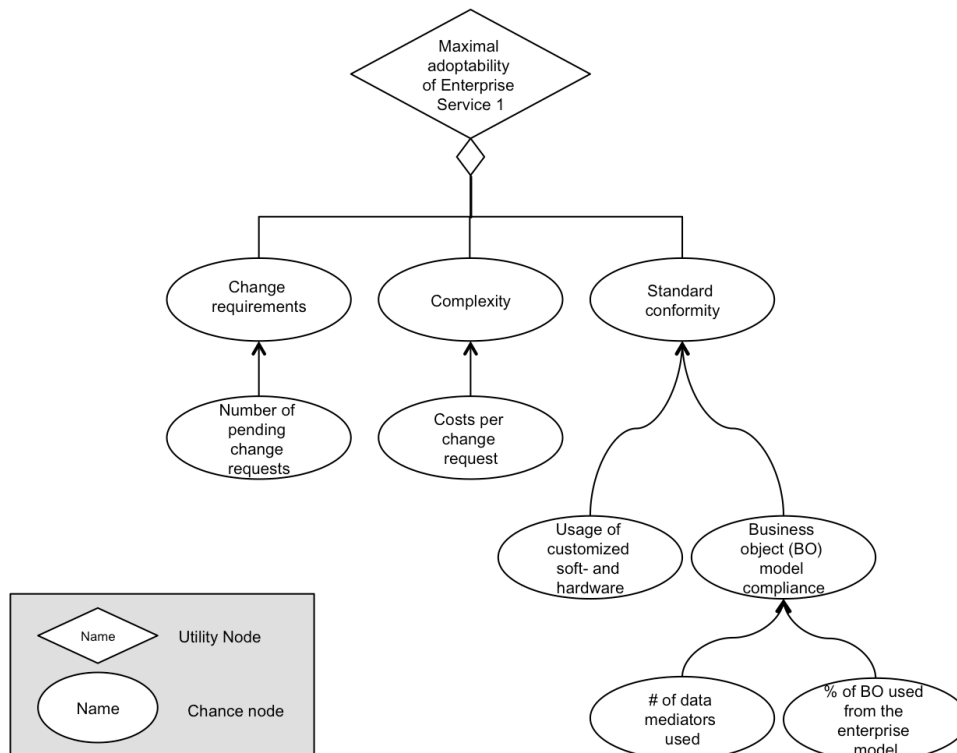


Figure 30: GQM EID hierarchy

Part two of the abstraction sheet defines the questionnaire outline. The detailed quality focus tries to find the characteristics of the top-level quality. In our analysis the

effectiveness goal has been divided into several sub-characteristics, like media discontinuity, adoption costs, etc. These attributes define the quality focus more precisely and later on help to define the actual metrics.

Part three of the abstraction sheet describes the as-is analysis of the artefact. The abstraction sheet guides the interviewee to metrics and the assumed impact of the variation factors on the measures of the baseline hypothesis.

First the questions are clustered and condensed to the most relevant question within each of the clusters. The second step is the assignment of the metrics that have been identified. To do so the baseline hypothesis and the impacts on it are analyzed. An example for an indirectly measurable metric is shown in the question depicted in Figure 31.

For the formal representations of the impact factors we use EIDs (Figure 30).

After having defined the GQM hierarchy the measurement plan is defined, containing measurement policies and organizational structures. For every metric the measurement frequency, the roles responsible for measuring and the actual measurement process have to be defined.

As a result of the case study we found that the structured approach of the GQM method allows building a consistent and focused set of metrics that are linked with goals which are derived from strategic viewpoints. When comparing the method presented here to other approaches, several advantages were found, three of which should be briefly named. The first improvement is the usage of goal hierarchies, which allows a consistency management of the various goals within the EA. Secondly the formalization approach allows a more precise consolidation and comparison of the used metrics. Third, after discussing the approach with the stakeholders involved in the original project, we found that this structured, formalized, and thus transparent approach created a much higher acceptance of the metrics among the stakeholders.

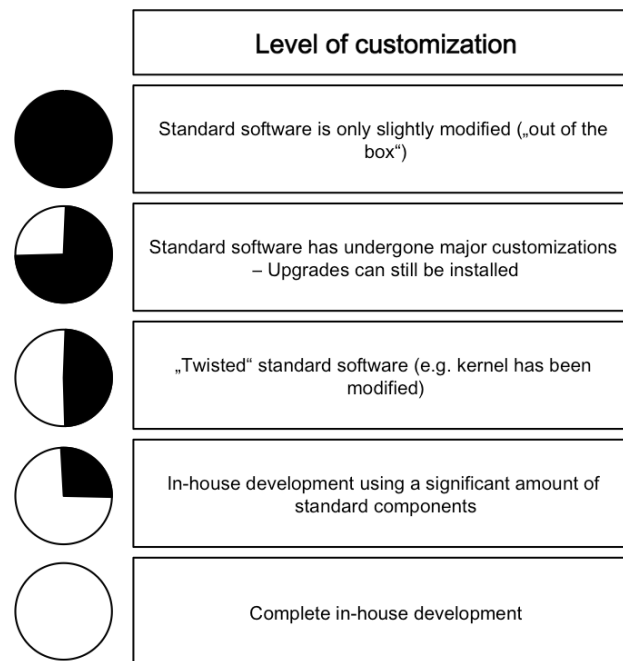


Figure 31: Example questionnaire for measuring the customization level

3.7 Summary and Outlook

In this paper we have described a method for the definition of metrics that can be utilized to evaluate SOA. Our method is based on the GQM method and thus ensures to deliver situational metrics, which are derived systematically. However, we assume that it is possible to identify certain SOA project types and certain context factors which can be combined with situations. If it is possible to identify an appropriate number of relevant situations, we could define a number of reference metrics which were immediately available for application. So our next research activities will focus on the identification of relevant project types and context factors.

4 Beitrag IV: A Flexible Approach to Service Management-Related Service Description in SOAs

4.1 Abstract

In order for service-oriented architectures (SOAs) to deliver their true value for the business, e.g. flexibility and transparency, a holistic service management needs to be set up in the enterprise. To perform all the service management tasks efficiently, heavy support by automated processes and tools is necessary. This article describes a service description approach that is based on OWL-S (Web Ontology Language for Services) and focuses on non-functional criteria. It starts with the necessary service management tasks and explains non-functional data elements and statements for its automated support. After covering related work it explains the proposed flexible extension to OWL-S. This extension is twofold. Firstly, simple service lifecycle elements are added using the normal extension mechanism. Secondly for adding QoS (Quality of Service) capabilities, the approach combines this extension mechanism with UML (Unified Modeling Language) Profile for QoS. A prototype delivers the proof-of-concept.

4.2 Introduction

In the last years, a lot of work regarding practical usability of technologies in the SOA (service-oriented architecture) area and especially Web services area has been done. Research work is more and more shifting from the technical areas like reliability and security to the business layer. One of the problems is the operational management – or IT service management – of an actual implemented service-oriented IT landscape in the enterprise. ITIL (IT Infrastructure Library, see <http://www.itil.co.uk/>) is a general and widespread IT service management framework. Being a de-facto standard, many other service management frameworks are based on it (Sallé, 2004). Among others, it covers best practices along two areas, *Service Support* and *Service Delivery* including configuration management, incident management, problem management, change management, release management, service level management, capacity management, availability management, IT continuity management, and financial management (Sallé, 2004). Part of the “IT service management” within the SOA is the “service component management” which deals specifically with managing the service components, e.g. Web services, during their lifecycle.

Due to the special characteristics of an SOA, its operational management is different from managing mature architectures. Additional requirements need to be covered. In an SOA, the implemented Web services are most likely much more fine granular than “normal” applications. In one landscape, there exist services that offer similar functionality and have different lifecycle stages. A high number of services need to be managed while a high reuse rate is a primary goal. At the same time in order for SOA to deliver its advantages, changing services and their orchestration should be easily possible. When managed without automated processes, tool support, and centralized repositories, these conditions can lead to confusion and chaos. The contrary of the original goals of SOA, among others more flexibility and more efficient IT, would be the outcome. Hence, an effective and efficient service management framework for SOAs is needed that is supported by automated processes and tools. The following SOA-specific functional blocks should be covered: service definition, service deployment lifecycle, service versioning, service migration, service registries, service message model, service monitoring, service ownership, service testing, and service security (Woolf, 2006). They represent SOA-specific functionalities in the broader area of the ITIL processes. (Edmond, OSullivan, & Hofstede, 2002) highlight the importance of service description and in particular non-functional service description for managing SOAs and mention in addition service discovery, substitution, and composition. Modeling functional and non-functional information in a machine-readable and semantically enriched way is a basis for a highly automated management of SOAs and in a broader sense of IT service management.

This article looks at a flexible service description approach to non-functional information. In Web services technology, UDDI repositories (Universal Description, Discovery, and Integration) and WSDL (Web Services Description Language) are used for service publication, discovery, and description but do not provide the necessary semantic functionality. Compared to the functional area less work has been done in the area of semantically enriched non-functional service description. Hence, this paper especially deals with the latter part. The approach builds on OWL-S (Web Ontology Language for Services). The two aspects that form the basis in the non-functional area are service lifecycle information and QoS (Quality of Service) guarantees offered by a service. Hence, it is necessary to look at semantic Web service description standards in general as well as to description standards in the QoS domain.

The remainder of this paper is organized as follows: In section 4.3, the requirements for describing services are examined. Section 4.4 gives an overview over the related work. Section 4.5 describes the extensions to OWL-S and section 4.6 the prototype. Section 4.7 describes the importance of this approach for matching, SLA negotiation, and SLA enforcement.

4.3 Requirements for Service Description

4.3.1 Requirements Overview

In order to support the above mentioned activities, like semi-automatic discovery, service level management, and service migration, several types of information need to be modelled within the service description. The following two sections describe requirements for service description regarding information relevant for service lifecycle management and QoS guarantees. Two aspects have to be considered: the content and the type of statements that can be modelled. The lists contain the most obvious points in both aspects. However, they cannot be regarded as complete. The available sources (e.g. (Dobson, 2004; Edmond u. a., 2002; Frølund & Koistinen, 1998; Maximilien & Singh, 2004)) describe very different non-functional characteristics. In order to be future-proof, the approach must allow for extension of both ontological terms and structure of statements used for description. Building on this extensibility, domain specific models can be built that capture most requirements relevant for the domain.

4.3.2 Information Relevant for Service Lifecycle Management

In the area of service lifecycle management, the following most obvious information should be covered as a starting point:

1. Service name
2. Service categories
3. Versioning information
4. Lifecycle status (“Planned”, “Design”, “Test”, “Pilot”, “Active – intensive maintenance”, “Active – regular maintenance”, “Sunsetting candidate”, “Sunsetting in progress”, “Sunsetted”) (based on (Woolf, 2006), extended)
5. Service provider information

6. Infrastructure the service runs on: server name, configuration management ID, etc.
7. Link to source code
8. Different responsibilities, roles, persons, e.g. for business aspect or maintenance
9. Link to further business description of the service
10. Pricing information (depending on QoS class)

For lifecycle management, the following obvious statement structures should be covered as a starting point:

1. Parameters with simple values, e.g. versioning information
2. Parameter names with RDF (Resource Description Framework) pointers to terms from predefined ontologies or resources (configuration database IDs for related infrastructure). Technically, this includes 1.
3. Tabular expressions, e.g. listing responsibilities for several areas
4. Free textual statements for a human reader

These statements are not very complex. As shown later, they can be realized relatively simply with OWL-S extensions. Free textual statements are introduced (also for the QoS) because we assume that in the first step it is not reasonable to put semantics behind every statement for automatic processing. Rare statements should be left for a human being to work with.

4.3.3 QoS Guarantees

Table 14 exemplarily describes QoS characteristics to be modelled in the service description.

Table 14: QoS information

QoS area	Explanation / Example
General area	
QoS-level	Service level regarding performance and quality (“Gold”, “Silver”, and “Bronze” are defined in a separate SLA document)
Service category	Type of service/service domain (several categories per service possible)
Communication	Communication pattern (e.g. real time and batch)
Cost area	
Price	Specification of tariff models, e.g. per period of time, per service call, and volume-fixed
Performance area	
Time	Response time
Capacity	Data capacity of a database (normal/max after extension)
Accuracy	Accuracy of the result of a calculation
Arrival pattern	Jitter; arrival distribution
Ratios	Number of service requests per time period (throughput of data sets, calculations per time) – (normal/max after extension)
Quality area	
Functional correctness	Error rate

Table 14: QoS information

QoS area	Explanation / Example
Reliability	Availability, business hours (weekdays/times), incident resolution time
End user usability	Rating with respect to ease of use/understanding
Security	Security level (high, medium, low – defined in separate document: encryption standard, access rights, and authenticity)
Other boundary conditions	
Organizational	Negative/positive list of partners
Cultural	Languages needed for end user communication
Normative	Compliance with laws/regulations, certification

The following structures of QoS statements should be supported as a basis to facilitate rich QoS specification in service description:

1. *Boolean statements*, e.g. “Component is Basel II certified – yes/no.”
2. *Absolute requirements*, e.g. “Reliability should be at least 99.9%.”
3. *Composed requirements*, e.g. “On weekdays, between 7am and 8pm, availability should be 99.9%; Otherwise, reliability should be 99%.”
4. *Level statements*, e.g. “The QoS requirements as defined in level “Gold” should be complied with.”
5. *Percentile statements*, e.g. “In 95% of the cases, response time should be below 10 ms.”
6. *Free textual statements* for a human reader

In addition, it should be possible to specify several sets of QoS guarantees (QoS-level) with added price tags for one Webservice that can be referred to during SLA (Service Level Agreement) negotiations.

4.4 Related Work

4.4.1 Standards for Service Description

A number of standards have evolved in the area of semantic service description. A quite mature one by now is OWL-S. OWL-S is an upper ontology language developed by the Semantic Web Services arm of the DAML (Darpa Agent Markup Language) program (D. Martin u. a., 2004; David Martin u. a., 2006). While using OWL-S, it is possible to describe Web services, their properties, and capabilities in a semantically enriched form. Given this, we have chosen OWL-S as the basis for our service description approach for two reasons. First of all, it is based on OWL, a well-established ontology language. Secondly, there are robust tools available for working with OWL ontologies as well as with OWL-S service descriptions. Both reasons support the intention of this article to show that, based on today's technology, standards, and tools, a reasonable basis for service management can be realized.

Other relevant semantic Web service description standards are WSMO (Web Services Modeling Ontology) and WSDL-S (WSDL with semantic extension). WSMO is a part of the WSMF (Web Services Modeling Framework) (ESSI WSMO Working Group, 2006). Its distinctiveness lies in its capability to import ontologies specified in other ontology languages, among others OWL, its usage of *mediators* bridging the gap between different Web services, as well as its *goal* concept describing functionality and interfaces from a user perspective.

WSDL-S heavily leverages the existing standard WSDL and is focused on compatibility (Akkiraju, Farrell, & Miller, 2005). It also is very flexible with respect to ontology languages (e.g. OWL) and mapping languages. However, being so flexible it is also more generic than WSMO and OWL-S.

4.4.2 QoS Specific Standards – UML Profile for QoS

Specification of QoS characteristics is an important topic in the area of IT systems. The existing standards can be grouped according to their main focus: software design/process description (e.g. UML Profile for QoS and QML – QoS Modeling Language (Frølund & Koistinen, 1998)), service/component description (e.g. WS-Policy), and SLA-centric approaches (e.g. WSLA – Web Service Level Agreements (IBM Corporation, 2006b; Ludwig, Keller, Dan, Ling, & Franck, 2003)), WSOL – Web Service Offerings Language (Tosic, Patel, & Pagurek, 2002), SLAng – Service Level

Agreement definition language (Lamanna, Skene, & Emmerich, 2003), and WS-Agreement (Andrieux, Czajkowski, Dan, & Keahey, 2007)). A good overview over most of them can be found in (Dobson, 2004).

Several languages have been developed to support SLA negotiation and specification in a service provider/service requestor scenario. The SLA-centric approaches are closely linked to the problem of QoS characteristics specification. The difference to other QoS specification languages is that they are more targeted towards SLA negotiation, specification, and SLA management.

UML Profile for QoS is a comprehensive framework for modeling QoS requirements and offerings in UML models. It extends the reference UML 2.0 meta-model mainly by using stereotypes. The current specification was published by OMG (Object Management Group) in May 2006 (OMG, 2003). Originally it has been developed for software engineering of object-oriented systems. This article shows that it is also applicable to service description. UML Profile for QoS uses the following approach for QoS description. It describes a QoS model specific to the respective domain separately from the actual elements to be annotated. Then in the actual UML model, the elements can be annotated using terms defined in the QoS model.

There are several reasons for choosing UML Profile for QoS for the extension of OWL-S. Firstly, it comes with its own general catalog of QoS characteristics which is not domain- or project-specific. Secondly, it can be well integrated with business process modeling which is part of the Web services matching problem. Thirdly, compared to other specifications, UML Profile for QoS is quite mature and has been accepted by OMG as a standard. Its definition goes back to a thesis by J. Aagedal published in 2001 where a lot of other QoS-related work has been considered (Aagedal, 2001).

4.4.3 Approaches to Semantic Service Description, Discovery, and Selection

Roy Grønmo and Michael C. Jaeger propose a methodology for Web service composition using QoS optimization (Grønmo & Jaeger, 2005). The main focus of their article is on a matchmaking algorithm that uses QoS requirements and offerings for achieving better results. For both, they use UML Profile for QoS. Other than in this article, they use a link from the WSDL operations to a document describing the QoS offerings.

Dobson (2004) proposes to have functional as well as non-functional specifications in separate repositories. By contrast, we recommend to use a single repository, since we do not see the necessity that a separate organization specifies the QoS characteristics. In fact, the functional and non-functional properties should be guaranteed together either by the organization itself or a third party. The third party could then be a trusted entity that is responsible for monitoring service levels or even for delivering the service levels itself.

Maximilien & Singh (2004) describe a framework and ontology for dynamic Web services selection. It uses an agent-based system to support dynamic service selection and QoS ontologies for describing the non-functional characteristics. Although the approach covers QoS very extensively and comes with a realistic example, it has shortcomings. It uses its own service ontology which makes it proprietary. Also, semantic description of service lifecycle information and functional service description is not explicitly covered by the approach.

WS-QoS is a framework that allows the definition of QoS requirements as well as offerings for Web services and provides an infrastructure for managing those QoS-aware Web services. WS-QoS is based on a WS-QoS XML schema and can be extended. Although it is compatible with UDDI and WSDL by using their extension mechanisms, it is a proprietary approach when it comes to the QoS specification (Tian, 2005).

In (M Klein & König-Ries, 2003), Klein and König-Ries present a process and a tool for describing services based on DAML-S. A layered set of ontologies is used and instantiated in a specific service description with the tool. The service description does not specifically deal with service management requirements. In (Michael Klein, König-Ries, & Müssig, 2005), Klein, König-Ries, and Müssig develop an alternative service description language, called DIANE Service Description (DSD) that implements additional requirements that are not covered by OWL-S and WSMO. However, in this article we want to rely on current standards and existing tools as much as possible.

Matching, i.e. service searching, ranking, and selection, is an interesting application of semantically enriched service description. A lot of work is going on in this area. Apart from functional information also the non-functional information is important to be considered as the already mentioned sources show (Grønmo & Jaeger, 2005;

Maximilien & Singh, 2004). However, functional matching is usually the first step to find appropriate services. The recently published OWLS-MX matcher uses a hybrid approach, combining logic-based reasoning and approximate semantic matching, in particular content-based information retrieval techniques for the input and output parameters specified in the service profile of OWL-S (Klusch, Fries, & Sycara, 2006).

4.5 Extension of OWL-S

The following section describes the proposed extension to OWL-S with respect to service lifecycle management and QoS.

4.5.1 Extension for Service Lifecycle Management

Extension of OWL-S happens in the *ServiceProfile*, one of the four classes OWL-S uses. It is targeted at describing functional and non-functional aspects for service discovery. For the functional description *Parameter*, *Input*, *Output*, *Condition*, *Result*, and *Process* are used. The first five refer to the process description in *ServiceModel*. For the non-functional description the following properties/classes are interesting: *serviceClassification*, *serviceProduct*, *serviceName*, *textDescription*, *ServiceCategory*, and *ServiceParameter*. The first five can be used for the requirements mentioned as they are. The Web service can be classified using *serviceClassification* (mapping to an OWL ontology of services, e.g. NAICS – North American Industrial Classification System), *serviceProduct* (mapping to an OWL ontology of products, e.g. UN-SPSC – United Nations Standard Product and Services Classification), as well as *ServiceCategory* (mapping to taxonomies potentially outside of OWL or OWL-S). A semantic name can be given to a service using *serviceName*. Free text descriptions can be represented with *textDescription*.

Especially important for the extension is *ServiceParameter*. With this element the remaining additional service lifecycle characteristics are defined (Table 15). Future extensions also can be realized using *ServiceParameter*.

Related Service Description in SOAs

Table 15: Defined elements for service lifecycle management

Service lifecycle parameter	Explanation	
Properties / Subclasses	Data type	Explanation
<i>ServiceVersion</i>	Versioning Information	
<i>VersionName</i>	String	Version Name described as literal
<i>VersionNumber</i>	Float	Version number x.x
<i>ServiceLifecycleStatus</i>	Lifecycle status of the service component	
<i>LifecycleStatus</i> (subclass of <i>owl:Thing</i>)	(Enumerated instances)	Enumerated instances: "Planned", "Design", "Test", "Pilot", "Active_intensive_maintenance", "Active_regular_maintenance", "Sunsetting_candidate", "Sunsetting_in_progress", "Sunsetted"
<i>ServiceProvider</i>	Service Provider information	
<i>ProviderLink</i>	anyURI	anyURI Link to external information (name, address, contacts, credentials, etc.) in provider database
<i>ServiceInfrastructure</i>	Infrastructure the service runs on	
<i>ServerID</i>	anyURI	List of server IDs the service runs on
<i>ResourceID</i>	anyURI	List of resource IDs the service uses
<i>SourceCodeLink</i>	Link to source code in code repository	
<i>SourceCode</i>	anyURI	Link to source code

Table 15: Defined elements for service lifecycle management

Service lifecycle parameter	Explanation	
Properties / Subclasses	Data type	Explanation
<i>ServiceResponsibility</i>	Responsibility for service from business and technical perspective	
<i>BizResponsibility</i>	anyURI	Link to organization/person with business responsibility
<i>TechResponsibility</i>	anyURI	Link to organization/person with technical responsibility
<i>BusinessDescription</i>	Information about business background	
<i>BizDescription</i>	String	Textual description of business background
<i>BizInfLink</i>	anyURI	Link to further information resources
<i>ServicePricing</i>	Pricing information	
<i>PricingModelQ1</i>	anyURI	Link to pricing model for QoS level 1, e.g. "Gold"
...
<i>PricingModelQ5</i>	anyURI	Link to pricing model for QoS level 5

ServiceParameter consists of the *serviceParameterName*, the actual name of the parameter, defined as literal or URI, and *sParameter* a link to the value within an OWL ontology. Figure 32 shows the definition of *ServiceVersion* in OWL-S as an example. *VersionName* and *VersionNumber* are defined as datatype properties (type *xsd:string* and *xsd:float*) of the class *ServiceVersionInfo* (subclass of *owl:Thing*). Figure 33 shows the *ServiceVersion* information in OWL-S in a service description for a logistics Web service. *CalculateRoute*. *ServiceVersion_10* and *ServiceVersionInfo_11* are instances that contain the actual version information “Snake” and “5.1”.

```
<owl:Class rdf:ID="ServiceVersion">
  <rdfs:subClassOf rdf:resource=
    "http://www.daml.org/services/owl-s/1.2/
    Profile.owl#ServiceParameter"/>
</owl:Class>
<owl:Class rdf:ID="ServiceVersionInfo"/>
/owl:Class>
<owl:DatatypeProperty rdf:ID="VersionName">
  <rdfs:domain rdf:resource="#ServiceVersionInfo"/>
  <rdfs:range rdf:resource="http://www.w3.org/
    2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="VersionNumber">
  <rdfs:domain rdf:resource="#ServiceVersionInfo"/>
  <rdfs:range rdf:resource="http://www.w3.org/
    2001/XMLSchema#float"/>
</owl:DatatypeProperty>
```

Figure 32: Definition of *ServiceVersion* in OWL-S

```

<ServiceVersion rdf:ID="ServiceVersion_10">
  <profile:sParameter>
    <ServiceVersionInfo rdf:ID="ServiceVersionInfo_11">
      <VersionName rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string" >
        Snake</VersionName>
      <VersionNumber rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#float"
        >5.1</VersionNumber>
    </ServiceVersionInfo>
  </profile:sParameter>
  <profile:serviceParameterName rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#string"
    >ServiceVersion</profile:serviceParameterName>
</ServiceVersion>
<profile:Profile rdf:ID="CalculateRoute_Profile">
  <profile:serviceParameter rdf:resource="#ServiceVersion_10"/>
  [...]
</profile:Profile>

```

Figure 33: Instance of a service description for CalculateRoute with details for ServiceVersion

4.5.2 Extension for QoS with UML Profile for QoS Description

Section 4.3.3 gives a flavour of the level of complexity needed when describing QoS offerings. It shows that a comprehensive and extensible QoS framework that builds on extensive experience needs to be leveraged. UML Profile for QoS is such a framework that suffices the requirements. Hence we propose to use UML Profile for QoS together with OWL-S to bring QoS functionality to Web services description.

The QoS model does not have to be defined in OWL-S. Its definition remains in UML and can be reused for other services and systems. This is very much in line with the idea of using the same QoS notation on the business process side as well as on the service description side to facilitate service level negotiation. The stereotypes *QoS Characteristic* and *QoS Dimension* are used in the QoS model to specify respectively quantify aspects of QoS. It is possible to use statistical values (maximum value, minimum value, range, mean, variance, standard deviation, percentile, frequency, moment, and distribution) as well as to express preferences about the direction when comparing or optimizing parameters (increasing or decreasing).

 Related Service Description in SOAs

For annotating the elements with QoS requirements and offerings, UML Profile for QoS uses three types of constraints: *QoS Required*, *QoS Offered*, and *QoS Contract*. *QoS Required* and *QoS Offered* describe required and offered limitations of *QoS Dimensions* for annotated elements, either by listing the allowed elements or by stating the limits. *QoS Contract* can be used for agreed limitations. Different QoS levels supported by a system, which can be used in SLAs, can be defined with *QoS Level*.

OCL (Object Constraint Language) expressions are used in the QoS statements. This enables rich expressions as those mentioned in section 4.3.3. The respective *QoS Characteristic* is indicated in the annotation statement via *context*. An example *QoS Offered* statement in OCL is shown below: “From Monday to Friday 8:00am to 8:00pm, the response time can be guaranteed to be below 10 ms.”

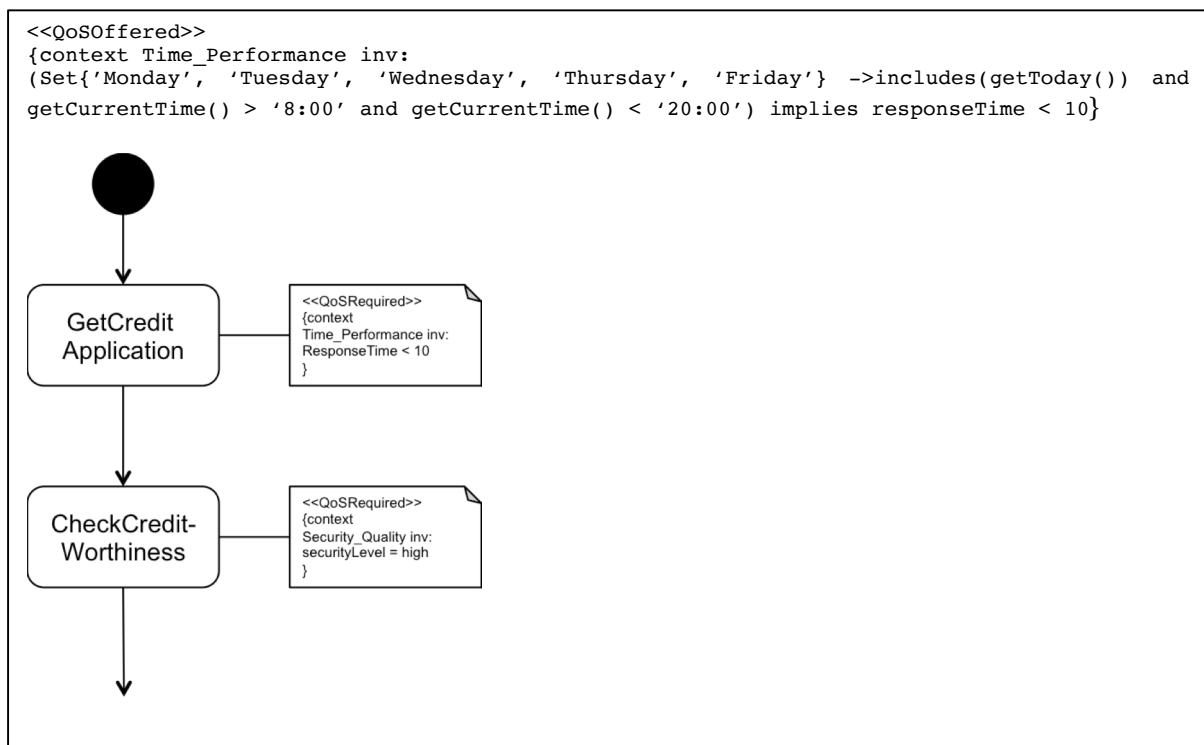


Figure 34: Example QoS requirements in a UML Activity diagram

Introducing such QoS annotations into the OWL-S service descriptions can simply be done by adding *QoSCharacteristics* as a new *ServiceParameter* in *ServiceProfile* and *QoSStatement* as a subclass of *owl:Thing*. *QoSStatement* has the datatype property *statement* of the type string. This field contains the QoS constraints in OCL of the element to be annotated. Figure 34 shows example QoS requirements on the service requestor side in a UML Activity diagram. *responseTime* of *GetCreditService* is required to be lower than 10 ms. Figure 35 shows the corresponding QoS offering in the service description of *GetCreditService* that would be a match during service matching.

```

<profile:Profile
  rdf:ID="GetCreditService_Profile">
  <profile:serviceParameter>
    <QoSCharacteristics rdf:ID="QoSCharacteristics_14">
      <profile:sParameter>
        <QoSStatement rdf:ID= "QoSStatement_15">
          <Statement rdf:datatype="http://
            www.w3.org/2001/XMLSchema#string">
            &lt;&lt;QoSOffered>> {context Time_Performance
              inv: responseTime &lt; 8}</Statement>
          </QoSStatement>
        </profile:sParameter>
      <profile:serviceParameterName rdf:datatype=
        http://www.w3.org/2001/XMLSchema#string>
        QoSCharacteristics</profile:serviceParameterName>
    </QoSCharacteristics>
  </profile:serviceParameter>
  [...]
</profile:Profile>

```

Figure 35: QoS offering for GetCreditService in the service description

4.6 Service Management Prototype

4.6.1 Overview – Architecture and Functionality

The first version of the prototype is a combination of self-developed systems and available open source tools. It is realized as a web application and contains a web browser-driven user interface and two service repositories, one for the standard UDDI publishing and discovery, and one for the semantic search. Two repositories are necessary because the OWL-S-based repository is not UDDI standard compliant, while UDDI as the current standard for service repositories does not offer semantic support. The UDDI registry can be filled automatically with the information from the OWL-S repository. In order to make that possible, a mapping for many of the repositories' elements has been defined.

Figure 36 gives an overview of the prototype's architecture which is structured in 3 layers. The first layer is the web client and client application. It contains the user interface as a web browser application. Via this web-based front-end the user has access to the functionality described in the next section. User authentication functionality as well as storing the account information in a database is implemented here. The client accesses the UDDI and OWL-S repository on a web application server via SOAP, the standardized XML-based message exchange format for Web services. The UDDI repository is based on jUDDI as persistence layer. The OWL-S repository builds on Jena, a semantic web service framework, for the semantic support. Jena facilitates the usage of internal and external reasoners and access to the database via RDQL (Resource Description Framework Query Language). The prototype uses it for interfacing with the database where the semantic description is stored and for performing several operations on the ontology database, in this case MySQL. The prototype itself is written in Java. It uses RMI (Remote Method Invocation) for communication between the Java components.

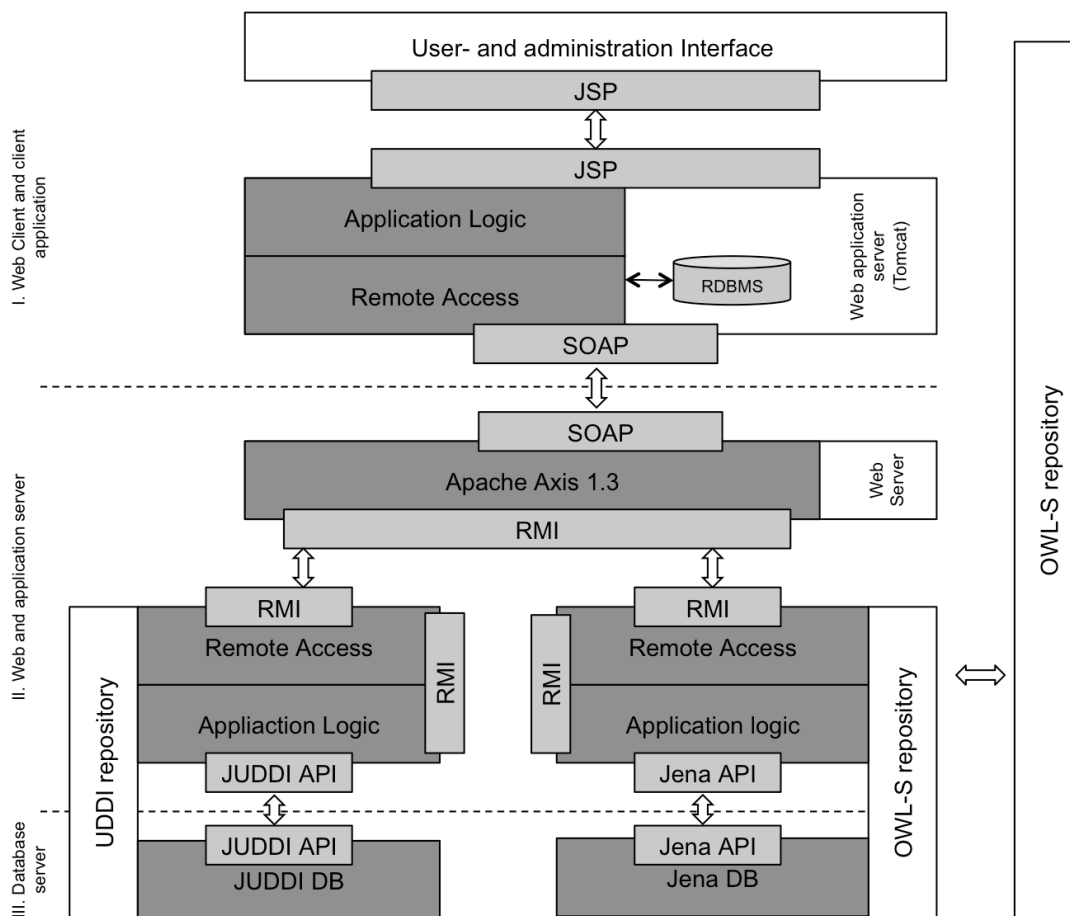


Figure 36: Overview of service management prototype

Apart from the self-written parts, the prototype uses the readily available packages Protégé, Protégé-OWL, and OWL-S Editor. Protégé is a free and open source ontology editor from Stanford University (Stanford Medical Informatics, 2006). Protégé with Protégé-OWL, a plug-in for defining ontologies in OWL, also from Stanford University⁵, is used for the taxonomy definition. OWL-S Editor is a Protégé plug-in developed at SRI International⁶. It helps to define services in OWL-S by making available the OWL-S ontology with its predefined elements and a special view on the service, profile, grounding, and process instances.

⁵ available at <http://protege.stanford.edu>

⁶ available at www.softcomplex.com/products/tigra_tree_menu

4.6.2 Functions and Methodology of the Prototype

The first version of the prototype supports the following tasks as a basis for the mentioned service management responsibilities: taxonomy/ontology definition, service description, semantic annotation, service registration, service discovery, service review, and user access control.

4.6.2.1 Taxonomy/Ontology Definition

The mentioned additions to the OWL-S ontology can be made with the OWL Editor adding new *ServiceParameter* and *owl:Thing* subclasses. Later, service descriptions and ontology extensions can be done using the OWL file. Also, a taxonomy for the service category field and input/output parameters can be developed with Protégé OWL. The generic way of defining/redefining the service taxonomy is an important feature. It is a matter of fact that there is no stable service description in complex environments.

4.6.2.2 Service Description and Semantic Annotations

Service description and semantic annotations are done with the OWL-S editor by loading the OWL file that contains the ontology extended by the above mentioned elements. It is possible to import existing WSDL descriptions. Once the extended OWL-S ontology is loaded, the services can be described. For specifying a parameter for a service, the predefined *ServiceParameter* has to be used. There are two ways of doing this. If the parameter contains listed elements, e.g. *ServiceLifecycleStatus*, a link to an existing instance can be used. If the parameter contains an element with free content like a number or a text field (e.g. *ServiceVersion*), a new parameter value instance has to be created. Apart from the non-functional elements, it is possible to semantically describe the input/output parameters using normal OWL-S functionality and the service parameter ontology defined.

4.6.2.3 Service Registration

Service registration is done by importing the OWL-S service description into the prototype and its database. This is necessary after each change to it. The prototype can then perform the search activities laid out in the next section.

4.6.2.4 Service Discovery and Review

The main functionality of the prototype is search functionality across the services registered and described. There are several possibilities for performing searches using the additional semantic information:

1. Simple queries – searching for services, input/output parameters, taxonomy expressions, etc. using the full names of these elements
2. Semantic queries for services using their input and output parameters
3. Semantic queries for services that match other services' input or output parameters
4. Semantic queries for services using taxonomy elements
5. Semantic queries using the other additional parameters such as *ServiceVersion*, *ServiceResponsibility*, and *ServiceLifecycleStatus*
6. Taxonomy tree search – services that belong to one taxonomy can be found by navigating through a simple taxonomy tree or a hyperbolic graph

Number 3 refers to a simple matching functionality that can be used for service orchestration and will be extended in the future. To increase the flexibility of the search, it is possible to use the outcome of one search run as the basis for another search.

4.6.2.5 User Access Control

For service management in complex environments it is absolutely necessary to support role-specific views combined with access rights management. The numerous services are the core of an IT system of an enterprise. Therefore they need to be protected against malicious attacks as well as erroneous and uncoordinated activities of careless or unaware users. Hiding unnecessary information improves usability, reduces the number of errors, and is sometimes a must when it comes to confidentiality. The prototype's user authentication module controls the activities of individual users according to the rights associated with their roles. An "Administrator" can add new accounts and associate them to a role. "Users" are only allowed to search and browse through the service repository. "Developers" can in addition perform detailed search operations. The "Architect" is also allowed to register and delete services in the repository.

4.7 Importance for Matching, SLA Negotiation and Enforcement

Currently, the search needs to be done manually. Having visibility about all services implemented and the possibility of managing meta-information of the services centrally and thus in a consistent way is a big advantage and a precondition for the success of an SOA. However, if the IT systems based on the services get bigger and bigger and the number of services is expanding, a process that includes more automated support is necessary. The semantic description of input and output parameters and non-functional characteristics is a prerequisite for that. Only if service requestor and service provider refer to the same ontological concepts, the service matching module can “understand” them. That is why the additional effort of managing the semantic metadata is justified. A common way of performing the matching or SLA negotiation is a two-step approach as proposed by METEOR-S (Grønmo & Jaeger, 2005), or in “Semantic WS-Agreement Partner Selection” (Oldham, Verma, Sheth, & Hakimpur, 2006). The first step performs functional matching. We suggest a hybrid semantic matching based on input and output parameters, e.g. by using OWLS-MX. In addition we propose to use the service category. Due to the semantic information not only exact matches of parameters and taxonomies are found but also parameters that stand in a class-sub-class relationship, e.g. car – convertible. The second matching step performs the non-functional matching using particularly the QoS-related information. Constraints about the QoS-characteristics on the service consumer side (*QoS Required*) are compared with the QoS-offerings specified in the service description (*QoS Offered*). The outcome is a ranking of the existing services that perform the desired functionality according to how well they meet the QoS requirements. Once a service is chosen, an SLA, a formal specification of the agreement between service consumer and service requestor (inter- or intra- organizational) can be specified.

It is planned to extend the prototype’s service matching functionality and also to introduce an SLA specification, and SLA management module. According to a service request with a set of semantically enriched functional and non-functional information this module will discover existing services in the repository, provide their WSDLs and specify the SLA in a nearly fully automated way. The format for the SLA will be WSLA or WS-Agreement. The machine-readable SLA is a good basis for automated SLA-enforcement and monitoring during run-time. In case of problems, the person responsible can find the respective service in the registry and has access to infor-

mation, e.g. contact details, infrastructure the service runs on. Matching and SLA specification functionality will ease the life of system developers as well as SLA authors/enforcers. It will also foster reuse, one of the goals of SOAs.

4.8 Conclusion and Outlook

As SOAs will be very complex from an IT service management point of view, in order to deliver their full value, automated tool support is necessary. Semantic description of non-functional service characteristics is one important prerequisite for that.

The contribution of the presented work is a practical approach to service description and discovery that is extensible regarding additional future requirements. The article shows that it is possible to build a semantically enriched service repository with OWL-S that supports several tasks that are the basis for higher level service management activities. With the approach, it is possible to describe – along with the functional characteristics – the non-functional characteristics with respect to service management (service lifecycle management and QoS) in a single OWL-S-based repository. The approach is extendable with respect to changes of the used taxonomy as well as the elements used for service description. At the same time it is a compatible upgrade of the existing Web services description standards. Besides the presented approach, the article also gave an overview over relevant standards and related work in the area of non-functional service description.

The prototype will be extended to support in a better integrated way service description functionality. Extensions for automated service discovery, SLA specification, and SLA management are planned.

5 Beitrag V: Cloud Computing and the Impact on Enterprise IT

5.1 Abstract

This paper examines the impact of the cloud computing paradigm on enterprise IT structures. The first section of the paper defines the term cloud computing with the relevant aspects that are relevant for this paper. In the second chapter the paper shows the potential of cloud computing for the enterprise IT. In the last section it will be briefly reflected how the future development of infrastructure technologies will create more potential for enterprise IT cloud cases in the future.

5.2 Introduction

Cloud computing is one of the most discussed issues in enterprise IT. Despite this discussion the use cases in cloud computing are mostly derived from a non-enterprise environment. As cloud computing was first used to provide IT infrastructure to consumer web applications the original cloud services were developed to meet their requirements. Enterprises have different requirements for their use cases.

Therefore this paper examines the demand of enterprise IT in the cloud for the segment of infrastructure as a service (IaaS) cloud. In the course of this analysis, two use cases are introduced and analyzed.

Some of the challenges that were found during the analysis cannot be sufficiently addressed by today's technologies. Two of those issues are described in the last section, showing the demand of new infrastructure technologies to further develop cloud computing for enterprises.

5.3 Cloud Computing

Cloud Computing is on-demand access to virtualized IT resources that are sourced inside or outside of a data center, scalable, shared by others, simple to use, paid for via subscription or as you go, and transparently accessible over the network. There is no common definition of clouds at the moment, neither from academic nor industry communities, but in general it is described as the provisioning of services at different layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) (see (Youseff, Butrico, & Da Silva, 2008), (Buyya et al., 2009), (Briscoe & Marinos, 2009)).

The lowest layer of clouds is represented by IaaS, where the object provided is a “server instance” with a network connection or other services like (virtual) storage devices. The customer obtains infrastructure components, which can be used for executing virtual machines.

On a layer above, PaaS offerings basically provide an execution platform or interpreter service for user-defined application codes. PaaS contains pure runtime environments as well as programmable interfaces that allow the usage of pre-defined application services.

The upmost layer in this three layer model is SaaS, which describes the delivery of specific applications comparable to the ASP (Application Service Provisioning) model. These applications can be generic applications (e.g. webmail) as also specific business or research applications (CRM or GRID services).

In addition, Cloud Services can be divided in “internal” and “external”. Whereas the external Cloud Services are offered as multi-tenant applications over the internet, internal services are only available in the intranet of a specific organization.

The important aspect in all three cases is the service delivery to the customer, which is always performed in a transparent manner independent of the actual network location. The technical key characteristic of all these concepts is the usage of X86 virtualization, which allows the isolation of workloads independent of the single application on standard platforms. In combination with additional non-technical aspects, like on-demand usage and payment, Cloud Computing is currently considered as a new way of large scale computing.

Nevertheless today’s Cloud Computing platforms are limited in functionality, often to predefined sets of applications and in terms of access capabilities. This leads to the rise of many small platforms for specific use cases, and following the diversity, Cloud Computing faces many different obstacles on the technical, legal and operational level. This creates the risk of a new complexity.

The specific challenge for enterprise IT customers to manage the possibilities of the cloud are analyzed in the remainder of this paper.

5.4 Enterprise IT in the Cloud – The Challenges

Understanding the challenges of enterprise IT in the cloud requires looking at the traditional cloud use cases first. Cloud computing was used primarily for consumer

oriented web applications. For these applications cloud computing could not only provide new technological capabilities to scale applications quicker, but it offered also the ability to gradually increase the costs while adding new users. This is relevant in the typically user driven revenue models of these applications and allowed many services to be commercially successful.

This main use case in cloud computing defined technology and setup of the offerings. When enterprises began to analyze how the cloud can optimize their IT landscapes a couple of challenges arose.

Time to market. A very important aspect of cloud computing for enterprises is the ability to reduce the time to market. Bringing services to the market or internal users in less time is crucial for business success.

Load optimization. Many enterprises have already an optimized datacenter environment. Compared to public cloud environments where a huge amount of independent customers and applications allow a statistical load balancing and thus load optimization, enterprise environments cannot deliver this. Enterprises have a smaller amount of applications, each having a larger size. Also many applications are interconnected. Because of this peak loads can occur at the same time.

Legacy application. Enterprises have complex landscapes of existing applications. When planning the migration of services into the cloud these existing applications must be taken into consideration. In particular technical aspects like network requirements, scaling technologies and platform compatibility need to be inspected carefully.

Security zones. Security is a huge topic in the cloud – as well for consumer-oriented applications as also for enterprise IT use cases. The difference between the two use cases is that consumer applications have a single security zone as all customers are treated equally secure. Enterprises instead need to differentiate these security aspects in more complex ways. Internal users, partners, suppliers and customers and visitors are user groups that need access to different levels of information and thus need to be separated in different security zones. This requirement often led to the introduction of more complex network schemes to control access at this level. These cannot easily be transferred to the cloud.

SLAs and consistency. SLAs in the cloud are different than those of usual enterprise hosting contracts. Cloud SLAs refer to the availability of the entire platform, but

not to the availability of a single instance running in the platform. Therefore single instances can fail more often without breaching the SLA commitment. For an enterprise, which is used to single instance SLAs this has a twofold impact. On the one hand the enterprise must implement failover mechanisms to manage cases where a single instance fails, but the overall platform remains functioning. On the other hand a user must manage consistency issues when a single instance fails. Data that is stored within the instance cannot be recovered.

Data center processes. Enterprises have clear rules and processes how the data center operation is done, which are required by legal compliance as well as service quality management. These processes need to be adjusted when self-service principles are being introduced. In order to meet quality and compliance goals cloud computing customers need to introduce new processes as well as change existing ones.

Despite these challenges cloud computing is being seen as one of the most important drivers for the enterprise IT in the upcoming years. Similarly to the movement from the mainframe to client server architectures the cloud computing paradigm is seen as a new movement towards centralized IT services. The benefits of cloud computing can mainly be seen in the simplification and thus decrease of time to market and the financial benefits. These will be explained in more detail in the next section using two use cases.

5.5 Enterprise IT in the Cloud – The Benefits

Enterprises using cloud computing can benefit in different ways from the cloud depending on the requirements of the specific use case. These should be demonstrated by two different use cases in this section.

5.5.1 Use Case 1: Solution Introduction in Enterprises

Internal communication and effective knowledge management is one of the most challenging tasks of modern knowledge driven enterprises.

Innovative software solutions are recently introduced into enterprise IT landscapes to improve communication and knowledge exchange. An example for these solutions is “enterprise microblogging”.

Solutions in this area are currently being adopted by very early adopters, as software products are available only since less than a year. These early adopters usually evaluate the solution by piloting it in defined areas of the company. When introducing

these services often the usage increases strongly already during the pilot phase, as the service is getting part of the daily work processes. Also this daily usage changes the requirements regarding the service levels, as downtimes are not tolerable.

Confronted with these challenges cloud computing can improve the introduction of a service in the following aspects:

1. Time to pilot: in the cloud a customer can create a pilot environment within a couple of hours instead of days. The infrastructure can be provisioned within minutes and the configuration can be conducted also within very short time.
2. Scale: with the user adoption the growing platform demand can be covered by dynamic scaling of the infrastructure.
3. Security / SLAs: in a cloud platform failover mechanisms can be implemented in a simplified manner. This leverages the cloud capabilities to meet the requirements of availability and consistency with the growing usage.
4. Costs: the resource allocation along the real demand not only saves money and thus reduces the total cost of ownership (TCO), but also the cost structure is optimized. Instead of huge amounts of capital expenses, when using the cloud nearly all infrastructure costs are flexible operating expenses. The overall cost improvements are depicted in Figure 37. This figure shows the improvements of an installation where 10 servers have been replaced by instances in a compute cloud.

5.5.2 Use Case 2: Peak Pooling by Design

Also in the production phase of an application cloud computing can deliver benefits. In this use case two independent enterprise applications are being grouped together for the analysis of their respective load profiles. As discussed in section 5.4 the load optimization on a statistical basis is not applicable in the enterprise case. Therefore a load profile analysis can define groups of applications that will not require peak resources at the same time, neither in scheduled load situations nor in unplanned ones. Once these applications are identified, the cloud computing technologies enable the automated scaling of workloads in the peak time and the downscaling after the peak. To gain the commercial benefit the two applications are separated in a base workload part and a peak workload part.

The base workload is being executed on an unshared platform, whereas the peak workload runs in the shared peak platform.

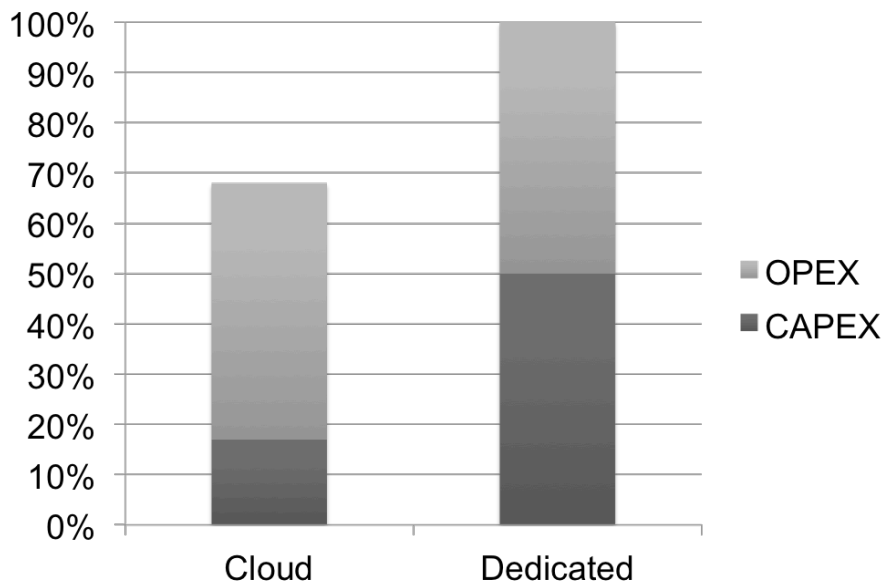


Figure 37: Cost comparison for a small cloud use case

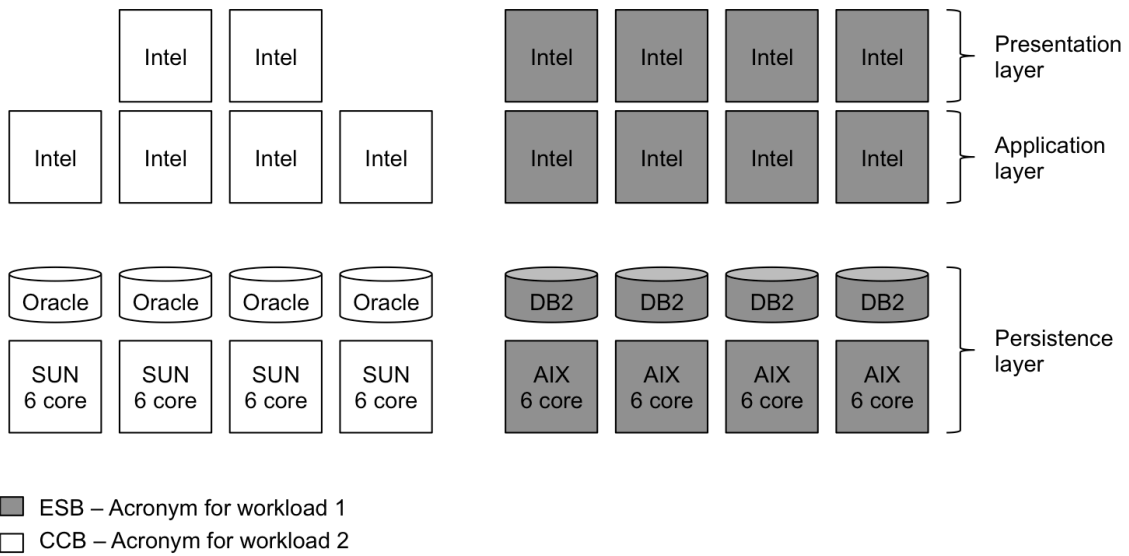


Figure 38: As-is situation for peak pooling

The overall savings in the presented use case are around 30% of the total cost of Ownership (TCO) of the two applications.

To realize these savings several factors are taken into consideration:

1. Reducing hardware cost: This reduction comes from the overall size reduction, but in addition significant savings derive from the usage of standard machines instead of expensive multicore environments.
2. Reducing operational costs: The constant resizing of the platform is happening automatically by the cloud technologies, which decreases the operational costs extremely.
3. Homogenization of the infrastructure: When peak pooling is implemented, all layers in the pool are running on a single homogenous platform. This simplifies a couple of processes and reduces the per node costs.

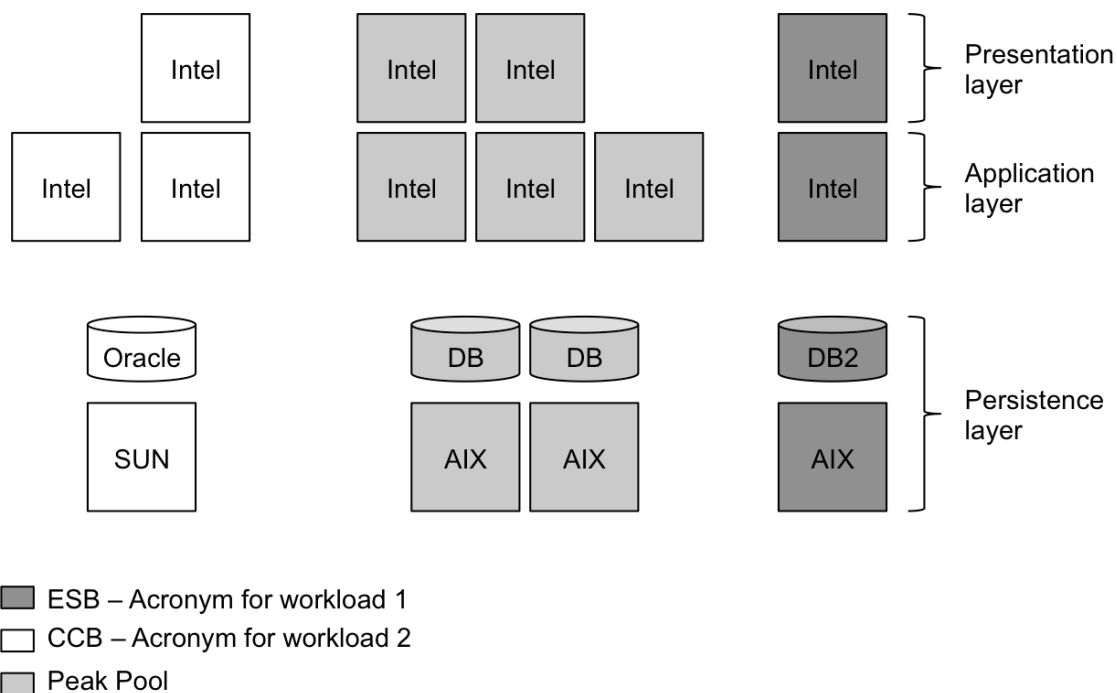


Figure 39: To-be situation

This use case demonstrates the potential of cloud computing in enterprises, but also shows the challenges as described in section 5.4.

- The two applications are legacy applications. The specific challenge with the shown applications is to allow the scalability also in the data layer.
- The applications are implemented on host based SLAs. Guaranteeing consistency in a cloud environment requires adoption of the applications.

- The data center processes need to be adjusted significantly. In the cloud the application requests and releases itself the required infrastructure resources instead of administrators managing this. The administrators have to manage and maintain the cloud platform instead. Only by implementing these procedural changes the benefits can be realized.

5.6 Next Generation Infrastructures and the Cloud

Although – as shown in the previous section – cloud computing can deliver huge benefits to the enterprise IT, some of the challenges described in section 5.4 cannot be sufficiently addressed. In this section two challenges will be briefly reviewed: SLA / consistency issues and security zones.

5.6.1 SLA / Consistency

SLAs and consistency are an inherent challenge for cloud computing platforms. On the one hand side creating high SLAs on a per host base does not comply with the business model of cloud computing, on the other hand consistency on a low SLA infrastructure requires constant synchronization. This leads to the demand of having a cloud of multiple locations with a high performance network between these. Only then high volume replications can be performed in short enough time to guarantee consistency across the sites. When consistency is achievable in such a distributed environment also very high SLAs can be guaranteed.

In this scenario a high performance but affordable network with quality of service management functionality is required to be compliant with enterprise demand.

5.6.2 Security Zones

Network management is one of the most complex tasks in cloud computing. As described in section 5.4 most cloud computing network structures do not comply with enterprise IT demands. Having a more flexible and secure network structure is a key factor for cloud success in the enterprise IT. When workloads are being resized and moved between locations automatically, also network structures must be able to connect them in an automated and transparent way.

5.7 Conclusion

Cloud Computing has a huge potential for enterprise IT. Nevertheless to realize the full potential of the cloud, enterprises have to do quite substantial adjustments to their internal structures and applications.

The biggest challenge in realizing the potential of flexibility of the cloud is the network structure. This challenge is caused by bandwidth shortage of many enterprises as well as structural network setups that cannot be adjusted in the dynamic way required by the cloud. New technologies in this area are required to fulfill as well security demands of the enterprise as also the dynamics of the cloud.

Referenzen

- Agedal, J. (2001). *Quality of service support in development of distributed systems*. University of Oslo.
- Aboulnaga, A., Salem, K., Soror, A. A., Minhas, U. F., Kokosielis, P., Kamath, S., ... Others. (2009). Special Issue on Data Management on Cloud Computing Platforms. (D. Lomet, B. Chin Ooi, & S. Parthasarathy, Hrsg.) *Data Engineering*. IEEE Computer Society.
- Agarwal, R., Choi, J., Ramamurthy, R., Selen, W., & Selim, H. M. (2012). Service-Oriented Architecture as A Driver of Service Innovation in Newly Emerging Service Systems An Exploratory View. *International Journal of Information Processing and Management*, 3(April), 85–97. <http://doi.org/10.4156/ijipm.vol3.issue2.10>
- Aggarwal, R., & Verma, K. (2004). Constraint driven web service composition in METEOR-S. In *IEEE International Conference on Services Computing* (S. 23–30). Shanghai.
- Aier, S. (2006). How Clustering Enterprise Architectures helps to Design Service Oriented Architectures. In *2006 IEEE International Conference on Services Computing (SCC'06)* (S. 269–272). IEEE. <http://doi.org/10.1109/SCC.2006.52>
- Aier, S., Fischer, C., Gleichauf, B., Riege, C., Schelp, J., & Winter, R. (2009). *Management von Integrationsprojekten*. (R. Winter, Hrsg.). Heidelberg: Springer.
- Aier, S., & Schelp, J. (2008). EAI und SOA - Was bleibt nach dem Hype? In *Multikonferenz Wirtschaftsinformatik* (S. 1469–1480).
- Aier, S., & Schönherr, M. (2006). Model Driven Service Domain Analysis. In M. Schönherr & M. Ahrens (Hrsg.), *Preproceedings of Service Oriented Modeling, 1st International Workshop on Service Oriented Modeling* (S. 98–109). Chicago: Gito Verlag.
- Aier, S., & Winter, R. (2009). Virtuelle Entkopplung von fachlichen und IT-Strukturen für das IT/Business Alignment–Grundlagen, Architekturgestaltung und Umsetzung am Beispiel der Domänenbildung. *Wirtschaftsinformatik*, 51(2), 175–191.
- Ait Ali Slimane, A., Pinheiro, M. K., & Souveyet, C. (2009). Goal Reasoning for Quality Elicitation in the ISOA approach. In *Third International Conference on Research Challenges in Information Science* (S. 39–48). Fez.
- Akkiraju, R., Farrell, J., & Miller, J. A. (2005). *Web Service Semantics - WSDL-S*. Georgia.
- Alizadeh, K., & Mohsenzadeh, M. (2011). *Mapping service concept and enterprise ontology in service identification*. *7th International Conference on Networked Computing (INC)* (S. 22–27).
- Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). *Web Services*. Berlin/Heidelberg: Springer.
- Alter, S. (2013). Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future. *Journal of the Association for Information Systems*, 14(2), 72–121.
- Andersson, B., Johannesson, P., & Zdravkovic, J. (2009). Aligning goals and services through goal and business modelling. *Information systems and e-Business Management*, 7(2), 143–169.
- Andreas, F., Krivelevich, M., & Goerdt, J. (2005). Recognizing More Unsatisfiable Random k-SAT Instances Efficiently. *SIAM Journal on Computing*, 35(2), 408.
- Andrews, T., Curbera, F., Dholakia, H., & Golland, Y. (2003). *Business Process Execution Language for Web Services - version 1.1*.
- Andrieux, A., Czajkowski, K., Dan, A., & Keahey, K. (2007). Web services agreement specification (WS-Agreement). *Open Grid Forum*.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... Zaharia, M. (2009). *Above the clouds: A Berkeley view of cloud computing* (No. 28). *University of California, Berkeley, Tech. Rep. UCB*. Berkeley.

- Arni-Bloch, N., & Ralyté, J. (2010). MISS: A Metamodel of Information System Service. In G. A. Papadopoulos, W. Wojtkowski, G. Wojtkowski, S. Wrycza, & J. Zupancic (Hrsg.), *Information Systems Development SE - 19* (S. 177–186). Springer US. http://doi.org/10.1007/b137171_19
- Arsanjani, A. (2004). *Service-oriented modeling and architecture. IBM developer works.*
- Arsanjani, A., Ghosh, S., & Allam, A. (2008). SOMA: A method for developing service-oriented solutions. *IBM systems Journal*, 47(3), 377–396.
- Ashish, J., & Saxena, P. (2002). Optimum Control Limits for Employing Statistical Process Control in Software Process. *IEEE Transactions on Software Engineering*, 28(12), 1126–1134.
- Azevedo, L. G., Santoro, F., & Baião, F. (2009). A method for service identification from business process models in a SOA approach. In T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, & R. Ukor (Hrsg.), *Enterprise, Business-Process and Information Systems Modeling* (S. 99–112). Berlin/Heidelberg: Springer.
- Baghdadi, Y. (2006). Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing. *Information systems frontiers*, 8(5), 395–410.
- Barbara, P., Philippsen, M., Tichy, W. F., & Unger-Lamprecht, L. (2002). Two Controlled Experiments Assessing the Usefulness of Design Pattern Documentation in Program Maintenance. *IEEE Transactions on Software Engineering*, 28(6), 595–606.
- Bardhan, I. R., Demirkan, H., Kannan, P. K., Kauffman, R. J., & Sougstad, R. (2014, Dezember 8). An Interdisciplinary Perspective on IT Services Management and Service Science. *Journal of Management Information Systems*. Routledge.
- Barnes, M. (2006). Applied SOA: Measuring Business Value. Abgerufen 24. Mai 2007, von http://www.gartner.com/teleconferences/attributes/attr_150040_115.pdf
- Barton, N. (2002). Measuring IT Performance—From Functional Models to Service Models. In A. Brown (Hrsg.), *Ninth European Conference on Information Technology Evaluation* (S. 55–61). Paris.
- Basili, V., & Weiss, D. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, 10(6), 728–738.
- BEA Systems. (2006). Dev2Dev Online: Service-oriented Architecture. Abgerufen 21. Juni 2006, von <http://dev2dev.bea.com/soa/>
- Beck, R., Tönker, M., & Dernbecher, S. (2013). Design and evaluation of a grid-in-a-cloud solution. In *ECIS 2013 - Proceedings of the 21st European Conference on Information Systems*. Goethe University Frankfurt, Department of E-Finance and Services Science: Association for Information Systems.
- Becker, A., Widjaja, T., & Buxmann, P. (2011). Nutzenpotenziale und Herausforderungen des Einsatzes von Serviceorientierten Architekturen. (German). *Wirtschaftsinformatik - Zeitschrift Der Angewandten Informatik*, 53(4), 187.
- Beims, M. (2012). *IT-Service Management in der Praxis mit ITIL 3: Zielfindung, Methoden, Realisierung* (3. Auflage). München: Carl Hanser.
- Berger, T. (2005). *Konzeption und Management von Service-Level-Agreements für IT-Dienstleistungen*. Technische Universität Darmstadt.
- Bernus, P., & Nemes, L. (1997). The contribution of the generalised enterprise reference architecture to consensus in the area of enterprise integration. *Enterprise Engineering and Integration*.
- Beverungen, D., Knackstedt, R., & Müller, O. (2008). Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung – Eine Konzeptionsmethode und ihre Anwendung am Beispiel des Recyclings elektronischer Geräte. (German). *Wirtschaftsinformatik*, 50(3), 220–234.
- Bianchini, D., Capiello, C., De Antonellis, V., & Pernici, B. (2014). Service identification in interorganizational process design. *IEEE Transactions on Services Computing*, 7(2), 265–278.

- Bianchini, D., & Cappiello, C. (2009). P2S: A methodology to enable inter-organizational process design through web services. In P. van Eck, J. Gordijn, & R. Wieringa (Hrsg.), *Advanced Information Systems Engineering* (S. 334–348). Berlin/Heidelberg: Springer.
- Birkmeier, D. Q., Gehlert, A., Overhage, S., & Schlauderer, S. (2013). Alignment of Business and IT Architectures in the German Federal Government: A Systematic Method to Identify Services from Business Processes. In *2013 46th Hawaii International Conference on System Sciences* (S. 3848–3857). IEEE. <http://doi.org/10.1109/HICSS.2013.77>
- Braun, C., Hafner, M., & Wortmann, F. (2004). *Methodenkonstruktion als wissenschaftlicher Erkenntnisansatz*. St. Gallen.
- Braun, C., & Winter, R. (2007). Integration of IT Service Management into Enterprise Architecture. In *Proceedings of the 2007 ACM Symposium on Applied Computing* (S. 1215–1219).
- Braunwarth, K. S., & Friedl, B. (2010). Towards a financially optimal design of it services. In M. Lacity, S. March, & F. Niederman (Hrsg.), *ICIS 2010 Proceedings - Thirty First International Conference on Information Systems*.
- Brown, P. C. (2007). *Succeeding with SOA: Realizing Business Value Through Total Architecture*. Pearson Education.
- Bub, U. (2003). *Mit integrierter, unternehmensweiter ICT den Umbruch gestalten* (No. 10). Diebold / Detecon Management Report (Bd. 10).
- Buglione, L., & Abran, A. (2000). Balanced Scorecards and GQM: what are the differences. In G. Poels & J. C. Verdún (Hrsg.), *Proceedings of FESMA- AEMES Software Measurement Conference*. Madrid.
- Buyya, R., Ranjan, R., & Calheiros, R. N. (2009). Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. In *Proceedings of the 7th High Performance Computing and Simulation (HPCS 2009) Conference* (S. 11–22). Leipzig.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 17. <http://doi.org/10.1016/j.future.2008.12.001>
- Caetano, A., Silva, A., & Tribolet, J. (2010). Identification of Services through Functional Decomposition of Business Processes. *Business Information Systems*, 47, 144–157.
- Cartlidge, A., Hanna, A., Rudd, C., Macfarlane, I., Windebank, J., & Rance, S. (2007). *An Introductory Overview of ITIL ® V3*. (A. Cartlidge & M. Lillycrop, Hrsg.). The UK Chapter of the itSMF Ltd.
- Chaari, S., Biennier, F., Favrel, J., & Benamar, C. (2007). Towards a Service-Oriented Enterprise based on Business Components Identification. In R. Goncalves, J. Müller, K. Mertins, & M. Zelm (Hrsg.), *Enterprise Interoperability II* (S. 495–506). London: Springer.
- Chappell, D. A. (2004). *Enterprise Service Bus*. Sebastopol, CA O'Reilly 2004.
- Chatziprimou, K., Lano, K., & Zschaler, S. (2013). Towards a Meta-model of the Cloud Computing Resource Landscape. In *MODELSWARD - 1st International Conference on Model-Driven Engineering and Software Development* (S. 111–116). Barcelona.
- Chen, C.-C., Wu, C.-S., & Wu, R. C.-F. (2006). e-Service enhancement priority matrix: The case of an IC foundry company. *Information & Management*, 43, 572–586.
- CIO-Council. (1999). Federal Enterprise Architecture Framework Version 1.1.
- CMMI Team. (2010). CMMI-SVC v 1.3. Abgerufen 13. Februar 2015, von <http://cmmiinstitute.com/resources/cmmi-services-version-13>
- Cokins, G. (2004). *Performance Management: Finding the Missing Pieces (to Close the Intelligence Gap)*. John Wiley & Sons.
- Córdoba, J.-R. (2009). Critical reflection in planning information systems: a contribution from critical systems thinking. *Information Systems Journal*, 19(2), 123–147.

- Dietrich, J. (2008). *Nutzung von Modellierungssprachen und-methodologien standardisierter B2B-Architekturen für die Integration unternehmensinterner Geschäftsprozesse*. GITO Verlag.
- DiMare, J. (2006). *Service-Oriented Architecture: A practical guide to measuring return on that investment*. IBM Institute for Business Value.
- Dobson, G. (2004). Quality of service in service-oriented architectures. *Dependability Infrastructure for Grid Services Project*.
- Dwivedi, V., & Kulkarni, N. (2008). A model driven service identification approach for process centric systems. In *Congress on Services Part II* (S. 65–72). Beijing.
- Edmond, D., OSullivan, J., & Hofstede, A. ter. (2002). What's in a service? Towards accurate description of non-functional service properties. *Distributed and Parallel Databases*, 12, 117–133.
- Enterprise Management Associates. (2006). IBM Tivoli Unified Process (ITUP): Connecting the Dots. Abgerufen 10. Januar 2015, von http://www-07.ibm.com/sg/governance/servicemanagement/downloads/EMA_ITUP.pdf
- Erik, D., Briand, L. C., & Adsholm, W. J. (2008). A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance. *IEEE Transactions on Software Engineering*, 34(3), 407–432.
- Erradi, A., Anand, S., & Kulkarni, N. (2006). SOAF: An architectural framework for service definition and realization. In *IEEE International Conference on Services Computing* (S. 151–158). Chicago, IL: IEEE.
- ESSI WSMO Working Group. (2006). Web Service Modelling Ontology. Abgerufen von <http://wsmo.org>
- Fareghzadeh, N. (2008). Service identification approach to SOA development. *Proceedings of World Academy of Science, Engineering and Technology*, 45, 258–266.
- Fleischmann, A., Schmidt, W., Stary, C., & Augl, M. (2013). Agiles Prozessmanagement mittels Subjektorientierung. (German). *HMD: Praxis der Wirtschaftsinformatik*, 50(2), 64.
- Fokaefs, M., & Stroulia, E. (2013). WSMeta: a meta-model for web services to compare service interfaces. In *Proceedings of the 17th Panhellenic Conference on Informatics* (S. 1–8). Thessaloniki: ACM New York.
- Foody, D. (2005). Getting web service granularity right. Abgerufen 12. Juni 2006, von <http://www.soazone.com/index.php?/archives/11-Getting-web-service-granularity-right.html>
- Frank, U., Strecker, S., Kattenstroth, H., & Heise, D. (2011). *Wissenschaftliche Untersuchung „Integration der COBIT und ITIL Standards mit der RA“*. Essen.
- Franke, U., Johnson, P., & König, J. (2014). An architecture framework for enterprise IT service availability analysis. *Software and Systems Modeling*, 13(4), 1417–1445. <http://doi.org/10.1007/s10270-012-0307-3>
- Frese, E. (2000). *Grundlagen der Organisation: Konzept—Prinzipien — Strukturen* (8. Auflage). Wiesbaden: Gabler.
- Frølund, S., & Koistinen, J. (1998). Quality of Service Specification in Distributed Object Systems Design. *COOTS*.
- Gammalgård, M., Närman, P., Ekstedt, M., & Norström, L. (2006). Business value evaluation of IT systems: developing a functional reference model. In *Proceedings of the the Fourth Conference on Systems Engineering*.
- Gericke, A., & Winter, R. (2009). Entwicklung eines Bezugsrahmens für Konstruktionsforschung und Artefaktkonstruktion in der gestaltungsorientierten Wirtschaftsinformatik. In J. Becker, H. Krcmar, & B. Niehaves (Hrsg.), *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik* (S. 195–210). Heidelberg: Physica Verlag. http://doi.org/10.1007/978-3-7908-2336-3_10
- Gierczak, M., Söllner, M., & Leimeister, J. M. (2012). Untersuchung bestehender Geschäftsmodelle

- etablierter Cloud Anbieter. In *ConLife 2012 Academic Conference*.
- Gleich, R., & Schimpf, T. (1999). Prozessorientiertes Performance Measurement-Ein Konzept der dauerhaften Leistungssteigerung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*.
- Goeken, M., Alter, S., Milicevic, D., & Patas, J. (2009). Metamodelle von Referenzmodellen am Beispiel ITIL-Vorgehen, Nutzen, Anwendung. In *Informatik 2009: Im Focus das Leben, Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI)* (S. 3701–3714). Lübeck.
- Goethert, W., & Fisher, M. (2003). *Deriving enterprise-based measures using the balanced scorecard and goal-driven measurement techniques* (No. CMU/SEI-2003-TN-024).
- Gold, N., Mohan, A., Knight, C., & Munro, M. (2004). Understanding service-oriented software. *Software, IEEE*.
- Gomes, L. (2006). For business-software developers, flexibility can be spelled SOA. *The Wall Street Journal Europe*, 32.
- Gottschlich, J., Hiemer, J., & Hinz, O. (2014). A Cloud Computing Broker Model for IaaS resources. In *ECIS 2014*. Tel-Aviv.
- Grønmo, R., & Jaeger, M. C. (2005). Model-driven methodology for building QoS-optimised Web Service Compositions. *Distributed Applications and Interoperable Systems*, 3543, 68–82.
- Grønmo, R., Jaeger, M. C., & Hoff, H. (2005). Transformations between UML and OWL-S. In A. Hartmann & D. Kreische (Hrsg.), *Model Driven Architecture—Foundations and Applications* (Bd. 3748, S. 269–283). Berlin/Heidelberg: Springer.
- Gu, Q., & Lago, P. (2010). Service identification methods: a systematic literature review. In E. Di Nitto & R. Yahyapour (Hrsg.), *Towards a Service-Based Internet, Third European Conference ServiceWave 2010* (S. 37–50). Ghent, Belgium.
- Haas, H., & Brown, A. (2004). Web services glossary. *W3C Working Group Note (11 February 2004)*.
- Hafner, M., Schelp, J., & Winter, R. (2006). Berücksichtigung des Architekturmanagements in serviceorientierten IT-Managementkonzepten am Beispiel von ITIL. In J. Schelp & R. Winter (Hrsg.), *Integrations-management* (S. 99–121). Berlin/Heidelberg: Springer.
- Hamdaqa, M., Livogiannis, T., & Tahvildari, L. (2011). A Reference Model for Developing Cloud Applications. *CLOSER*.
- Harrison, R., Blevins, T., & Homan, P. (2007). *Togaf Version 8.1.1 Enterprise Edition*. Van Haren.
- Hassan, Q. F. (2009). Aspects of SOA: An Entry Point for Starters. *Annals. Computer Science Series*, 7(2), 125.
- Heinrich, B., Klier, M., Lewerenz, L., & Zimmermann, S. (2015). Quality-of-Service-Aware Service Selection: A Novel Approach Considering Potential Service Failures and Nondeterministic Service Values. *Service Science*, 7(1), 48–69. <http://doi.org/10.1287/serv.2015.0093>
- Heinrich, B., & Zimmermann, S. (2012). Granularity Metrics for IT Services. In *International Conference on Information Systems, ICIS 2012* (Bd. 4, S. 2820–2838). Orlando.
- Henseler, J., Jonen, A., & Lingnau, V. (2006). *Die Rolle des Controllings bei der Ein- und Weiterführung der Balanced Scorecard - Eine empirische Untersuchung. Beiträge zur Controlling-Forschung*. Kaiserslautern.
- Heschl, J., Gaulke, M., & Bitterli, P. (2005). *CobiT 4.0 Deutsche Version*. IT Governance Institute.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105.
- Hochstein, A. (2006, Januar). *Planerische Prozesse eines industrialisierten Informationsmanagements*. Universität St. Gallen.
- Huergo, R. S., Pires, P. F., Delicato, F. C., Costa, B., Cavalcante, E., & Batista, T. (2014). A systematic survey of service identification methods. *Service Oriented Computing and Applications*, 8(3), 199–219. <http://doi.org/10.1007/s11761-014-0161-y>

- Iacob, M.-E., Jonkers, H., Lankhorst, M., & Proper, E. (2012). *ArchiMate 2.0 Specification*. Van Haren.
- IBM Corporation. (2006a). developerWorks: SOA and Web services. Abgerufen 21. Juni 2006, von <http://www-128.ibm.com/developerworks/webservices>
- IBM Corporation. (2006b). Emerging Technologies Toolkit.
- IFAC/IFIP. (1999). *GERAM: Generalised Enterprise Reference Architecture and Methodology Version 1.6.3*.
- Irani, Z., Love, P. E. D., Elliman, T., Jones, S., & Themistocleous, M. (2005). Evaluating e-government: learning from the experiences of two UK local authorities. *Information Systems Journal*, 15(1), 61–82.
- ISO. (2000). Industrial Automation Systems - Requirements for Enterprise Reference Architectures and Methodologies. In *ISO 15704*.
- ISO. ISO/IEC 20000-1 Information Technology – Service Management Part 1 Specification (2005).
- Jamshidi, P., Khoshnevis, S., Teimourzadegan, A., & Nikraves, F. (2009). Toward automatic transformation of enterprise business model to service model. In *Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems* (S. 70–74).
- Johnson, P., & Lagerstrom, R. (2006). Extended influence diagrams for enterprise architecture analysis. In *10th IEEE International Enterprise Distributed Object Computing Conference*. Hong-Kong.
- Jonkers, H., Lankhorst, M., Van Buuren, R., Hoppenbrouwers, S., & Bonsangue, M. (2004). Concepts for Modelling Enterprise Architectures. *International Journal of Cooperative Information Systems*, 13(03), 257–287.
- Kalle, R., & Lyytinen, J. (2000). Components of Software Development Risk: How to Address Them? A Project Manager Survey. *IEEE Transactions on Software Engineering*, 26(2), 98–112.
- Kang, H., & Bradley, G. (1999). Measuring the Service Performance of Information Technology Departments: An Internal Aervice Management pproach. In *Proceedings of the 9th Workshop on Quality in Services* (S. 462–473).
- Kaplan, R. S., & Norton, D. P. (1992). The Balanced Scorecard - Measures That Drive Performance. *Harvard Business Review*, 70(1), 71–79.
- Khaddaj, S., & Horgan, G. (2004). The Evaluation of Software Quality Factors in Very Large Information Systems. *Electronic Journal of Information Systems Evaluation*, 7(1), 43–48.
- Kim, G., Shin, B., Kim, K., & Lee, H. (2011). IT Capabilities, Process-Oriented Dynamic Capabilities, and Firm Financial Performance. *Journal of the Association for Information Systems*.
- Kim, S., Kim, M., & Park, S. (2008). Service identification using goal and scenario in service oriented architecture. In *15th Asia-Pacific Software Engineering Conference* (S. 419–426). IEEE.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*.
- Klein, M., & König-Ries, B. (2003). A Process and a Tool for Creating Service Descriptions based on DAML-S. *Technologies for E-Services*.
- Klein, M., König-Ries, B., & Müssig, M. (2005). What is Needed for Semantic Service Descriptions? A Proposal for Suitable Language Constructs. *International Journal of Web and Grid Services*.
- Klems, M., Nimis, J., & Tai, S. (2009). Do clouds compute? a framework for estimating the value of cloud computing. *Designing E-Business Systems: Markets, Services, and Networks*, 22, 110–123.
- Klose, K., Knackstedt, R., & Beverungen, D. (2007). Identification of Services-A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning. *ECIS*.
- Klus, M., Fries, B., & Sycara, K. (2006). Automated semantic web service discovery with OWLS-MX. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems* (S. 915–922).

- Kohlborn, T., Korthaus, A., Chan, T., & Rosemann, M. (2009). Identification and analysis of business and software services—a consolidated approach. *IEEE Transactions on Services Computing*, 2(1), 55–64.
- Kohlmann, F., & Alt, R. (2007). Business-Driven Service Modelling-A Methodological Approach from the Finance Industry. *BPSC*.
- Krcmar, H. (1990). Bedeutung und Ziele von Informationssystem-Architekturen. *Wirtschaftsinformatik*, 32(5), 395–402.
- Kritikos, K., & Plexousakis, D. (2009). Requirements for QoS-Based Web Service Description and Discovery. *IEEE Transactions on Services Computing*, 2(4), 320–337.
- Lamanna, D., Skene, J., & Emmerich, W. (2003). SLAng: a language for service level agreements. In *The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems* (S. 100).
- Lankhorst, M. (2013). *Enterprise Architecture at Work*. Berlin/Heidelberg: Springer.
- Latum, F. Van, Solingen, R. van, Oivo, M., Hoisl, B., Rombach, D., & Ruhe, G. (1998). Adopting GQM-based measurement in an industrial environment. *IEEE Software*, 15(1), 78–86.
- Lee, J., Muthig, D., & Naab, M. (2008). An Approach for Developing Service Oriented Product Lines. In *12th International Software Product Line Conference* (S. 275–284). Limerick.
- Lenk, A., Klems, M., Nimis, J., Tai, S., & Sandholm, T. (2009). What's inside the cloud? An architectural map of the cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD 2009* (S. 23–31). <http://doi.org/10.1109/CLOUD.2009.5071529>
- Leymann, F., Roller, D., & Schmidt, M. (2002). Web Services and Business Process Management. *IBM Systems Journal*, 41(2), 198–211.
- Li, S., & Tahvildari, L. (2008). E-BUS: A Toolkit for Extracting Business Services from Java Software Systems. In *Companion of the 13th international conference on Software engineering - ICSE Companion '08* (S. 961–962). New York, New York, USA: ACM Press. <http://doi.org/10.1145/1370175.1370208>
- Li, W., Liu, K., Yang, H., & Yu, C. (2014). Integrated clinical pathway management for medical quality improvement - based on a semiotically inspired systems architecture. *European Journal of Information Systems*, 23(4), 400–417.
- Lima, A. S., de Sousa, J. N., Oliveira, J. A., Sauve, J., & Moura, A. (2010). Towards business-driven continual service improvement. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops* (S. 95–98). IEEE. <http://doi.org/10.1109/NOMSW.2010.5486594>
- Linthicum, D. S. (1999). *Enterprise Application Integration*. Addison-Wesley Professional.
- Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards and. *Nist Special Publication*, 500(292), 35.
- Loske, A., Widjaja, T., & Buxmann, P. (2013). Cloud computing providers' unrealistic optimism regarding it security risks: A threat to users? In *International Conference on Information Systems (ICIS 2013): Reshaping Society Through Information Systems Design* (Bd. 1, S. 773–792). Technische Universität Darmstadt.
- Ludwig, H., Keller, A., Dan, A., Ling, R. P., & Franck, R. (2003). *Web Service Level Agreement (WSLA) Language Specification*. IBM Report.
- Luftman, J. N., & McLean, E. R. (2004). Key Issues for IT Executives. *MIS Quarterly Executive*, 3(2), 89–104.
- Mackenzie, M., Laskey, K., McCabe, F., Brown, P., & Metz, R. (2006). OASIS Reference Model for Service Oriented Architecture 1.0le. *OASIS Standard*, 12.
- March, S. T., & Smith, G. F. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15(4), 251–266.

- Markus, M., & Robey, D. (1988). Information technology and organizational change: causal structure in theory and research. *Management science*.
- Martin, D., Ankolekar, A., Burstein, M., Denker, G., Elenius, D., Hobbs, J., ... Washington, R. (2006). DAML: DAML Services. Abgerufen 8. Mai 2015, von <http://www.daml.org/services/owl-s/1.1/>
- Martin, D., Burstein, M., Hobbs, J., & Lassila, O. (2004). OWL-S: Semantic Markup for Web Services.
- Maximilien, E., & Singh, M. (2004). A framework and ontology for dynamic web services selection. *Internet Computing, IEEE*.
- McCoy, D., & Natis, Y. (2003). Service Oriented Architecture: Mainstream Straight Ahead. *Gartner Research*, (LE-19-7652).
- McDonald, M. P., & Aron, D. (2011). *Reimagining IT: The 2011 CIO Agenda*.
- McGabe, F. G., Estefan, J. A., Laskey, K., & Thornton, D. (2012). Reference Architecture Foundation for Service Oriented Architecture. OASIS Committee Specification.
- Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*. Gaithersburg.
- Menascé, D., Casalicchio, E., & Dubey, V. (2008). A heuristic approach to optimal service selection in service oriented architectures. In *Proceedings of the 7th International Workshop on Software and Performance* (S. 13–24).
- Microsoft Corporation. (2006). .NET Architecture Center: Service Oriented Architecture. Abgerufen 21. Juni 2006, von <http://msdn.microsoft.com/architecture/soa/>
- Microsoft Corporation. (2008). Microsoft Operations Framework MOF 4.0. Abgerufen 23. Februar 2015, von [http://download.microsoft.com/download/6/5/8/658bc1e9-e262-45ca-bb6e-e87c058bbd37/microsoft operations framework 4.zip](http://download.microsoft.com/download/6/5/8/658bc1e9-e262-45ca-bb6e-e87c058bbd37/microsoft%20operations%20framework%204.zip)
- Mooney, J., Gurbaxani, V., & Kraemer, K. (1996). A process oriented framework for assessing the business value of information technology. *ACM SIGMIS Database*.
- Mora, M., & Raisinghani, M. (2014). An extensive review of IT service design in seven international ITSM processes frameworks: Part I. *International Journal of Information Technologies and Systems Approach*, 7(2), 83–106.
- Morfonios, K., Konakas, S., Ioannidis, Y., & Kotsis, N. (2007). ROLAP Implementations of the Data Cube. *ACM Computing Surveys*, 39(4), 1–53.
- Motahari-Nezhad, H. R., Stephenson, B., & Singhal, S. (2009). *Outsourcing Business to Cloud Computing Services: Opportunities and Challenges* (No. 23). *IEEE Internet Computing* (Bd. 10). Palo Alto.
- Namiot, D., & Sneps-Sneppe, M. (2014). On Micro-services Architecture. *International Journal of Open Information Technology*, 2(9), 24–27.
- Napier, N. P., Mathiassen, L., & Johnson, R. D. (2009). Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study. *IEEE Transactions on Software Engineering*, 35(5), 593–606.
- Newcomer, E. (2002). *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison-Wesley Professional.
- Newcomer, E., & Lomow, G. (2004). *Understanding SOA with Web Services*. Addison-Wesley Professional.
- OASIS. (2006). OASIS Committees by Category: SOA.
- Object Management Group. (2006). Business Process Modeling Notation Specification.
- Offermann, P. (2008a). SOAM - A method to design software systems according to service-oriented architecture. *Wirtschaftsinformatik*, 50(6), 461–471. <http://doi.org/10.1365/s11576-008-0094-1>
- Offermann, P. (2008b). SOAM – Eine Methode zur Konzeption betrieblicher Software mit einer

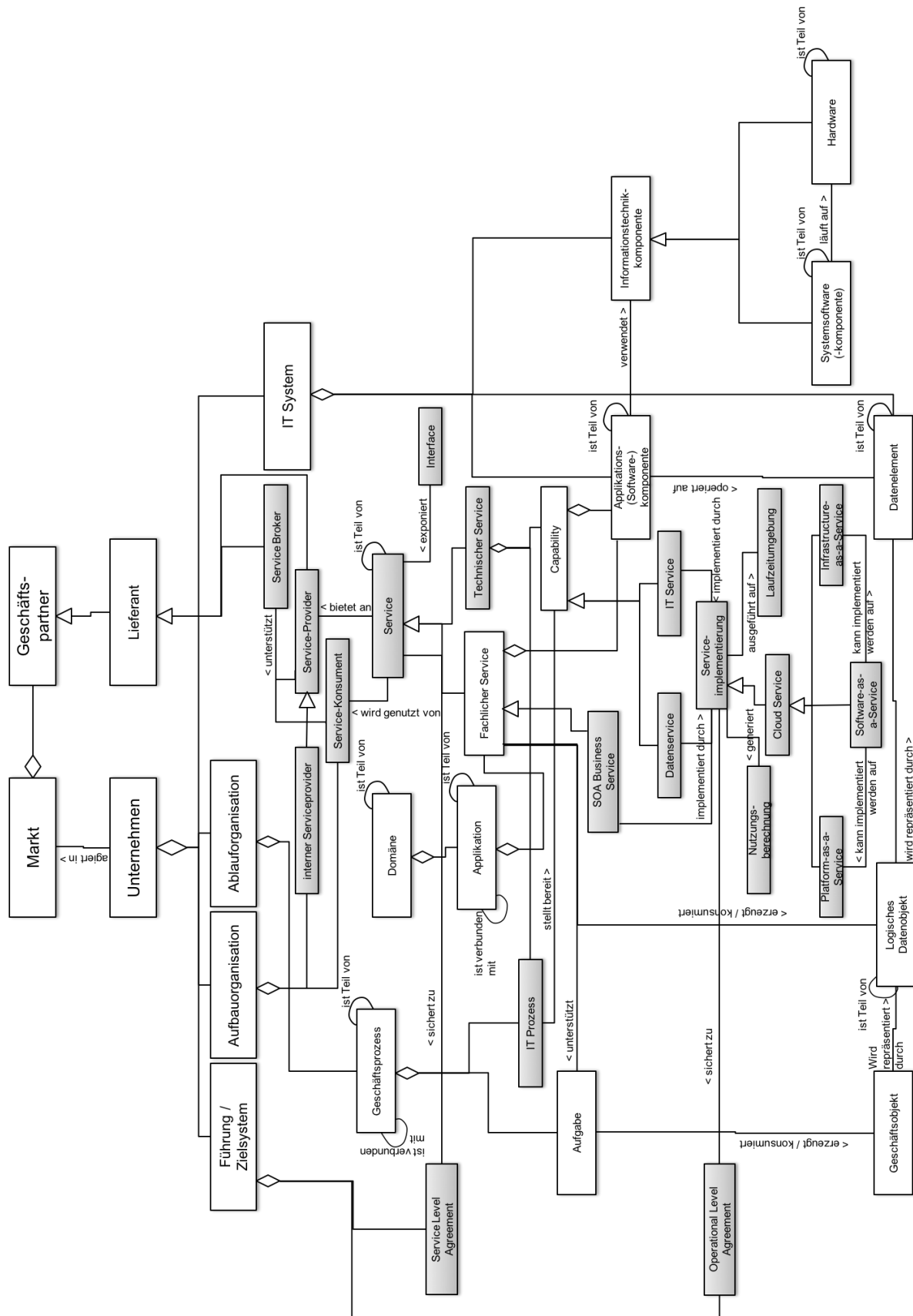
- Serviceorientierten Architektur. *WIRTSCHAFTSINFORMATIK*, 50(6), 461–471. <http://doi.org/10.1365/s11576-008-0094-1>
- Office Of Government Commerce, T. (2005). *Introduction to ITIL*. London: Stationary Office.
- Oldham, N., Verma, K., Sheth, A., & Hakimpur, F. (2006). Semantic WS-Agreement Partner Selection. In *Proceedings of the 15th International Conference on World Wide Web* (S. 697–706).
- OMG, U. (2003). Profile for modeling quality of service and fault tolerance characteristics and mechanisms. *Revised submission, Object Management Group*.
- Orriens, B., & Yang, J. (2005). Bridging the gap between business and IT in service oriented business collaboration. ... *of the IEEE International Conference on ...*
- Oshri, I., Kotlarsky, J., & Willcocks, L. (2007). Managing Dispersed Expertise in IT Offshore Outsourcing: Lessons From TATA Consultancy Services. *MIS Quarterly Executive*, 6(2), 53–65.
- Österle, H. (1995). *Business in the Information Age - Heading for New Processes*. Springer New York.
- Österle, H., & Blessing, D. (2005). Ansätze des Business Engineering. *HMD - Praxis der Wirtschaftsinformatik*, 241(February), 7–17.
- Österle, H., Winter, R., Höning, F., Kurpjuweit, S., & Osl, P. (2007). Business Engineering: Core-Business-Metamodell. *WISU*, (2/07), 191–194.
- Parasuraman, A., Zeithaml, V., & Berry, L. (1985). A conceptual model of service quality and its implications for future research. *The Journal of Marketing*, 49(4), 41–50.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77. <http://doi.org/10.2307/40398896>
- Prakash, S., Deelman, E., & Bagrodia, R. (2000). Asynchronous Parallel Simulation of Parallel Programs. *IEEE Transactions on Software Engineering*, 26(5), 385–400.
- Rabhi, F., Yu, H., Dabous, F., & Wu, S. (2007). A service-oriented architecture for financial business processes. *Information Systems and E-Business Management*, 5(2), 185–200.
- Radhakrishnan, R. (2008). *Enterprise Architecture & IT Service Management - ITSM Frameworks and Processes and their Relationship to EA Frameworks and Processes* (No. W078). San Francisco, USA.
- Ramollari, E., Drandis, D., & Simons, A. J. (2007). A survey of service oriented development methodologies. In *The 2nd European Young Researchers Workshop on Service Oriented Computing* (S. 75–82).
- Repschläger, J., & Zarnekow, R. (2011). *Studie: Cloud Computing in der IKT-Branche*.
- Research Studio Austria. (2006). UMM. Abgerufen 12. Juni 2006, von <http://www.ifs.univie.ac.at/ummaddin/>
- Rimal, B., Jukan, A., Katsaros, D., & Goeleven, Y. (2011). Architectural requirements for cloud computing systems: an enterprise cloud approach. *Journal of Grid Computing*.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorenzen, W. (1991). *Object-Oriented Modeling and Design*. Prentice-Hall International.
- Sallé, M. (2004). IT Service Management and IT Governance: review, comparative analysis and their impact on utility computing. *Hewlett-Packard Company*.
- SAP AG. (2006). SAP - Enterprise Service-Oriented Architecture.
- Scannapieco, M., Virgillito, A., Marchetti, C., Mecella, M., & Baldoni, R. (2004). The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems. *Information Systems*, 29(Data Quality in Cooperative Information Systems), 551–582.
- Schallert, M., & Rosemann, M. (2003). Issues in the design of enterprise architectures. *GI-Arbeitskreis EA Frühjahrskonferenz*.

- Schekckermann, J. (2006). *How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework* (3rd Aufl.). Victoria, British Columbia: Trafford.
- Schelp, J., & Winter, R. (2007). Towards a Methodology for Service Construction. *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, 64–72. <http://doi.org/10.1109/HICSS.2007.565>
- Schelp, J., & Winter, R. (2008). Entwurf von Anwendungssystemen und Entwurf von Enterprise Services. *Wirtschaftsinformatik*, 50(1), 6–15.
- Schmelzer, R. (2005). Solving the Service Granularity Challenge. Abgerufen 12. Juni 2006, von <http://www.zaphthink.com/report.html?id=ZAPFLASH-200639>
- Schönherr, M. (2006). Enterprise Architecture Frameworks. In S. Aier & M. Schönherr (Hrsg.), *Enterprise Application Integration - Serviceorientierung und nachhaltige Architekturen* (S. 342). GITO mbH Verlag.
- Shiang, W., Rau, H., & Lin, Y. (2009). Service identification of a collaborative design workflow in a dynamically changing environment. In *International Conference on Networking, Sensing and Control* (S. 685–690). Okayama: IEEE.
- Shirazi, H., Fareghzadeh, N., & Seyyedi, A. (2009). A combinational approach to service identification in SOA. *Journal of Applied Science Research*, 5(10).
- Simon, B., Goldschmidt, B., & Kondorosi, K. (2013). A Metamodel for the Web Services Standards. *Journal of Grid Computing*, 11(4), 735–752. <http://doi.org/10.1007/s10723-013-9273-4>
- Son, S., Weitzel, T., & Laurent, F. (2005). Designing a process-oriented framework for IT performance management systems. In *Proceedings of the 12th European Conference on IT Evaluation* (S. 433–444).
- Sonnenberg, C., & vom Brocke, J. (2012). Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In K. Peffers, M. A. Rothenberger, & B. Kuechler (Hrsg.), *Design Science in Information Systems. Advances in Theory and Practice* (Bd. 7286, S. 381–397).
- Souza, A., Capelli, C., Santoro, F., Azevedo, L. G., Leite, J. C. S. D. P., & Batista, T. (2011). Service identification in aspect-oriented business process models. In *IEEE 6th International Symposium on Service Oriented System Engineering (SOSE)* (S. 164–174). Irvine, CA: IEEE.
- Sparx Systems Pty Ltd. (2006). UML Tools for Software Development and Modelling - Enterprise Architect Full Lifecycle UML Modelling Tool.
- Stanford Medical Informatics. (2006). Welcome to Protégé. Abgerufen von <http://protege.stanford.edu/overview/protege-owl.html>
- Stephen G., M., & MacDonell, L. (2011). Factors that Affect Software Systems Development Project Outcomes: A Survey of Research. *ACM Computing Surveys*, 43(4), 24.1–24.56.
- Stewart, G., & Chakraborty, A. (2010). Service Identification through Value Chain Analysis and Prioritization. In *16th Americas Conference on Information Systems (AMCIS)* (S. 1–12). Lima.
- Sunita, T., Li, Z., & Rudraraju, J. (2004). Evaluating Web Software Reliability Based on Workload and Failure Data Extracted from Server Logs. *IEEE Transactions on Software Engineering*, 30(11), 754–769.
- Svejvig, P., Storgaard, T., & Møller, C. (2013). Hype or Reality: Will Enterprise Systems as a Service Become an Organizing Vision for Enterprise Cloud Computing in Denmark? In G. Poels (Hrsg.), *Enterprise Information Systems of the Future* (Bd. 139, S. 190–197). Berlin/Heidelberg: Springer. <http://doi.org/10.1007/978-3-642-36611-6>
- Szyperski, C., Bosch, J., & Weck, W. (1999). Component-Oriented Programming. In A. Moreira (Hrsg.), *Object-Oriented Technology ECOOP99 Workshop Reader* (Bd. 1793, S. 184–192). Berlin/Heidelberg: Springer.
- Tan, W.-G., Cater-Steel, A., & Toleman, M. (2009). Implementing it service management: A case

- study focussing on critical success factors. *Journal of Computer Information ...*, 50(2), 1–12.
- Teka, A., Condori-Fernandez, N., & Sapkota, B. (2012). A systematic literature review on service description methods. In B. Regnell & D. Damian (Hrsg.), *Requirements Engineering: Foundation for Software Quality* (Bd. 7195, S. 239–255). Berlin/Heidelberg: Springer.
- The Stationary Office. (2011). An Introductory Overview of ITIL 2011. Abgerufen 30. März 2015, von http://www.itsmf-library.org/Player/eKnowledge/itsmf_an_introductory_overview_of_itil_v3.pdf
- Thones, J. (2015). Microservices. *IEEE Software*, 32(1), 116.
- Tian, M. (2005). *QoS integration in Web services with the WS-QoS framework*.
- Tosic, V., Patel, K., & Pagurek, B. (2002). WSOL — Web Service Offerings Language. In C. Bussler, R. Hull, S. McIlraith, M. Orłowska, B. Pernici, & J. Yang (Hrsg.), *Web Services, E-Business, and the Semantic Web* (Bd. 2512, S. 57–67). Berlin/Heidelberg: Springer.
- Trujillo-Rasua, R., & Domingo-Ferrer, J. (2013). On the privacy offered by (k, δ)-anonymity. *Information Systems*, 38(4), 491–494.
- UN / CEFACT. (2003). UN/CEFACT – ebXML Business Process Specification Schema. Abgerufen 12. Juni 2006, von http://www.untmg.org/index.php?option=com_docman&task=view_category&Itemid=137&subcat=3&catid=63&limitstart=0&limit=5
- UN / CEFACT. (2006a). Core Components Technical Specification Version 2.2. Abgerufen 12. Juni 2006, von http://www.untmg.org/index.php?option=com_docman&task=docclick&Itemid=137&bid=43&limitstart=0&limit=5
- UN / CEFACT. (2006b). UN/CEFACT’s Modeling Methodology (UMM). Abgerufen 12. Juni 2006, von http://www.untmg.org/index.php?option=com_docman&task=view_category&Itemid=137&subcat=1&catid=63&limitstart=0&limit=5
- Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 38(2), 398–424.
- Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A Break in the Clouds: Towards a Cloud Definition. *ACM SIGCOMM Computer Communication Review*, 39(1), 50–55. <http://doi.org/10.1145/1496091.1496100>
- Vasconcelos, A., da Silva, M., Fernandes, A., & Tribolet, J. (2004). An Information System Architectural Framework for Enterprise Application Integration. In *Proceedings of the 37th Annual Hawaii International Conference on Systems Science*.
- Vasconcelos, A., Sousa, P., & Tribolet, J. (2007). Information System Architecture Metrics: an Enterprise Engineering Evaluation Approach. *The Electronic Journal Information Systems Evaluation*, 10(1), 91–122.
- VHB. (2015). VHB Jourqual 3 - Teilranking WI. Abgerufen 15. April 2015, von <http://vhbonline.org/service/jourqual/vhb-jourqual-3/teilrating-wi/>
- Vinoski, S. (1993). Distributed Object Computing with {CORBA}. *C++ Report*, 5, 32–38.
- vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. *ECIS*.
- Wang, L., Tao, J., Kunze, M., Castellanos, A. C., Kramer, D., & Karl, W. (2008). Scientific Cloud Computing: Early Definition and Experience. *2008 10th IEEE International Conference on High Performance Computing and Communications*, 825–830. <http://doi.org/10.1109/HGCC.2008.38>
- Wang, Z., Xu, X., & Zhan, D. (2005). Normal forms and normalized design method for business service. In *IEEE International Conference on e-Business Engineering* (S. 79–86). Beijing: IEEE. <http://doi.org/10.1109/ICEBE.2005.91>
- Weigand, H., & Johannesson, P. (2009). Value-based service modeling and design: Toward a unified view of services. In P. van Eck, J. Gordjin, & R. Wieringa (Hrsg.), *Advanced Information Systems*

- Engineering* (S. 410–424). Berlin/Heidelberg: Springer.
- Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinel, T., Michalk, W., & Stöber, J. (2009). Cloud Computing – A Classification, Business Models, and Research Directions. *Business & Information Systems Engineering*, 1(5), 391.
- Winkler, V., & Buhl, H. (2007). Identifikation und Gestaltung von Services. (German). *Identification and design of services – proceeding and exemplar use in the financial services sector. (English)*, 49(4), 257.
- Winter, R. (2003). Modelle, Techniken und Werkzeuge im Business Engineering. In H. Österle & R. Winter (Hrsg.), *Business Engineering* (S. 87–118). Berlin/Heidelberg: Springer. http://doi.org/http://dx.doi.org/10.1007/978-3-642-19003-2_5
- Winter, R. (2008a). Business Engineering – Betriebswirtschaftliche Konstruktionslehre und ihre Anwendung in der Informationslogistik. In *Business Engineering*.
- Winter, R. (2008b). Design Science Research in Europe. *European Journal of Information Systems*, 17(5), 470–475. <http://doi.org/10.1057/ejis.2008.44>
- Winter, R. (2011). *Business Engineering Navigator: Gestaltung und Analyse von Geschäftslösungen «Business-to-IT»*. (H. Österle, R. Winter, & W. Brenner, Hrsg.). Berlin/Heidelberg: Springer. <http://doi.org/10.1007/978-3-642-15913-8>
- Winter, R., & Fischer, R. (2006). Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In *EDOC Workshop on Trends in Enterprise Architecture (TEAR)*. Los Alamitos: IEEE Computer Society.
- Winter, R., & Fischer, R. (2007). Essential Layers , Artifacts and Dependencies of Enterprise Architecture. *Journal of Enterprise Architecture*, 3(2), 7–18.
- Wohlin, C. (2001). Guest Editor’s Introduction: Seventh International Software Metrics Symposium. *IEEE Transactions on Software Engineering*, 27(11), 961–962.
- Woolf, B. (2006). Introduction to SOA governance. *IBM Developer Works*.
- World Wide Web Consortium. (2004). Web Services Architecture. Abgerufen 12. Juni 2006, von <http://www.w3.org/TR/ws-arch/>
- Yin, R. (2003). *Case Study Research. Design and Methods*. London, New Delhi: Thousand Oaks.
- Yousef, R., Odeh, M., Coward, D., & Sharieh, A. (2009). A Semantically Enriched Framework to Derive Software Service Oriented Models from Business Process Architectures. *International Journal of Information Studies*, 1(4), 231–241.
- Youseff, L., Butrico, M., & Da Silva, D. (2008). Toward a Unified Ontology of Cloud Computing. *2008 Grid Computing Environments Workshop*, 1–10. <http://doi.org/10.1109/GCE.2008.4738443>
- Yusuf, Y., Sarhadi, M., & Gunasekaran, A. (1999). Agile Manufacturing: The Drivers, Concepts and Attributes. *International Journal of Production Economics*, 62(1-2), 33–43.
- Zarnekow, R., Hochstein, A., & Brenner, W. (2005). *Serviceorientiertes IT-Management - ITIL-Best-Practices und -Fallstudien*. Berlin/Heidelberg: Springer.
- Zhang, H. H., Leyuan, S., Meyer, R., Nazareth, D., & D’Souza, W. (2009). Solving Beam-Angle Selection and Dose Optimization Simultaneously via High-Throughput Computing. *INFORMS Journal on Computing*, 21(3), 427–444.
- Zickert Roman, F. (2012). Coping with Existing Systems in Information Systems Development. *IEEE Transactions on Software Engineering*, 38(5), 1027–1039.

Appendix A – Metamodell Spezialisierung



Appendix B – Ergebnisse der Literaturrecherche

Ergebnisse der Literaturrecherche zur Servicegestaltung

1	Abbasi Arab, A. C. (2008). Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. <i>ACM Transactions on Information Systems</i> , 26(3), 1–34.
2	Abdelaziz, E., Mustapha, A., & Mohammed, S. (2011). A service-oriented approach for information systems development. In <i>International Conference on Multimedia and Systems</i> .
3	Abrahamsson KieranXiaofeng Wang, P. (2009, August). «Lots done, more to do»: the current state of agile systems development research. <i>European Journal of Information Systems</i> , S. 281–284.
4	Aier, S. (2006). How Clustering Enterprise Architectures helps to Design Service Oriented Architectures. In <i>2006 IEEE International Conference on Services Computing (SCC'06)</i> (S. 269–272). IEEE. doi:10.1109/SCC.2006.52
5	Aier, S. (2012). Strategies for Establishing Service Oriented Design in Organizations. In F. Joey (Hrsg.), <i>Proceedings of the International Conference on Information Systems ICIS 2012</i> (Bd. 2, S. 1250–1271). Orlando, FL.
6	Alahmari, S. (2010). A model-driven architecture approach to the efficient identification of services on service-oriented enterprise architecture. (EDOCW), 2010 14th IEEE
7	Alizadeh, K., & Mohsenzadeh, M. (2011). Mapping service concept and enterprise ontology in service identification. <i>7th International Conference on Networked Computing (INC)</i> (S. 22–27).
8	Allen, G. N., Burk, D. L., & Davis, G. B. (2006). ACADEMIC DATA COLLECTION IN ELECTRONIC ENVIRONMENTS: DEFINING ACCEPTABLE USE OF INTERNET RESOURCES. <i>MIS Quarterly</i> , 30(3), 599–610.
9	Alonso-Mendo GuyFrias-Martinez, Enrique, F. (2009). Understanding web site redesigns in small- and medium-sized enterprises (SMEs): a U.K.-based study on the applicability of e-commerce Stage Models. <i>European Journal of Information Systems</i> , 18(3), 264–279.

10	Alt, R. (1), & Smits, M. (2). (2010). Design options for service directories in business networks. In 18th European Conference on Information Systems, ECIS 2010 (S. 15p.). (1)University of Leipzig.
11	Amano Akira, K. (2005). A Superpolynomial Lower Bound for a Circuit Computing the Clique Function with at most $(1/6)\log \log n$ Negation Gates. <i>SIAM Journal on Computing</i> , 35(1), 201.
12	Andersson, B., Johannesson, P., & Zdravkovic, J. (2009). Aligning goals and services through goal and business modelling. <i>Information systems and e-Business Management</i> , 7(2), 143–169.
13	Andreas, F., Krivelevich, M., & Goerdts, J. (2005). Recognizing More Unsatisfiable Random k-SAT Instances Efficiently. <i>SIAM Journal on Computing</i> , 35(2), 408.
14	Antunes, P., Herskovic, V., Ochoa, S. F., & Pino, J. A. (2012). Structuring Dimensions for Collaborative Systems Evaluation. <i>ACM Computing Surveys</i> , 44(2), 8–8:28.
15	Arsanjani, A., Ghosh, S., & Allam, A. (2008). SOMA: A method for developing service-oriented solutions. <i>IBM systems Journal</i> , 47(3), 377–396.
16	Asano JiiTokuyama, Takeshi, T. (2007). ZONE DIAGRAMS: EXISTENCE, UNIQUENESS, AND ALGORITHMIC CHALLENGE. <i>SIAM Journal on Computing</i> , 37(4), 1182–1198.
17	ASPNES, J., ATTIYA, H., & CENSOR-HILLEL, K. (2012). Polylogarithmic Concurrent Data Structures from Monotone Circuits. <i>Journal of the ACM</i> , 59(1), 2–2:24.
18	Azevedo, L. G., Santoro, F., & Baião, F. (2009). A method for service identification from business process models in a SOA approach. In T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, & R. Ukor (Hrsg.), <i>Enterprise, Business-Process and Information Systems Modeling</i> (S. 99–112). Berlin/Heidelberg: Springer.
19	Badri, H., Bashiri, M., & Hejazi, T. H. (2013). Integrated strategic and tactical planning in a supply chain network design with a heuristic solution method. <i>Computers and Operations Research</i> , 40, 1143–1154.
20	Baghdadi, Y. (2006). Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing. <i>Information systems</i>

	frontiers, 8(5), 395–410.
21	Bar-Yehuda, R., Bendel, K., Freund, A., & Rawitz, D. (2004). Local Ratio: A Unified Framework for Approximation Algorithms. <i>ACM Computing Surveys</i> , 36(4), 422–463.
22	Baskerville, R., Pries-Heje, J., & Ramesh, B. (2007). The enduring contradictions of new software development approaches: a response to 'Persistent Problems and Practices in ISD'. <i>Information Systems Journal</i> , 17(3), 241–245.
23	Beck, R., Tönker, M., & Dernbecher, S. (2013). Design and evaluation of a grid-in-a-cloud solution. In <i>ECIS 2013 - Proceedings of the 21st European Conference on Information Systems</i> . Goethe University Frankfurt, Department of E-Finance and Services Science: Association for Information Systems.
24	Becker, A., Widjaja, T., & Buxmann, P. (2011). Nutzenpotenziale und Herausforderungen des Einsatzes von Serviceorientierten Architekturen. (German). <i>Wirtschaftsinformatik - Zeitschrift Der Angewandten Informatik</i> , 53(4), 187.
25	Berenbrink, V., Purucker, J., & Bahlinger, T. (2013). Die Bedeutung von Wireframes in der agilen Softwareentwicklung. (German). <i>HMD: Praxis der Wirtschaftsinformatik</i> , 50(2), 27.
26	BERNARDI, S., MERSEGUER, J., & PETRIU, D. C. (2012). Dependability Modeling and Analysis of Software Systems Specified with UML. <i>ACM Computing Surveys</i> , 45(1), 2:1–2:48.
27	Beszédes RudolfGyimóthy, TiborDolenc, AndréKarsisto, Konsta, Á. (2003). Survey of Code-Size Reduction Methods. <i>ACM Computing Surveys</i> , 35(3), 223–267.
28	Beverungen, D., Knackstedt, R., & Müller, O. (2008). Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung – Eine Konzeptionsmethode und ihre Anwendung am Beispiel des Recyclings elektronischer Geräte. (German). <i>Wirtschaftsinformatik</i> , 50(3), 220–234.
29	Bianchini, D., Capiello, C., De Antonellis, V., & Pernici, B. (2014). Service identification in interorganizational process design. <i>IEEE Transactions on Services Computing</i> , 7(2), 265–278.

30	Bianchini, D., & Cappiello, C. (2009). P2S: A methodology to enable inter-organizational process design through web services. In P. van Eck, J. Gordijn, & R. Wieringa (Hrsg.), <i>Advanced Information Systems Engineering</i> (S. 334–348). Berlin/Heidelberg: Springer.
31	Biddle, R., Chiasson, S., & Van Orschot, P. C. (2012). Graphical Passwords: Learning from the First Twelve Years. <i>ACM Computing Surveys</i> , 44(4), 19:1–19:41.
32	Birkmeier, D. Q., Gehlert, A., Overhage, S., & Schlauderer, S. (2013). Alignment of Business and IT Architectures in the German Federal Government: A Systematic Method to Identify Services from Business Processes. In 2013 46th Hawaii International Conference on System Sciences (S. 3848–3857). IEEE. doi:10.1109/HICSS.2013.77
33	Böhm, A., & Kanne, C.-C. (2011). Demaq/Transscale: Automated distribution and scalability for declarative applications. <i>Information Systems</i> , 36(3), 565–578.
34	Böhm, T., Leimeister, J., & Möslin, K. (2014). Service-Systems-Engineering. (German). <i>Service Systems Engineering</i> . (English), 56(2), 83.
35	BOSTOEN SAPEBERBERS, YOLANDE, T. (2013). Power-Reduction Techniques for Data-Center Storage Systems. <i>ACM Computing Surveys</i> , 45(3), 33:1–33:38.
36	Boyland, J. T. (2005). Remote Attribute Grammars. <i>Journal of the ACM</i> , 52(4), 627–687.
37	Braunwarth, K. S., & Friedl, B. (2010). Towards a financially optimal design of it services. In M. Lacity, S. March, & F. Niederman (Hrsg.), <i>ICIS 2010 Proceedings - Thirty First International Conference on Information Systems</i> .
38	Brocke, J., Sonnenberg, C., & Simons, A. (2009). Wertorientierte Gestaltung von Informationssystemen: Konzeption und Anwendung einer Potenzialmodellierung am Beispiel Serviceorientierter Architekturen. (German). <i>Value-oriented Information Systems Design: The Concept of Potentials Modeling and its Application to Service-oriented Architectures</i> . (English), 51(3), 261.
39	Brooks Cindy K.Hardgrave, Bill C.O'Leary-Kelly, Anne M., N. G. R. (2011). IT professional identity: needs, perceptions, and belonging. <i>European Journal of Information Systems</i> , 20(1), 87–102.

40	BROWN, N. (2009). Chemoinformatics-An Introduction for Computer Scientists. <i>ACM Computing Surveys</i> , 41(2), 8:1–8:38.
41	Brüggermann, T., & Breitner, M. (2006). Mobile Preisvergleichsdienste am Scheideweg. (German). <i>Wirtschaftsinformatik -- Zeitschrift Der Angewandten Informatik</i> , 48(6), 430.
42	Caetano, A., Silva, A., & Tribolet, J. (2010). Identification of Services through Functional Decomposition of Business Processes. <i>Business Information Systems</i> , 47, 144–157.
43	Canonaco, P., Legato, P., Mazza, R. M., & Musmanno, R. (2008). A queuing network model for the management of berth crane operations. <i>Computers and Operations Research</i> , 35(Queues in Practice), 2432–2446.
44	Cao KannanPeng XuRamesh, Balasubramaniam, L. (2009). A framework for adapting agile development methodologies. <i>European Journal of Information Systems</i> , 18(4), 332–343.
45	Cardoso, J. M. P., Diniz, P. C., & Weinhardt, M. (2010). Compiling for Reconfigurable Computing: A Survey. <i>ACM Computing Surveys</i> , 42(4), 13:1–13:65.
46	Catalán, J., Epstein, R., Guajardo, M., Yung, D., & Martinez, C. (2009). Solving multiple scenarios in a combinatorial auction. <i>Computers and Operations Research</i> , 36, 2752–2758.
47	Chaari, S., Biennier, F., Favrel, J., & Benamar, C. (2007). Towards a Service-Oriented Enterprise based on Business Components Identification. In R. Goncalves, J. Müller, K. Mertins, & M. Zelm (Hrsg.), <i>Enterprise Interoperability II</i> (S. 495–506). London: Springer.
48	Chang, K.-N., & Kim, D. (2001). Optimal prioritized channel allocation in cellular mobile systems. <i>Computers and Operations Research</i> , 28, 345–356.
49	Cheung, B., Ting, S., Tsang, A., & Lee, W. (2014). A methodological approach to optimizing RFID deployment. <i>Information Systems Frontiers</i> , 16(5), 923.
50	Chin, K.-S., Wang, Y.-M., Ka Kwai Poon, G., & Yang, J.-B. (2009). Failure mode and effects analysis using a group-based evidential reasoning approach. <i>Computers and</i>

	Operations Research, 36, 1768–1779.
51	Choi, J., Nazareth, D. L., & Jain, H. K. (2013). The Impact of SOA Implementation on IT-Business Alignment. <i>ACM Transactions on Management Information Systems (TMIS)</i> , 4(1), 1.
52	Chor, B., & Tuller, T. (2006). Finding a Maximum Likelihood Tree Is Hard. <i>Journal of the ACM</i> , 53(5), 722–744.
53	Chua, H. C.-Y. K.-S. (2007). Soft Pattern Matching Models for Definitional Question Answering. <i>ACM Transactions on Information Systems</i> , 25(2), 1–30.
54	Chua, S. G. W.-H. L.-S. (2006). A Maximal Figure-of-Merit (MFoM)-Learning Approach to Robust Classifier Design for Text Categorization. <i>ACM Transactions on Information Systems</i> , 24(2), 190–218.
55	Cochran, J. K., & Roche, K. T. (2009). A multi-class queuing network analysis methodology for improving hospital emergency department performance. <i>Computers and Operations Research</i> , 36(Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X)), 1497–1512.
56	Cole Ramesh, R. (2003). TREE PATTERN MATCHING TO SUBSET MATCHING IN LINEAR TIME. <i>SIAM Journal on Computing</i> , 32(4), 1056–1066.
57	Córdoba, J.-R. (2009). Critical reflection in planning information systems: a contribution from critical systems thinking. <i>Information Systems Journal</i> , 19(2), 123–147.
58	Cronholm, S. (2009). Using Agile Methods? - Expected Effects. In G. A. Papadopoulos (Hrsg.), (S. 913–922). New York, London, Springer.
59	Crouch, C. J. (2006). Dynamic Element Retrieval in a Structured Environment. <i>ACM Transactions on Information Systems</i> , 24(4), 437–454.
60	Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre Open-Source Software Development: What We Know and What We Do Not Know. <i>ACM Computing Surveys</i> , 44(2), 7–7:35.
61	Cruz-Jesus, F., Oliveira, T., & Bacao, F. (2012). Digital divide across the European

	Union. Information & Management, 49(6), 278–291.
62	Czarnecki, C., Winkelmann, A., & Spiliopoulou, M. (2013). Referenzprozessabläufe für Telekommunikationsunternehmen. (German). Reference Process Flows for Telecommunication Companies. (English), 55(2), 83.
63	D’Amico, S. J., Wang, S.-J., Batta, R., & Rump, C. M. (2002). A simulated annealing approach to police district design. Computers and Operations Research, 29(Location Analysis), 667–684.
64	Dasgupta PetrosHarb, BoulosKumar, RaviMahoney, Michael W., A. (2009). SAMPLING ALGORITHMS AND CORESETS FOR ℓ_p REGRESSION. SIAM Journal on Computing, 38(5), 2060–2078.
65	Davis, C. J., & Hikmet, N. (2008). Training as regulation and development: An exploration of the needs of enterprise systems users. Information & Management, 45(6), 341–348.
66	Detlefs, D., Nelson, G., & Saxe, J. B. (2005). Simplify: A Theorem Prover for Program Checking. Journal of the ACM, 52(3), 365–473.
67	Dinh, T., & Nguyen-Ngoc, A. (2010). A conceptual framework for designing service-oriented inter-organizational information systems. In Symposium on Information and Communication Technology (S. 147–154).
68	Dinitz, Y., Moran, S., & Rajsbaum, S. (2008). Bit Complexity of Breaking and Achieving Symmetry in Chains and Rings. Journal of the ACM, 55(1), 1–28.
69	Dinter, B., & Stroh, F. (2009). Design factors for service-oriented architecture applied to analytical information systems: An explorative analysis. In 17th European Conference on Information Systems, ECIS 2009 (S. 12p.). Institute of Information Management, University of St.Gallen.
70	Du, F., & Evans, G. W. (2008). A bi-objective reverse logistics network analysis for post-sale service. Computers and Operations Research, 35(Queues in Practice), 2617–2634.
71	Du, X., Song, W., & Munro, M. (2009). A Method for Transforming Existing Web Service Descriptions into an Enhanced Semantic Web Service Framework. In G. A. Pa-

	padopoulos (Hrsg.), (S. 217–226). New York, London, Springer.
72	Ducournau, R. (2011). Implementing statically typed object-oriented programming languages. <i>ACM Computing Surveys</i> , 43(3), 1–48.
73	Dwivedi, V., & Kulkarni, N. (2008). A model driven service identification approach for process centric systems. In <i>Congress on Services Part II</i> (S. 65–72). Beijing.
74	Elsner, H., & Krämer, J. (2013). Nutzungsmanagement von Unternehmensportalen mithilfe von Empfehlungssystemen. (German). <i>Managing Corporate Portal Usage with Recommender Systems</i> . (English), 55(4), 205.
75	Erradi, A., Anand, S., & Kulkarni, N. (2006). SOAF: An architectural framework for service definition and realization. In <i>IEEE International Conference on Services Computing</i> (S. 151–158). Chicago, IL: IEEE.
76	Eskandarpour, M., Nikbakhsh, E., & Zegordi, S. H. (2014). Variable neighborhood search for the bi-objective post-sales network design problem: A fitness landscape analysis approach. <i>Computers and Operations Research</i> , 52(Part B), 300–314.
77	ESPARZA, J., KIEFER, S., & LUTTENBERGER, M. (2010). Newtonian Program Analysis. <i>Journal of the ACM</i> , 57(6), 33:1–33:47.
78	Fareghzadeh, N. (2008). Service identification approach to SOA development. <i>Proceedings of World Academy of Science, Engineering and Technology</i> , 45, 258–266.
79	Featherman, M. S., Valacich, J. S., & Wells, J. D. (2006). Is that authentic or artificial? Understanding consumer perceptions of risk in e-service encounters. <i>Information Systems Journal</i> , 16(2), 107–134.
80	Feng ElizabethDillon, Tharam, L. (2002). A Semantic Network-Based Design Methodology for XML Documents. <i>ACM Transactions on Information Systems</i> , 20(4), 390.
81	Fink, A., Kliewer, N., Mattfeld, D., Mönch, L., Rothlauf, F., Schryen, G., ... Voß, S. (2014). Modellbasierte Entscheidungsunterstützung in Produktions- und Dienstleistungsnetzwerken. (German). <i>Model-Based Decision Support in Manufacturing and Service Networks</i> . (English), 56(1), 21.

82	Fitzgerald GerardConboy, Kieran, B. (2006). Customising agile methods to software practices at Intel Shannon. <i>European Journal of Information Systems</i> , 15(2), 200–213.
83	Fleischer Martin, L. (2007). QUICKEST FLOWS OVER TIME. <i>SIAM Journal on Computing</i> , 36(6), 1600–1630.
84	Fleischmann, A., Schmidt, W., Stary, C., & Augl, M. (2013). Agiles Prozessmanagement mittels Subjektorientierung. (German). <i>HMD: Praxis der Wirtschaftsinformatik</i> , 50(2), 64.
85	Fomin, V., Egyedi, T. M. M.. (2007). Multi-method approach to guide design and use of ICT infrastructure services. In <i>Proceedings of the 15th European Conference on Information Systems, ECIS 2007 (S. 1410–1420)</i> . (1)GSCM, Montpellier Business School.
86	Frieze Gregory B., A. (2006). THE PROBABILISTIC RELATIONSHIP BETWEEN THE ASSIGNMENT AND ASYMMETRIC TRAVELING SALESMAN PROBLEMS. <i>SIAM Journal on Computing</i> , 36(5), 1435–1452.
87	Fuller, R. M., Murthy, U., & Schafer, B. A. (2010). The effects of data model representation method on task performance. <i>Information & Management</i> , 47(4), 208–218.
88	Germonprez, M., & Hovorka, D. S. (2013). Member engagement within digitally enabled social network communities: new methodological considerations. <i>Information Systems Journal</i> , 23(6), 525–549.
89	Ghashghai, E., & Rardin, R. L. (2002). Using a hybrid of exact and genetic algorithms to design survivable networks. <i>Computers and Operations Research</i> , 29, 53–66.
90	Ghinita, G., Zhao, K., Papadias, D., & Kalnis, P. (2010). A reciprocal framework for spatial K-anonymity. <i>Information Systems</i> , 35(3), 299–314.
91	Goldberg RussellPaterson, Mike, L. A. (2005). Strong Spatial Mixing with Fewer Colors for Lattice Graphs. <i>SIAM Journal on Computing</i> , 35(2), 486.
92	GOMES PEDRO R. M.PEREIRA, MANUELAFREIRE, MÁRIO M.MONTEIRO, PAULO P., J. V. I. (2013). Detection and Classification of Peer-to-Peer Traffic: A Survey. <i>ACM Computing Surveys</i> , 45(3), 30:1–30:40.

93	Gonnet, P. (2012). A Review of Error Estimation in Adaptive Quadrature. <i>ACM Computing Surveys</i> , 44(4), 22:1–22:36.
94	Gopalakrishnan Nair, T. R., Vaidehi, M., Suma, V., & Rashmi, K. S.. (2012). An enhanced scheduling strategy to accelerate the business performance of the cloud system. <i>Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012, INDIA 2012 (Bd. 132 AISC)</i> .
95	Gouveia, L., Paias, A., & Sharma, D. (2008). Modeling and solving the rooted distance-constrained minimum spanning tree problem. <i>Computers and Operations Research</i> , 35(Part Special Issue: Location Modeling Dedicated to the memory of Charles S. ReVelle), 600–613.
96	Granic, A., Mitrovic, I., & Marangunic, N. (2009). Web Portal Design: Employment of a Range of Assessment Methods. In G. A. Papadopoulos (Hrsg.), (S. 131–140). New York, London, Springer.
97	Guan, T., & Wong, K. F. (2003). Nstar: an interactive tool for local web search. <i>Information & Management</i> , 41(2), 213.
98	Guha Andrew, S. (2009). STREAM ORDER AND ORDER STATISTICS: QUANTILE ESTIMATION IN RANDOM-ORDER STREAMS. <i>SIAM Journal on Computing</i> , 38(5), 2044–2059.
99	Gustas, R., & Gustiene, P. (2009). A New Method for Conceptual Modelling of Information Systems. In G. A. Papadopoulos (Hrsg.), (S. 157–166). New York, London, Springer.
100	Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review. (2006). <i>ACM Computing Surveys</i> , 38(3), 1–24.
101	HARABAGIU FINLEY, S. (2010). Using Topic Themes for Multi-Document Summarization. <i>ACM Transactions on Information Systems</i> , 28(3), 13:1–13:47.
102	Hardin, A. M., Looney, C. A., & Fuller, M. A. (2014). Self-efficacy, learning method appropriation and software skills acquisition in learner-controlled CSSTS environments. <i>Information Systems Journal</i> , 24(1), 3–27.

103	Heinrich, B., & Zimmermann, S. (2012). Granularity metrics for IT services. In International Conference on Information Systems, ICIS 2012 (Bd. 4, S. 2820–2838). University of Regensburg.
104	HIERONS, R. M., KAPOOR, K., BOGDANOV, K., KRAUSE, P., BOWEN, J. P., LÜTTGEN, G., ... ZEDAN, H. (2009). Using Formal Specifications to Support Testing. <i>ACM Computing Surveys</i> , 41(2), 9:1–9:76.
105	Hoest Nir, G. (2006). TOWARD A TOPOLOGICAL CHARACTERIZATION OF ASYNCHRONOUS COMPLEXITY. <i>SIAM Journal on Computing</i> , 36(2), 457–497.
106	Huang, M.-G., Chang, P.-L., & Chou, Y.-C. (2002). Buffer allocation in flow-shop-type production systems with general arrival and service patterns. <i>Computers and Operations Research</i> , 29, 103–121.
107	livari, J., & livari, N. (2011). Varieties of user-centredness: an analysis of four systems development methods. <i>Information Systems Journal</i> , 21(2), 125–153.
108	livari, J., Isomäki, H., & Pekkola, S. (2010). The user the great unknown of systems development: Reasons, forms, challenges, experiences and intellectual contributions of user involvement. <i>Information Systems Journal</i> , 20(2), 109–117. doi:10.1111/j.1365-2575.2009.00336.x
109	Implementing Sorting in Database Systems. (2006). <i>ACM Computing Surveys</i> , 38(3), 1–37.
110	Islam, S. M. S., Bennamoun, M., Owens, R. A., & Davies, R. (2012). A Review of Recent Advances in 3D Ear-and Expression-Invariant Face Biometrics. <i>ACM Computing Surveys</i> , 44(3), 14–14:34.
111	Janson, M., Cecez-Kecmanovic, D., & Zupančič, J. (2007). Prospering in a transition economy through information technology-supported organizational learning. <i>Information Systems Journal</i> , 17(1), 3–36.
112	Jarrah, A. I., & Bard, J. F. (2012). Large-scale pickup and delivery work area design. <i>Computers and Operations Research</i> , 39, 3102–3118.
113	Jiang, J. J., Klein, G., Hwang, H.-G., Huang, J., & Hung, S.-Y. (2004). An exploration of the relationship between software development process maturity and project per-

	formance. <i>Information & Management</i> , 41(3), 279–288.
114	Jiang, J. J., & Muhanna, W. A. (2000). User resistance and strategies for promoting acceptance across system types. <i>Information & Management</i> , 37(1), 25.
115	Jin, Z., & Zhu, H. (2008). A Framework for Agent-based Service-Oriented Modelling. <i>Service-Oriented System Engineering</i> , 2008. SOSE'08.
116	Kamidoi NoriyoshiNagamochi, Hiroshi, Y. (2006). A DETERMINISTIC ALGORITHM FOR FINDING ALL MINIMUM k-WAY CUTS. <i>SIAM Journal on Computing</i> , 36(5), 1329–1341.
117	Kang, D., Song, C., & Baik, D.-K. (2008). A Method of Service Identification for Product Line. In 2008 Third International Conference on Convergence and Hybrid Information Technology (Bd. 2, S. 1040–1045). IEEE. doi:10.1109/ICCIT.2008.113
118	Karlsson Pär, F. (2009). Exploring agile values in method configuration. <i>European Journal of Information Systems</i> , 18(4), 300–316.
119	Karlsson, F., & Hedstrom, K. (2009). Negotiating a Systems Development Method. In G. A. Papadopoulos (Hrsg.), (S. 491–500). New York, London, Springer.
120	Kautz, K., Madsen, S., & Nørbjerg, J. (2007). Continuing the debate: a response to a response – persistent problems and practices in Information Systems Development as enduring contradictions of new software development approaches? <i>Information Systems Journal</i> , 17(3), 247–249.
121	Kazemi, A., & Rostampour, A. (2011). A genetic algorithm based approach to service identification.), 2011 IEEE World
122	Kedlaya Sergey, K. S. Y. (2009). LOCALLY DECODABLE CODES FROM NICE SUBSETS OF FINITE FIELDS AND PRIME FACTORS OF MERSENNE NUMBERS. <i>SIAM Journal on Computing</i> , 38(5), 1952–1969.
123	Kerschbaum, F. (2011). Sicheres und nachhaltiges Benchmarking in der Cloud. (German). <i>Secure and Sustainable Benchmarking in Clouds</i> . (English), 53(3), 129.
124	KHANDEKAR, R., RAO, S., & VAZIRANI, U. (2009). Graph Partitioning using Single

	Commodity Flows. <i>Journal of the ACM</i> , 56(4), 19:1–19:15.
125	Kim, C. S., Klimenok, V., & Taramin, O. (2010). A tandem retrial queueing system with two Markovian flows and reservation of channels. <i>Computers and Operations Research</i> , 37(Algorithmic and Computational Methods in Retrial Queues), 1238–1246.
126	Kim, H. -k., Kim, Y. -j., & Jung, J. -i. (2001). A Mobility Prediction Handover Algorithm for Quality of Service Improvement in Wireless ATM. In M. K. Sein (Hrsg.), (S. 119–128). New York, Kluwer Academic/Plenum, London.
127	Kim, S., Kim, M., & Park, S. (2008). Service identification using goal and scenario in service oriented architecture. In 15th Asia-Pacific Software Engineering Conference (S. 419–426). IEEE.
128	Klein, E. E., & Herskovitz, P. J. (2007). Philosophy of science underpinnings of prototype validation: Popper vs. Quine. <i>Information Systems Journal</i> , 17(1), 111–132.
129	Klendauer, R., Berkovich, M., Gelvin, R., Leimeister, J. M., & Krcmar, H. (2012). Towards a competency model for requirements analysts. <i>Information Systems Journal</i> , 22(6), 475–503.
130	Klose, K., Knackstedt, R., & Beverungen, D. (2007). Identification of Services-A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning. ECIS.
131	Kohlmann, F., & Alt, R. (2007). Business-Driven Service Modelling-A Methodological Approach from the Finance Industry. BPSC.
132	Kreimer, J. (2002). Real-time system with homogeneous servers and nonidentical channels in steady-state. <i>Computers and Operations Research</i> , 29, 1465–1473.
133	Krishen, A. S., Raschke, R. L., & Kachroo, P. (2011). A feedback control approach to maintain consumer information load in online shopping environments. <i>Information & Management</i> , 48(8), 344–352.
134	Kuk, G., & Janssen, M. (2013). Assembling infrastructures and business models for service design and innovation. <i>Information Systems Journal</i> , 23(5), 445–469.
135	Kumar, R. L., Smith, M. A., & Bannerjee, S. (2004). User interface features influencing

	overall ease of use and personalization. <i>Information & Management</i> , 41(3), 289–302.
136	Kytöjoki, J., Nuortio, T., Bräysy, O., & Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. <i>Computers and Operations Research</i> , 34, 2743–2757.
137	Lee, A. S., & Dennis, A. R. (2012). A hermeneutic interpretation of a controlled laboratory experiment: a case study of decision-making with a group support system. <i>Information Systems Journal</i> , 22(1), 3–27.
138	Lee, J., Muthig, D., & Naab, M. (2008). An Approach for Developing Service Oriented Product Lines. In <i>12th International Software Product Line Conference</i> (S. 275–284). Limerick.
139	Lee, S., & Ahn, H. Assessment of process improvement from organizational change, <i>45 Information & Management</i> 270–280 (Juli 2008).
140	Lee, Z., Wagner, C., & Shin, H. K. (2008). The effect of decision support system expertise on system use behavior and performance. <i>Information & Management</i> , 45(6), 349–358.
141	Legner, C., & Heutschi, R. (2007). SOA adoption in practice-Findings from early SOA implementations. In <i>Proceedings of the 15th European Conference on Information Systems, ECIS 2007</i> (S. 1643–1654). University of St. Gallen.
142	Lehtonen, M. (2006). Preparing Heterogeneous XML for Full-Text Search. <i>ACM Transactions on Information Systems</i> , 24(4), 455–474.
143	Leontiev M. TamerSzafron, Duane, Y. (2002). On Type Systems for Object-Oriented Database Programming Languages. <i>ACM Computing Surveys</i> , 34(4), 409–449.
144	Li, S., & Tahvildari, L. (2008). E-BUS: A Toolkit for Extracting Business Services from Java Software Systems. In <i>Companion of the 13th international conference on Software engineering - ICSE Companion '08</i> (S. 961–962). New York, New York, USA: ACM Press. doi:10.1145/1370175.1370208
145	Li, W., Liu, K., Yang, H., & Yu, C. (2014). Integrated clinical pathway management for medical quality improvement - based on a semiotically inspired systems architecture. <i>European Journal of Information Systems</i> , 23(4), 400–417.

146	Lin, J. (2007). An Exploration of the Principles Underlying Redundancy-Based Factoid Question Answering. <i>ACM Transactions on Information Systems</i> , 25(2), 1–55.
147	Liu, K. C. (2006). iVIBRATE: Interactive Visualization-Based Framework for Clustering Large Datasets. <i>ACM Transactions on Information Systems</i> , 24(2), 245–294.
148	Lo, S.-C., & Hall, R. W. (2008). The design of real-time logistics information system for trucking industry. <i>Computers and Operations Research</i> , 35(Part Special Issue: Topics in Real-time Supply Chain Management), 3439–3451.
149	Loebbecke, C. (2007). Use of innovative content integration information technology at the point of sale. <i>European Journal of Information Systems</i> , 16(3), 228–236.
150	Loebbecke, C., Thomas, B., & Ullrich, T. (2012). ASSESSING CLOUD READINESS AT CONTINENTAL AG. <i>MIS Quarterly Executive</i> , 11(1), 11–23.
151	Loske, A., Widjaja, T., & Buxmann, P. (2013). Cloud computing providers' unrealistic optimism regarding it security risks: A threat to users? In <i>International Conference on Information Systems (ICIS 2013): Reshaping Society Through Information Systems Design</i> (Bd. 1, S. 773–792). Technische Universität Darmstadt.
152	Luo SiCallan, J. (2003). A Semisupervised Learning Method to Merge Search Engine Results. <i>ACM Transactions on Information Systems</i> , 21(4), 457.
153	Madsen KarlheinzVidgen, Richard, S. (2006). A framework for understanding how a unique and local IS development method emerges in practice. <i>European Journal of Information Systems</i> , 15(2), 225–238.
154	Mangalaraj RadhaKantaNerur, Sridhar, G. (2009). Acceptance of software process innovations -- the case of extreme programming. <i>European Journal of Information Systems</i> , 18(4), 344–354.
155	Mantzana MarinosIrani, ZahirMorabito, Vincenzo, V. (2007). Identifying healthcare actors involved in the adoption of information systems. <i>European Journal of Information Systems</i> , 16(1), 91–102.
156	Martens, B., Teuteberg, F., & Gräuler, M. (2011). Design and implementation of a community platform for the evaluation and selection of Cloud Computing Services: A market analysis. In <i>19th European Conference on Information Systems, ECIS 2011</i>

	(S. 7p.). University of Osnabrueck.
157	Maruping Viswanath, L. M. X. Z. (2009). Role of collective ownership and coding standards in coordinating expertise in software project teams. <i>European Journal of Information Systems</i> , 18(4), 355–371.
158	Mathiassen, L., & Puroo, S. (2002). Educating reflective systems developers. <i>Information Systems Journal</i> , 12(2), 81–102.
159	McAvoy Tom, J. (2009). The role of project management in ineffective decision making within Agile software development projects. <i>European Journal of Information Systems</i> , 18(4), 372–383.
160	McAvoy, J., Nagle, T., & Sammon, D. (2013). Using mindfulness to examine ISD agility. <i>Information Systems Journal</i> , 23(2), 155–172.
161	McCann Nicholas, M. (2005). SRT Division Algorithms as Dynamical Systems. <i>SIAM Journal on Computing</i> , 34(6), 1279.
162	McKenna, B., Tuunanen, T., & Gardner, L. (2013). Consumers' adoption of information services. <i>Information & Management</i> , 50(5), 248–257. doi:10.1016/j.im.2013.04.004
163	Menascé, D., Casalicchio, E., & Dubey, V. (2008). A heuristic approach to optimal service selection in service oriented architectures. In <i>Proceedings of the 7th International Workshop on Software and Performance</i> (S. 13–24).
164	Mester, D., & Bräysy, O. (2005). Active guided evolution strategies for large-scale vehicle routing problems with time windows. <i>Computers and Operations Research</i> , 32, 1593–1614.
165	Mojsilović, A., Ray, B., Lawrence, R., & Takriti, S. (2007). A logistic regression framework for information technology outsourcing lifecycle management. <i>Computers and Operations Research</i> , 34(Operations Research and Outsourcing), 3609–3627.
166	Morfonios, K., Konakas, S., Ioannidis, Y., & Kotsis, N. (2007). ROLAP Implementations of the Data Cube. <i>ACM Computing Surveys</i> , 39(4), 1–53.
167	Mortensen, C. W. (2006). FULLY DYNAMIC ORTHOGONAL RANGE REPORTING

	ON RAM. <i>SIAM Journal on Computing</i> , 35(6), 1494–1525.
168	Nakamura, E. F., Loureiro, A. A. F., & Frery, A. C. (2007). Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications. <i>ACM Computing Surveys</i> , 39(3), 1–55.
169	Nakatsu, R. T., & Iacovou, C. L. (2009). A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. <i>Information & Management</i> , 46(1), 57–68.
170	Nie, C., & Leung, H. (2011). A Survey of Combinatorial Testing. <i>ACM Computing Surveys</i> , 43(2), 11:1–11:29.
171	O’Flaherty, B., Pope, A., Thornton, C., & Woodworth, S. (2013). Capturing multi-stakeholder needs in customer-centric cloud service design. In <i>International Conference on Information Systems (ICIS 2013): Reshaping Society Through Information Systems Design</i> (Bd. 1, S. 688–702). Business Information Systems, University College Cork.
172	Offermann, P. (2008). SOAM - A method to design software systems according to service-oriented architecture. <i>Wirtschaftsinformatik</i> , 50(6), 461–471. doi:10.1365/s11576-008-0094-1
173	Offermann, P., & Bub, U. (2009). Empirical comparison of methods for information systems development according to SOA. In <i>17th European Conference on Information Systems, ECIS 2009</i> (S. 13p.). Deutsche Telekom Laboratories.
174	ORGERIE MARCOS DIASLEFEVRE, LAURENT, A.-C. A. (2014). A Survey on Techniques for Improving the Energy Efficiency of Large-Scale Distributed Systems. <i>ACM Computing Surveys</i> , 46(4), 47:1–47:31.
175	Oshri, I., Kotlarsky, J., & Willcocks, L. (2007). Managing Dispersed Expertise in IT Offshore Outsourcing: Lessons From TATA Consultancy Services. <i>MIS Quarterly Executive</i> , 6(2), 53–65.
176	Oshri, I., van Fenema, P., & Kotlarsky, J. (2008). Knowledge transfer in globally distributed teams: the role of transactive memory. <i>Information Systems Journal</i> , 18(6), 593–616.

177	Pal, A., Harper, F. M., & Konstan, J. A. (2012). Exploring Question Selection Bias to Identify Experts and Potential Experts in Community Question Answering. <i>ACM Transactions on Information Systems</i> , 30(2), 10.1–10.28.
178	Peng, T., Leckie, C., & Ramamohanarao, K. (2007). Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems. <i>ACM Computing Surveys</i> , 39(1), 1–42.
179	Peters, C., & Leimeister, J. M. (2013). TM3-A modularization method for telemedical services: Design and evaluation. In <i>ECIS 2013 - Proceedings of the 21st European Conference on Information Systems</i> . Kassel University: Association for Information Systems.
180	Petter, S., & McLean, E. R. (2009). A meta-analytic assessment of the DeLone and McLean IS success model: An examination of IS success at the individual level. <i>Information & Management</i> , 46(3), 159–166.
181	Pitts, A. M. (2006). Alpha-Structural Recursion and Induction. <i>Journal of the ACM</i> , 53(3), 459–506.
182	Port Tung, D. (2009). Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications. <i>European Journal of Information Systems</i> , 18(4), 317–331.
183	Purao, S., & Vaishnavi, V. (2003). Product Metrics for Object-Oriented Systems. <i>ACM Computing Surveys</i> , 35(2), 191–221.
184	PÜSCHEL, M., MILDNER, P. A., & HOE, J. C. (2009). Permuting Streaming Data Using RAMs. <i>Journal of the ACM</i> , 56(2), 10–10:34.
185	QUAN WANGJUN XUHANG LICRASWELL, N. (2013). Regularized Latent Semantic Indexing: A New Approach to Large-Scale Topic Modeling. <i>ACM Transactions on Information Systems</i> , 31(1), 1–44.
186	Rajavel, R., & Mala, T. (2012). Achieving Service Level Agreement in Cloud Environment Using Job Prioritization in Hierarchical Scheduling. <i>Proceedings of the International Conference on Information Systems Design & Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012</i> , 547.

187	Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. <i>Information Systems Journal</i> , 20(5), 449–480.
188	Ramsin, R., & Paige, R. F. (2008). Process-Centered Review of Object Oriented Software Development Methodologies. <i>ACM Computing Surveys</i> , 40(1), 1–89.
189	Rebuge, Á., & Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. <i>Information Systems</i> , 37(2), 99–116.
190	Remus, U., & Wiener, M. (2010). A multi-method, holistic strategy for researching critical success factors in IT projects. <i>Information Systems Journal</i> , 20(1), 25–52.
191	Ricca, F., & Marchetto, A. (2009). A quick and dirty meet-in-the-middle approach for migrating to SOA. <i>Proceedings of the joint international and annual</i>
192	Richardson, H., & Robinson, B. (2007). The mysterious case of the missing paradigm: a review of critical information systems research 1991–2001. <i>Information Systems Journal</i> , 17(3), 251–270.
193	Rittgen, P. (2006). A language-mapping approach to action-oriented development of information systems. <i>European Journal of Information Systems</i> , 15(1), 70–81.
194	Rothenberger, M. A., Kao, Y.-C., & Van Wassenhove, L. N. (2010). Total quality in software development: An empirical study of quality drivers and benefits in Indian software projects. <i>Information & Management</i> , 47(7/8), 372–379.
195	Rudin III R., J. F. C. (2003). IMPROVED BOUNDS FOR THE ONLINE SCHEDULING PROBLEM. <i>SIAM Journal on Computing</i> , 32(3), 717.
196	Ruiz-Hernández, D., Delgado-Gómez, D., & López-Pascual, J. (2014). Restructuring bank networks after mergers and acquisitions: A capacitated delocation model for closing and resizing branches. <i>Computers and Operations Research</i> .
197	Saebo, O., Paivarinta, T., Austbo, J. H., & Scheie, S. S. (2009). Elaborating the WARE Method for eParticipation Requirements. In G. A. Papadopoulos (Hrsg.), (S. 785–792). New York, London, Springer.

198	SALFNER, F., LENK, M., & MALEK, M. (2010). A Survey of Online Failure Prediction Methods. <i>ACM Computing Surveys</i> , 42(3), 10:1–10:42.
199	Sari, A. K., Rahayu, W., & Bhatt, M. (2013). An approach for sub-ontology evolution in a distributed health care enterprise. <i>Information Systems</i> , 38(5), 727–744.
200	Sarkar, P. K., & Cybulski, J. (2004). PROCESS OF REQUIREMENTS EVOLUTION IN WEB-ENABLED EMPLOYEE SERVICE SYSTEMS. In H. Linger (Hrsg.), (S. 331–340). New York, London, Kluwer Academic/Plenum Publishers.
201	Sarker Charles L.Sarker, SuprateekChakraborty, Suranjan, S. (2009). Assessing the relative contribution of the facets of agility to distributed systems development success: an Analytic Hierarchy Process approach. <i>European Journal of Information Systems</i> , 18(4), 285–299.
202	Schmidt, R. (2011). A service-system based identification of meta-services for service-oriented enterprise architecture. In <i>IEEE International Enterprise Distributed Computing Workshop, EDOC</i> (S. 293–300).
203	Schulman Erik, R. (2009). PROGRAMMABLE CONTROL OF NUCLEATION FOR ALGORITHMIC SELF-ASSEMBLY. <i>SIAM Journal on Computing</i> , 39(4), 1581–1616.
204	Schümmer, J., & Reichenberger, K. (2013). Semantische Netze in agilen Projekten. (German). <i>HMD: Praxis der Wirtschaftsinformatik</i> , 50(2), 35.
205	Scott, J. E., & Walczak, S. (2009). Cognitive engagement with a multimedia ERP training tool: Assessing computer self-efficacy and technology acceptance. <i>Information & Management</i> , 46(4), 221–232.
206	Seidel Micha, R. (2005). Top-Down Analysis of Path Compression. <i>SIAM Journal on Computing</i> , 34(3), 515.
207	Shackleton, P., Fisher, J., & Dawson, L. (2004). E-GOVERNMENT SERVICES: ONE LOCAL GOVERNMENT'S APPROACH. In H. Linger (Hrsg.), (S. 581–592). New York, London, Kluwer Academic/Plenum Publishers.
208	Shaft, T. M., Albert, L. J., & Jaspersen, J. (2008). Managing change in an information systems development organization: understanding developer transitions from a structured to an object-oriented development environment. <i>Information Systems Journal</i> ,

	18(3), 275–297.
209	Shirazi, H., Fareghzadeh, N., & Seyyedi, A. (2009). A combinational approach to service identification in SOA. <i>Journal of Applied Science Research</i> , 5(10).
210	Shoval, P., & Kabeli-Shani, J. (2009). Designing Class Methods from Dataflow Diagrams. In G. A. Papadopoulos (Hrsg.), (S. 1–10). New York, London, Springer.
211	Siau, K., Tan, X., & Sheng, H. (2010). Important characteristics of software development team members: an empirical investigation using Repertory Grid. <i>Information Systems Journal</i> , 20(6), 563–580.
212	Smolander MattiPurao, Sandeep, K. (2008). Software architectures: Blueprint, Literature, Language or Decision? <i>European Journal of Information Systems</i> , 17(6), 575–588.
213	Song, J., & Kim, S. (2009). A study on applying context-aware technology on hypothetical shopping advertisement. <i>Information Systems Frontiers</i> , 11(5), 561.
214	Souza, A., Capelli, C., Santoro, F. ., Azevedo, L. G., Leite, J. C. S. D. P., & Batista, T. (2011). Service identification in aspect-oriented business process models. In <i>IEEE 6th International Symposium on Service Oriented System Engineering (SOSE)</i> (S. 164–174). Irvine, CA: IEEE.
215	Spakowski Rahul, H. (2009). HIERARCHICAL UNAMBIGUITY. <i>SIAM Journal on Computing</i> , 38(5), 2079–2112.
216	Stamelos, I., Angelis, L., Morisio, M., Sakellaris, E., & Bleris, G. L. (2003). Estimating the development cost of custom software. <i>Information & Management</i> , 40(8), 729.
217	Stephen G., M., & MacDonell, L. (2011). Factors that Affect Software Systems Development Project Outcomes: A Survey of Research. <i>ACM Computing Surveys</i> , 43(4), 24.1–24.56.
218	Stewart, G., & Chakraborty, A. (2010). Service Identification through Value Chain Analysis and Prioritization. In <i>16th Americas Conference on Information Systems (AMCIS)</i> (S. 1–12). Lima.

219	Sunyaev, A., Hansen, M., & Krcmar, H. (2009). Method Engineering: A Formal Description. In G. A. Papadopoulos (Hrsg.), (S. 645–654). New York, London, Springer.
220	Symmetry in Temporal Logic Model Checking. (2006). ACM Computing Surveys, 38(3), 1–36.
221	Thomas, O., Walter, P., & Loos, P. (2008). Product-Service Systems: Konstruktion und Anwendung einer Entwicklungsmethodik. (German). Product-Service Systems: Design and Usage of an Engineering Methodology. (English), 50(3), 208.
222	Tobias, K., & Wolfgang, M. (2013). Towards empirically validated ubiquitous information systems: Results from a pretest and three empirical studies. In ECIS 2013 - Proceedings of the 21st European Conference on Information Systems. (1)Institute of Technology Management (ITEM-HSG), University of St.Gallen: Association for Information Systems.
223	Touloum, K. (1), Idoughi, D. (1), Seffah, A. (2), & Sabroux, C.-R. (2). (2013). Scenarios + touchpoints = A method for analyzing crisis situations and designing management and rescue services. In ISCRAM 2013 Conference Proceedings - 10th International Conference on Information Systems for Crisis Response and Management (S. 497–501). (1)Laboratory of Applied Mathematics, University A. Mira: Karlsruher Institut für Technologie (KIT).
224	Trujillo-Rasua, R., & Domingo-Ferrer, J. (2013). On the privacy offered by (k, δ) -anonymity. Information Systems, 38(4), 491–494.
225	Turetken, O., & Demirors, O. (2011). Plural: A decentralized business process modeling method. Information & Management, 48, 235–247.
226	Uludag, S., Lui, K.-S., Nahrstedt, K., & Brewster, G. (2007). Analysis of Topology Aggregation Techniques for QoS Routing. ACM Computing Surveys, 39(3), 1–31.
227	Valtonen, K., & Leppanen, M. (2009). Business Architecture Development at Public Administration - Insights from Government EA Method Engineering Project in Finland. In G. A. Papadopoulos (Hrsg.), (S. 765–774). New York, London, Springer.

228	Vernekar, S. S., & Game, P. (2012). Component Based Resource Allocation in Cloud Computing. Proceedings of the International Conference on Information Systems Design & Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012, 907.
229	Wang, E. T. G., Ju, P.-H., Jiang, J. J., & Klein, G. (2008). The effects of change control and management review on software flexibility and project performance. <i>Information & Management</i> , 45(7), 438–443.
230	Wang, M., Ni, B., Hua, X., & Chua, T. (2012). Assistive Tagging: A Survey of Multimedia Tagging with Human-Computer Joint Exploration. <i>ACM Computing Surveys</i> , 44(4), 25:1–25:24.
231	Wang, Z., Xu, X., & Zhan, D. (2005). Normal forms and normalized design method for business service. In <i>IEEE International Conference on e-Business Engineering</i> (S. 79–86). Beijing: IEEE. doi:10.1109/ICEBE.2005.91
232	Weigand, H., & Johannesson, P. (2009). Value-based service modeling and design: Toward a unified view of services. In P. van Eck, J. Gordjin, & R. Wieringa (Hrsg.), <i>Advanced Information Systems Engineering</i> (S. 410–424). Berlin/Heidelberg: Springer.
233	Weisbecker, A., Falkner, J., & Höß, O. (2014). Integrationsszenarios und -plattformen für die Migration von Anwendungssystemen in die Cloud. (German). <i>Integration Scenarios and Platforms for the Migration From Application Systems Into the Cloud</i> . (English), 51(2), 119.
234	White Baker, E. (2011). Why situational method engineering is useful to information systems development. <i>Information Systems Journal</i> , 21(2), 155–174.
235	Winkler, V., & Buhl, H. (2007). Identifikation und Gestaltung von Services. (German). <i>Identification and design of services — proceeding and exemplar use in the financial services sector</i> . (English), 49(4), 257.
236	Winter, R. (2008). Design Science Research in Europe. <i>European Journal of Information Systems</i> , 17(5), 470–475. doi:10.1057/ejis.2008.44
237	Wiredu, G. O. (2011). Understanding the functions of teleconferences for coordinating

	global software development projects. <i>Information Systems Journal</i> , 21(2), 175–194.
238	Wistrand, K., Karlsson, F., & Agerfalk, P. J. (2009). Exploring the Role of Method Rationale in the Context of Teaching Information Systems Development Methods. In G. A. Papadopoulos (Hrsg.), (S. 801–810). New York, London, Springer.
239	WOODCOCK, J. I. M., LARSEN, P. G., BICARREGUI, J., & FITZGERALD, J. (2009). Formal Methods: Practice and Experience. <i>ACM Computing Surveys</i> , 41(4), 19:1–19:36.
240	Wu, J.-H. (2009). A design methodology for form-based knowledge reuse and representation. <i>Information & Management</i> , 46(7), 365–375.
241	XIAO FANGLIU SHENG MICHAEL, O. R. C. (2007). ServiceFinder: A Method Towards Enhancing Service Portals. <i>ACM Transactions on Information Systems</i> , 25(4), 17–17.28.
242	XIAO, X. W. (2010). Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction. <i>ACM Transactions on Information Systems</i> , 28(2), 8:1–8:34.
243	Yang Hye-RyunMason, Robert M., H.-D. (2008). An exploratory study on meta skills in software development teams: antecedent cooperation skills and personality for shared mental models. <i>European Journal of Information Systems</i> , 17(1), 47–61.
244	Ye Du SanjuktaRamesh, R., A. (2013). Efficient Risk Hedging by Dynamic Forward Pricing: A Study in Cloud Computing. <i>INFORMS Journal on Computing</i> , 25(4), 625–642.
245	YE, J. Z. X. H. (2014). Modeling Term Associations for Probabilistic Information Retrieval. <i>ACM Transactions on Information Systems</i> , 32(2), 1–47.
246	Yetim, F. (2006). Acting with genres: discursive-ethical concepts for reflecting on and legitimating genres. <i>European Journal of Information Systems</i> , 15(1), 54–69.
247	Yousef, R., Odeh, M., Coward, D., & Sharieh, A. (2009). A Semantically Enriched Framework to Derive Software Service Oriented Models from Business Process Architectures. <i>International Journal of Information Studies</i> , 1(4), 231–241.

248	Yu, Z., & Nakamura, Y. (2010). Smart Meeting Systems: A Survey of State-of-the-Art and Open Issues. <i>ACM Computing Surveys</i> , 42(2), 8–8.20.
249	Zachariadis, E. E., & Kiranoudis, C. T. (2010). An open vehicle routing problem metaheuristic for examining wide solution neighborhoods. <i>Computers and Operations Research</i> , 37, 712–723.
250	Zang, C., Fan, Y., & Liu, R. (2008). Architecture, implementation and application of complex event processing in enterprise information systems based on RFID. <i>Information Systems Frontiers</i> , 10(5), 543.
251	Zhang, Z., Zhou, D., Yang, H., & Zhong, S. (2010). A service composition approach based on sequence mining for migrating e-learning legacy system to SOA. <i>International Journal of ...</i>
252	Zhao, J. (1), Sun, S. X. (1), & Sarkar, S. (2). (2013). Coordinating one-to-many concurrent negotiation for service provision. In <i>International Conference on Information Systems (ICIS 2013): Reshaping Society Through Information Systems Design (Bd. 1, S. 910–928)</i> . (1)Department of Information Systems, City University of Hong Kong.
253	Zhao, Y., Si, H., Ni, Y., & Qi, H. (2009). A Service-Oriented Analysis and Design Approach Based On Data Flow Diagram. In <i>2009 International Conference on Computational Intelligence and Software Engineering (S. 1–5)</i> . IEEE. doi:10.1109/CISE.2009.5365568
254	Ziekow, H., & Günther, O. (2010). Sharing RFID and complex event data among organizations. <i>Information Systems Frontiers</i> , 12(5), 541.
255	Zimmer, J. C., Arsal, R. E., Al-Marzouq, M., & Grover, V. (2010). Investigating online information disclosure: Effects of information relevance, trust and risk. <i>Information & Management</i> , 47, 115–123.

Ergebnisse der Literaturrecherche zur Serviceverbesserung

1	Ahuja Krishna C.Orlin, James B.Sharma, Dushyant, R. K. J. (2007). Very Large-Scale Neighborhood Search for the Quadratic Assignment Problem. <i>INFORMS Journal on</i>
---	--

	<i>Computing</i> , 19(4), 646–657.
2	Akinde, M., Böhlen, M. H., Chatziantoniou, D., & Gamper, J. (2011). multi-dimensional aggregation. <i>Information Systems</i> , 36(2), 341–358.
3	Alter, S. (2013). Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future. <i>Journal of the Association for Information Systems</i> , 14(2), 72–121.
4	Antoniol GerardoCasazza, Gerardode Lucia, AndreaMerlo, Ettore, G. (2002). Recovering Traceability Links between Code and Documentation. <i>IEEE Transactions on Software Engineering</i> , 28(10), 970–983.
5	Arisholm Lionel C.Hove, Siw ElisabethLabiche, Yvan, E. (2006). The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation. <i>IEEE Transactions on Software Engineering</i> , 32(6), 365–381.
6	Ashish, J., & Saxena, P. (2002). Optimum Control Limits for Employing Statistical Process Control in Software Process. <i>IEEE Transactions on Software Engineering</i> , 28(12), 1126–1134.
7	Baker Ibrahim, R. (2013). An Empirical Evaluation of Mutation Testing for Improving the Test Quality of Safety-Critical Software. <i>IEEE Transactions on Software Engineering</i> , 39(6), 787–805.
8	Barbara, P., Philippsen, M., Tichy, W. F., & Unger-Lamprecht, L. (2002). Two Controlled Experiments Assessing the Usefulness of Design Pattern Documentation in Program Maintenance. <i>IEEE Transactions on Software Engineering</i> , 28(6), 595–606.
9	Bardhan, I. R., Demirkan, H., Kannan, P. K., Kauffman, R. J., & Sougstad, R. (2014, Dezember 8). An Interdisciplinary Perspective on IT Services Management and Service Science. <i>Journal of Management Information Systems</i> . Routledge.
10	Becker, A., Widjaja, T., & Buxmann, P. (2011). Nutzenpotenziale und Herausforderungen des Einsatzes von Serviceorientierten Architekturen. (German). <i>Wirtschaftsinformatik - Zeitschrift Der Angewandten Informatik</i> , 53(4), 187.

11	Behrend, A., Schüller, G., & Wieneke, M. (2013). Efficient tracking of moving objects using a relational database. <i>Information Systems</i> , 38(8), 1269–1284.
12	Böhmman, T., & Krcmar, H. (2004). Grundlagen und Entwicklungstrends im IT-Servicemanagement. <i>HMD - Praxis der Wirtschaftsinformatik</i> , 41(237), 1–18.
13	Bowman, M., Briand, L. C., & Labiche, Y. (2010). Solving the Class Responsibility Assignment Problem in Object-Oriented Analysis with Multi-Objective Genetic Algorithms. <i>IEEE Transactions on Software Engineering</i> , 36(6), 817.
14	Braunwarth, K. S., & Friedl, B. (2010). Towards a financially optimal design of it services. In M. Lacity, S. March, & F. Niederman (Hrsg.), <i>ICIS 2010 Proceedings - Thirty First International Conference on Information Systems</i> .
15	Cangussu Raymond A.Mathur, Aditya P., J. W. D. (2003). Using Sensitivity Analysis to Validate a State Variable Model of the Software Test Process. <i>IEEE Transactions on Software Engineering</i> , 29(5), 430–443.
16	Chen, C.-C., Wu, C.-S., & Wu, R. C.-F. (2006). e-Service enhancement priority matrix: The case of an IC foundry company. <i>Information & Management</i> , 43, 572–586.
17	Comuzzi, M., Jacobs, G., & Grefen, P. (2013). Understanding SLA Elements in Cloud Computing. <i>Collaborative Systems</i>
18	Córdoba, J.-R. (2009). Critical reflection in planning information systems: a contribution from critical systems thinking. <i>Information Systems Journal</i> , 19(2), 123–147.
19	Elliott, B., Cheng, E., Mayes, S., & Ozsoyoglu, Z. M. (2009). Efficiently calculating inbreeding on large pedigrees databases. <i>Information Systems</i> , 34(6), 469–492.
20	Erik, D., Briand, L. C., & Adsholm, W. J. (2008). A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance. <i>IEEE Transactions on Software Engineering</i> , 34(3), 407–432.
21	Fluri MichaelPinzger, MartinGall, Harald C., B. (2007). Change Distilling: Tree Differencing for Fine-Grained Source Code Change Extraction. <i>IEEE Transactions on Software Engineering</i> , 33(11), 725–743.

22	Franke, U., Johnson, P., & König, J. (2014). An architecture framework for enterprise IT service availability analysis. <i>Software and Systems Modeling</i> , 13(4), 1417–1445. doi:10.1007/s10270-012-0307-3
23	Hafner, M., Schelp, J., & Winter, R. (2006). Berücksichtigung des Architekturmanagements in serviceorientierten IT-Managementkonzepten am Beispiel von ITIL. In J. Schelp & R. Winter (Hrsg.), <i>Integrations-management</i> (S. 99–121). Berlin/Heidelberg: Springer.
24	HARABAGIU FINLEY, S. (2010). Using Topic Themes for Multi-Document Summarization. <i>ACM Transactions on Information Systems</i> , 28(3), 13:1–13:47.
25	Harter, D. E., Kemerer, C. F., & Slaughter, S. A. (2012). Does Software Process Improvement Reduce the Severity of Defects? A Longitudinal Field Study. <i>IEEE Transactions on Software Engineering</i> , 38(4), 810–827.
26	Heinrich, B., Klier, M., Lewerenz, L., & Zimmermann, S. (2015). Quality-of-Service-Aware Service Selection: A Novel Approach Considering Potential Service Failures and Nondeterministic Service Values. <i>Service Science</i> , 7(1), 48–69. doi:10.1287/serv.2015.0093
27	Hicks, I. V. (2005). Planar Branch Decompositions II: The Cycle Method. <i>INFORMS Journal on Computing</i> , 17(4), 413–421.
28	Hoi-Ming ChiErsoy HerbertAltinkemer, Kemal, O. K. M. (2007). Toward Automated Intelligent Manufacturing Systems (AIMS). <i>INFORMS Journal on Computing</i> , 19(2), 302–312.
29	Irani, Z., Love, P. E. D., Elliman, T., Jones, S., & Themistocleous, M. (2005). Evaluating e-government: learning from the experiences of two UK local authorities. <i>Information Systems Journal</i> , 15(1), 61–82.
30	Jørgensen Martin, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. <i>IEEE Transactions on Software Engineering</i> , 33(1), 33–53.
31	Kalle, R., & Lyytinen, J. (2000). Components of Software Development Risk: How to Address Them? A Project Manager Survey. <i>IEEE Transactions on Software Engineering</i> , 26(2), 98–112.

32	Karlsson Kai, F. (2006). Combining method engineering with activity theory: theoretical grounding of the method component concept. <i>European Journal of Information Systems</i> , 15(1), 82–90.
33	Kim, G., Shin, B., Kim, K., & Lee, H. (2011). IT Capabilities, Process-Oriented Dynamic Capabilities, and Firm Financial Performance. <i>Journal of the Association for Information Systems</i> .
34	Kim, H. -k., Kim, Y. -j., & Jung, J. -i. (2001). A Mobility Prediction Handover Algorithm for Quality of Service Improvement in Wireless ATM. In M. K. Sein (Hrsg.), (S. 119–128). New York, Kluwer Academic/Plenum, London.
35	Kim, S.-W., Park, S., & Chu, W. W. (2004). Efficient processing of similarity search under time warping in sequence databases: an index-based approach. <i>Information Systems</i> , 29(5), 405.
36	Kuk, G., & Janssen, M. (2013). Assembling infrastructures and business models for service design and innovation. <i>Information Systems Journal</i> , 23(5), 445–469.
37	Lehtonen, M. (2006). Preparing Heterogeneous XML for Full-Text Search. <i>ACM Transactions on Information Systems</i> , 24(4), 455–474.
38	Lerner StefanOsterweil, Leon J.Bendraou, RedaKannengiesser, UdoWise, Alexander, B. S. (2010). Exception Handling Patterns for Process Modeling. <i>IEEE Transactions on Software Engineering</i> , 36(2), 162–183.
39	Li, W., Liu, K., Yang, H., & Yu, C. (2014). Integrated clinical pathway management for medical quality improvement - based on a semiotically inspired systems architecture. <i>European Journal of Information Systems</i> , 23(4), 400–417.
40	Lima, A. S., de Sousa, J. N., Oliveira, J. A., Sauve, J., & Moura, A. (2010). Towards business-driven continual service improvement. In <i>2010 IEEE/IFIP Network Operations and Management Symposium Workshops</i> (S. 95–98). IEEE. doi:10.1109/NOMSW.2010.5486594
41	Maxion Robert T., R. A. O. (2000). Eliminating Exception Handling Errors with Dependability Cases: A Comparative, Empirical Study. <i>IEEE Transactions on Software Engineering</i> , 26(9), 888–906.

42	Mora, M., & Raisinghani, M. (2014). An extensive review of IT service design in seven international ITSM processes frameworks: Part I. <i>International Journal of Information Technologies and Systems Approach</i> , 7(2), 83–106.
43	Napier, N. P., Mathiassen, L., & Johnson, R. D. (2009). Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study. <i>IEEE Transactions on Software Engineering</i> , 35(5), 593–606.
44	Prakash, S., Deelman, E., & Bagrodia, R. (2000). Asynchronous Parallel Simulation of Parallel Programs. <i>IEEE Transactions on Software Engineering</i> , 26(5), 385–400.
45	Prechett Barbara, L. (2001). An Experiment Measuring the Effects of Personal Software Process (PSP) Training. <i>IEEE Transactions on Software Engineering</i> , 27(5), 465–472.
46	QUAN WANGJUN XUHANG LICRASWELL, N. (2013). Regularized Latent Semantic Indexing: A New Approach to Large-Scale Topic Modeling. <i>ACM Transactions on Information Systems</i> , 31(1), 1–44.
47	Radhakrishnan, R. (2008). <i>Enterprise Architecture & IT Service Management - ITSM Frameworks and Processes and their Relationship to EA Frameworks and Processes</i> (No. W078). San Francisco, USA.
48	Reichenbach, C., Burke, M. G., Peshansky, I., & Raghavachari, M. (2009). Analysis of imperative XML programs. <i>Information Systems</i> , 34(7), 624–642.
49	Scannapieco, M., Virgillito, A., Marchetti, C., Mecella, M., & Baldoni, R. (2004). The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems. <i>Information Systems</i> , 29(Data Quality in Cooperative Information Systems), 551–582.
50	Song Adam Foster, Jeffrey S., C. (2014). iTree: Efficiently Discovering High-Coverage Configurations Using Interaction Trees. <i>IEEE Transactions on Software Engineering</i> , 40(3), 251–265.
51	Stensrud Ingunn, E. (2003). Identifying High Performance ERP Projects. <i>IEEE Transactions on Software Engineering</i> , 29(5), 398–416.

52	Sunita, T., Li, Z., & Rudraraju, J. (2004). Evaluating Web Software Reliability Based on Workload and Failure Data Extracted from Server Logs. <i>IEEE Transactions on Software Engineering</i> , 30(11), 754–769.
53	Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review. <i>IEEE Transactions on Software Engineering</i> , 38(2), 398–424.
54	VanderMeer, D., Dutta, K., & Datta, A. (2012). A COST-BASED DATABASE REQUEST DISTRIBUTION TECHNIQUE FOR ONLINE E-COMMERCE APPLICATIONS. <i>MIS Quarterly</i> , 36(2), 479–507.
55	Vasconcelos, A., Sousa, P., & Tribolet, J. (2007). Information System Architecture Metrics: an Enterprise Engineering Evaluation Approach. <i>The Electronic Journal Information Systems Evaluation</i> , 10(1), 91–122.
56	Wohlin, C. (2001). Guest Editor's Introduction: Seventh International Software Metrics Symposium. <i>IEEE Transactions on Software Engineering</i> , 27(11), 961–962.
57	Zarnekow, R., Hochstein, A., & Brenner, W. (2005). <i>Serviceorientiertes IT-Management - ITIL-Best-Practices und -Fallstudien</i> . Berlin/Heidelberg: Springer.
58	Zhang, H. H., Leyuan, S., Meyer, R., Nazareth, D., & D'Souzaa, W. (2009). Solving Beam-Angle Selection and Dose Optimization Simultaneously via High-Throughput Computing. <i>INFORMS Journal on Computing</i> , 21(3), 427–444.
59	Zickert Roman, F. (2012). Coping with Existing Systems in Information Systems Development. <i>IEEE Transactions on Software Engineering</i> , 38(5), 1027–1039.

Lebenslauf

Persönliche Information

Name: Johannes Maximilian Ahrens
Adresse: Winsstr. 41, 10405 Berlin, Deutschland
Geboren am: 08.04.1979
Geboren in: Kiel, Deutschland
Familienstand: verheiratet, 2 Kinder

Berufserfahrung

2013 – heute Deutsche Börse Cloud Exchange AG
2013 – heute Vorstand
2007 – heute Zimory GmbH
2009 – heute Chief Product Officer, Geschäftsführer
2007 – 2009 VP Produkt, Prokurist
2005 – 2008 Deutsche Telekom Laboratories
2005 - 2008 Projektmanager, Research Scientist
2003 BMW Motorrad AG
2003 Intern Consultant
2002-2004 Deutscher Bundestag
2002 - 2004 Wissenschaftlicher Mitarbeiter im Büro von Klaus Haupt, MdB

Akademische Ausbildung

1999-2005 Technische Universität Berlin
Abschluss als Diplom Wirtschaftsingenieur
2003-2004 Libera Università Internazionale degli Studi Sociali Guido Carli (LUISS)