

RESEARCH

Sebastian Herden

Model-Driven- Configuration- Management

Ein modellgetriebener Ansatz für das
Konfigurationsmanagement von
IT-Systemlandschaften



Springer Vieweg

Model-Driven-Configuration- Management

Sebastian Herden

Model-Driven- Configuration- Management

Ein modellgetriebener Ansatz für
das Konfigurationsmanagement
von IT-Systemlandschaften

 Springer Vieweg

Sebastian Herden
Magdeburg, Deutschland

Dissertation Otto-von-Guericke-Universität Magdeburg, 2012

ISBN 978-3-658-01106-2
DOI 10.1007/978-3-658-01107-9

ISBN 978-3-658-01107-9 (eBook)

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

Springer Vieweg

© Springer Fachmedien Wiesbaden 2013

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist eine Marke von Springer DE. Springer DE ist Teil der Fachverlagsgruppe Springer Science+Business Media.
www.springer-vieweg.de

Vorwort

Den Begriff *Model-Driven-Configuration-Management* habe ich erstmalig 2008 in einer sehr angeregten Diskussion mit meinem besten Freund und Forscherkollegen André Zwanziger im SAP Center for Very Large Business Applications der Arbeitsgruppe Wirtschaftsinformatik in Magdeburg verwendet. Es wäre doch schön, wenn IT-Systemlandschaften auf einen Klick aus Modellen erzeugt werden können — zumal diese Modelle sowieso existieren. Wieso müssen IT-Abteilungen alles noch einmal erfassen und modellieren oder planen, was Fachabteilungen mit Beratern und Unternehmensarchitekten in meist aufwendigen Workshops erarbeitet haben? Warum können sie nicht sofort an die Umsetzung gehen? Es ist doch alles gesagt und schwarz auf weiß dokumentiert. Die Antwort auf diese Frage ist verblüffend einfach, wie sich im Laufe dieser vorliegenden Forschungsarbeit herauskristallisiert hat: Sie reichen nicht aus. Modelle haben einen Zweck und dieser ist bei Unternehmensarchitekturen ein anderer als bei Konfigurationsmodellen. Die einen möchten einen Überblick und vernachlässigen Details (Architektur) — die anderen möchten so viele Details wie möglich für ihren bestimmten Themenbereich (Konfiguration) und verlieren den Blick auf das große Ganze. Bei meiner Arbeit als IT-Berater und Software-Entwickler in diversen Projekten bei großen und kleinen Unternehmen wurde oft von Parallelwelten gesprochen.

Die Idee war geboren. Es gilt einen modellgetriebenen Ansatz für das Konfigurationsmanagement von IT-Systemlandschaften zu entwickeln. Hierbei soll in Anlehnung an eine modellgetriebene Softwareentwicklung der Erstellungsprozess durch automatisierte Modelltransformationen in das gewünschte Zielprodukt überführt werden. Modelle sind demnach nicht ein gutaussehendes Nebenprodukt bei der Planung und Umsetzung von IT-Systemlandschaften, sie sind kein Beweis, dass IT-Berater gearbeitet haben, sondern sie sind inhärenter Bestandteil des Prozesses selbst. Sie führen und steuern ihn. Sie stellen eine konsistente Verbindung von abstrakten Architekturen zu detaillierten Konfigurationen her. Das Ziel muss also sein, dass Konfigurationen von IT-Systemlandschaften an den Bedürfnissen der Fachabteilungen und Kundenwünschen ausgerichtet sind, dass Entscheidungen bei der Umsetzung transparent und nachvollziehbar sind und bleiben. Model-Driven-Configuration-Management ist somit der Klebstoff für beide (Parallel-)Welten.

Die erste Reaktion meiner Betreuerin PD Dr.-Ing. Susanne Patig war ungefähr: „Das ist eine sehr interessante Idee, doch sie ist mir nicht konkret genug. Ich kann mir noch nicht vorstellen, wie das funktionieren soll.“ Das war genau die richtige Motivation und auch das Leitmotiv unserer sehr guten Zusammenarbeit. An dieser Stelle möchte ich mich dafür bedanken, dass sie, nach dem plötzlichen Tod von Prof. Dr. Rautenstrauch, die Betreuung meiner Forschungsarbeit neben ihrer zusätzlichen Tätigkeit als Professorin an der Universität in Bern übernommen hat. Auch bedanke ich mich für die zwei wichtigsten Fragen: „Wie willst du das umsetzen?“ und „Welchen Nutzen hat es?“. Beides wird in der vorliegenden Arbeit aus unterschiedlichen Perspektiven beantwortet. Eine qualitative Anforderungsanalyse auf Basis von Experteninterviews und einer Literaturanalyse

bringt beide Welten zusammen und zeigt ein Konzept und den Nutzen beim Einsatz des Model-Driven-Configuration-Managements. Eine prototypische Implementierung sowie Validierung einer *Werkzeugarchitektur* und eines *Vorgebensmodells* für das Model-Driven-Configuration-Management zeigen die Umsetzung und Anwendbarkeit des Ansatzes.

Während der Forschungsarbeit haben mich viele Menschen begleitet, denen ich nun abschließend danken möchte. Zunächst möchte ich sämtlichen Interviewpartnern danken, die ich jedoch an dieser Stelle nicht namentlich nennen kann. Die Implementierung und Validierung ist während eines längeren Forschungsaufenthaltes im SAP Research CEC in Belfast (UK) und bei der IBM Rational Software Group in Raleigh (USA) entstanden. An dieser Stelle einen besonderen Dank an André Zwanziger, Philip Robinson, Tariq Ellahi, Chad Holliday und Timothy McMackin. Für spannende und tiefe Einblicke in die Welt von Unternehmensarchitekten möchte ich Bernd Bergmann danken. Wir haben 10 Monate zusammen in einem Beratungsprojekt gearbeitet, welches die eigentliche thematische Motivation für die Beschäftigung mit Unternehmensarchitekturen war. Dies war auch der Auslöser für den Entschluss, die Forschungsarbeit an der Otto-von-Guericke-Universität Magdeburg zu beginnen. Ich danke neben meiner Betreuerin PD Dr.-Ing. Susanne Patig den weiteren Gutachtern Prof. Dr. Gunter Saake und Prof. Dr. Jan-Marco Leimeister für die Begutachtung dieser Arbeit sowie für spannende Fragen und Anregungen. Weiterer Dank gilt den weiteren Mitgliedern der Promotionskommission Prof. Dr. Jürgen Dassow und Prof. Dr. Klaus Turowski. Allen zusammen gilt der Dank dafür, dass mir die Verteidigung meiner Dissertation sehr viel Freude bereitet hat. Außerdem danke ich für die Zusammenarbeit mit den Kollegen der SAP Center for Very Large Business Application an der TU München und der Otto-von-Guericke-Universität Magdeburg.

Das Schreiben einer Dissertationsschrift erfordert viel Zeit und Geduld – verlangt viele Opfer ab. Davon können vor allem Freunde und Familie berichten. Ich danke daher insbesondere meiner Frau Nadine, meinen Eltern Birgit und Hannjörg, meinem Bruder Tobias sowie meinen Großeltern Evelin und Eberhardt, für den beständigen Glauben an mich, auch in Tiefphasen, für die Geduld und für die Zeit beim Lesen und Zuhören. Diese Zeit dürft ihr nun mit Recht einfordern.

Abschließend wünsche ich viel Spaß und Erkenntnisgewinn beim Lesen der Arbeit. Ich denke, der Einsatz von IT in Unternehmen wird im Kern immer ein Ringen mit der Komplexität von Systemen sein. Dieses Ringen kann aber durch automatisierte Werkzeuge, systematisches und methodisches Vorgehen verbessert werden. Die Überführung von Unternehmensstrategien und Geschäftsprozessen in nutzenbringende IT-Systemlandschaften kann nur im Rahmen einer ingenieurmäßigen Disziplin wie das *System Landscape Engineering* erfolgen. Das *Model-Driven-Configuration-Management* leistet hierbei einen kleinen Beitrag.

Sebastian Herden
Magdeburg, 10. Oktober 2012

Inhaltsverzeichnis

Abbildungsverzeichnis	xi
Tabellenverzeichnis	xiii
Abkürzungsverzeichnis	xv
1 IT-Systemlandschaften im Spannungsfeld zwischen Strategie und Betrieb	1
2 Unternehmensarchitekturen und IT-Konfigurationen	8
2.1 Enterprise-Architecture-Management (EAM)	8
2.1.1 Unternehmensarchitektur und Management	8
2.1.2 Enterprise-Architecture-Management-Frameworks	14
2.1.3 Bedeutung für die Gestaltung von IT-Systemlandschaften	18
2.2 IT-Konfigurationsmanagement (ITKM)	20
2.2.1 Konfiguration und Management	20
2.2.2 Bedeutung für die Gestaltung von IT-Systemlandschaften	22
2.3 Zusammenhang	26
2.4 Zusammenfassung	28
3 Empirische Anforderungsanalyse für die Integration von EAM und ITKM	29
3.1 Gewählter Ansatz für die empirische Anforderungsanalyse	29
3.2 Datenerhebung und Auswertung	33
3.2.1 Interviewleitfaden	33
3.2.2 Auswahl der Interviewpartner	35
3.2.3 Qualitative Inhaltsanalyse	37
3.3 Zusammenfassung der Interviewergebnisse	40
3.3.1 Enterprise-Architecture-Management	40
3.3.2 Dokumentation der Unternehmensarchitektur und der IT-Systemlandschaft	44
3.3.3 Änderungen an der Unternehmensarchitektur	49
3.3.4 Umsetzung im IT-Betrieb	51
3.3.5 Probleme und Lösungen	56
3.4 Anforderungen an eine Integration von EAM und ITKM	60
3.4.1 Rahmenbedingungen	60
3.4.2 Qualitätsanforderungen	62
3.4.3 Funktionale Anforderungen	63
3.5 Kritische Diskussion der Untersuchung	65
3.5.1 Objektivität der Erhebung	65
3.5.2 Reliabilität der Erhebung	66

3.5.3	Validität der Erhebung und Analyse	66
3.6	Zusammenfassung	68
4	Konzept des Model-Driven-Configuration-Managements (MDCM)	69
4.1	Motivation für einen modellgetriebenen Ansatz	69
4.1.1	Model-Driven-Engineering	69
4.1.2	Lösungsansatz des MDCM	74
4.2	Modelle und Transformationen	79
4.2.1	Metamodell des MDCM	79
4.2.2	Modell-Transformationen des MDCM	83
4.3	Konzeption eines MDCM-Vorgehensmodells	85
4.3.1	Herleitung des MDCM-Vorgehensmodells	85
4.3.2	IT-Systemlandschaftsgrobplanung	89
4.3.3	IT-Systemlandschaftsfeinplanung	91
4.3.4	Beschaffung	93
4.3.5	Inbetriebnahme	93
4.4	Konzeption einer MDCM-Werkzeugarchitektur	94
4.4.1	Überblick zur MDCM-Werkzeugarchitektur	94
4.4.2	Aufbau eines MDCM-Clients	95
4.4.3	Aufbau der MDCM-Repository-Komponente	96
4.4.4	Aufbau der MDCM-Automation-Engine-Komponente	97
4.4.5	Aufbau der MDCM-Analyzer-Komponente	98
4.5	Zusammenfassung	99
5	Prototypische Implementierung und Validierung der Ergebnisse	101
5.1	Ansatz der prototypischen Implementierung und Validierung	101
5.1.1	Zielstellung und Anforderungen	101
5.1.2	Lösungsmöglichkeiten und gewählter Ansatz	102
5.2	Szenario: Planung und Aufbau von SAP-Systemlandschaften	109
5.3	Implementierung und Anwendung des MDCM-Werkzeugs	113
5.3.1	Domänenspezifische Spracherweiterung	113
5.3.2	Planung und Aufbau von IT-Systemlandschaften	127
5.3.3	Modellanalyse	137
5.4	Beurteilung der Ergebnisse	142
5.4.1	Einschätzung der prototypischen Implementierung	142
5.4.2	Einschätzung der Anwendbarkeit	145
5.5	Zusammenfassung	149
6	Zusammenfassung und Ausblick	150
6.1	Ergebnisse und kritische Würdigung	150
6.2	Ausblick und weiterer Forschungsbedarf	154
A	Experteninterviews und Auswertung	156
A.1	Transkriptionen der durchgeführten Experteninterviews	156
A.1.1	Transkription Interview 1	156
A.1.2	Transkription Interview 2	165
A.1.3	Transkription Interview 3	173
A.1.4	Transkription Interview 4	183
A.1.5	Transkription Interview 5	191
A.2	Extrahiertes Kategoriensystem	203

A.3	Zwischenergebnisse der qualitativen Inhaltsanalyse	210
A.3.1	Zwischenergebnis Interview 1	210
A.3.2	Zwischenergebnis Interview 2	212
A.3.3	Zwischenergebnis Interview 3	214
A.3.4	Zwischenergebnis Interview 4	217
A.3.5	Zwischenergebnis Interview 5	219
A.4	Ergebnisse der qualitativen Inhaltsanalyse	222
A.4.1	Ergebnis Themenkomplex A: Enterprise-Architecture-Management	222
A.4.2	Ergebnis Themenkomplex B: Dokumentation der Unternehmensarchitektur und IT-Systemlandschaft	227
A.4.3	Ergebnis Themenkomplex C: Änderungen der Unternehmensarchitektur	231
A.4.4	Ergebnis Themenkomplex D: Umsetzung im IT-Betrieb	233
A.4.5	Ergebnis Themenkomplex E: Probleme und Lösungen	237
B	Anhang zum MDCM-Konzept	240
B.1	Abdeckung der Prinzipien durch Komponenten	240
B.2	Unterstützung der Projektphasen durch Komponenten	241
C	Anhang zur prototypischen Implementierung	242
C.1	Beschreibung der Spracherweiterung	242
C.1.1	Erstellte Units für SAP-Systemlandschaften	242
C.1.2	Erstellte Units für SAP-System-Konfigurationen	243
C.2	Transformator für Spracherweiterungen	244
C.2.1	Generieren der Schema Definition	244
C.2.2	Generieren der Datei plugin.xml	246
C.2.3	Generieren der Modellierungsmuster	248
C.3	Beispielszenario	249
C.4	Extension-Point: SAP-Transformation	252
C.5	Extension-Point: Metrics Analyzer	253
	Literaturverzeichnis	256

Abbildungsverzeichnis

2.1	Konzeptuelles Modell von Architekturbeschreibungen	9
2.2	Gestaltungsebenen und -objekte von Unternehmensarchitekturen	11
2.3	Klassifikationsschema für EAM-Frameworks	15
2.4	Zeitliche Entwicklung von TOGAF, Zachman, DoDAF und FEA	18
2.5	Vergleich untersuchter Frameworks	19
2.6	Detaillierungsebenen von Architekturbeschreibungen	20
2.7	Aufgabengebiete des Konfigurationsmanagements	21
2.8	Zusammenhang zwischen Betrachtungsgegenständen, KM-Aufgaben und Lebenszyklusphasen	23
2.9	IT-Service-Management-Prozesse	23
2.10	Unternehmensarchitektur zur IT-Konfiguration	27
3.1	Rekonstruierte Funktionsweise der Übersetzung	61
4.1	Merkmale des MDE auf Basis der MDA	71
4.2	Transformationskette des MDCM	75
4.3	MDCM-Werkzeugarchitektur	78
4.4	MDCM-Metamodell Abgrenzung zur Architektur	79
4.5	MDCM-Konfigurationselemente	80
4.6	MDCM-Metamodell Beziehungen und Beschreibung	80
4.7	MDCM-Metamodell Sichtweisen	81
4.8	Modelle und Transformationen	84
4.9	Phasenmodell des MDCM	88
4.10	IT-Architektur, IT-Konfiguration und IT-Systemlandschaft	90
4.11	Aktivitäten der Phase IT-Systemlandschaftsgrobplanung	91
4.12	Aktivitäten der Phase IT-Systemlandschaftsfeinplanung	92
4.13	Aktivitäten der Phase Beschaffung	93
4.14	MDCM-Werkzeugarchitektur (UML)	94
4.15	MDCM Client	95
4.16	MDCM Repository	97
4.17	MDCM Automation Engine	98
4.18	MDCM Analyzer	99
5.1	Überblick IBM Rational Software Architect	103
5.2	Kernelemente IBM Deployment Architecture Platform	104
5.3	Abhängigkeiten IBM Deployment Architecture Platform	104
5.4	SAP System Landscape (Archimate)	111
5.5	Metamodell SAP-Systemlandschaft	113

5.6	Metamodell SAP-Instanzen I	115
5.7	Metamodell SAP-Instanzen II	115
5.8	Entwicklungsansatz Spracherweiterung	117
5.9	Benutzeroberfläche für die Spracherweiterung	118
5.10	UML-Sequenzdiagramm DomainConfigurationReader	119
5.11	UML-Sequenzdiagramm DomainConfigurationGenerator I	120
5.12	UML-Sequenzdiagramm DomainConfigurationGenerator II	122
5.13	UML-Sequenzdiagramm DomainConfigurationGenerator III	124
5.14	Beispiel Ergebnis der M2M-Transformation	130
5.15	Beispiel SAP-BI-Systemlandschaft (Ausschnitt)	131
5.16	Distributed Topology	131
5.17	Automatischer Abgleich von Modellen (MDCM-Automation Engine)	133
5.18	SAP Deployment Dialog (MDCM-Automation Engine)	136
5.19	Beispiel für Validation and Resolution	138
5.20	Dialog zum Zusammenstellen von Analysemodellen	139
5.21	Ergebnisdarstellung der Modellanalyse	139
5.22	Abgrenzung zu themenverwandten Arbeiten	148

Tabellenverzeichnis

1.1	Untersuchungsfragen	4
2.1	Elemente einer Sichtweisenexplikation	10
2.2	Abdeckung der Konfigurationsmanagementaufgaben	25
3.1	Gütekriterien qualitativer und quantitativer Forschung	31
3.2	Allgemeines Ablaufmodell der Untersuchung	33
3.3	Verwendeter Interviewleitfaden	34
3.4	Typisierung der Interviewpartner	36
3.5	Transkriptionsregeln	38
3.6	Analysegegenstände der Auswertung	39
3.7	Ergebnisse Themenkomplex A	40
3.8	Ergebnisse Themenkomplex B	44
3.9	Ergebnisse Themenkomplex C	49
3.10	Ergebnisse Themenkomplex D	51
3.11	Ergebnisse Themenkomplex E	56
3.12	Rahmenbedingungen bei Integration	60
3.13	Qualitätsanforderungen bei Integration	63
3.14	Funktionale Anforderungen bei Integration	64
4.1	Vergleich von MDE, CASE und EAM	73
4.2	MDCM-Prinzipien	74
4.3	Spezifikation der Datenflüsse	78
4.4	Sichtweisen des MDCM	82
4.5	Wiederholungszyklen	91
5.1	Abbildung der Metamodelle MDCM und IBM DAP	105
5.2	Umsetzung der MDCM-Komponenten	106
5.3	Phasen MDCM-Vorgehensmodell	108
5.4	Vorgehen bei der Fallstudie	112
5.5	SAP Usage Types	114
5.6	SAP-Instanzen, SAP-Instanz-Komponenten und Abhängigkeiten	115
5.7	Abbildungsrelation Archimate zu Topology	127
5.8	Metriken des MDCM Metrics Analyzer	140
5.9	Messergebnisse komplexe Modellierungsmuster	146
6.1	Untersuchungsfragen	150
A.1	Extrahiertes Kategoriensystem	203

A.2	Zwischenergebnis Interview 1	210
A.3	Zwischenergebnis Interview 2	212
A.4	Zwischenergebnis Interview 3	214
A.5	Zwischenergebnis Interview 4	217
A.6	Zwischenergebnis Interview 5	219
A.7	Ergebnisse für Themenkomplex A	222
A.8	Ergebnisse für Themenkomplex B	227
A.9	Ergebnisse für Themenkomplex C	231
A.10	Ergebnisse für Themenkomplex D	233
A.11	Ergebnisse für Themenkomplex E	237
B.1	Zusammenhang MDCM-Komponenten und Prinzipien	240
B.2	Zusammenhang MDCM-Komponenten und Phasen	241
C.1	Units SAP-Systemlandschaft	242
C.2	Units SAP-Deployment	243

Abkürzungsverzeichnis

BI	Business Intelligence
BPMN	Business Process Modeling Notation
BW	Business (Data) Warehouse
CASE	Computer Aided Software Engineering
CI	Configuration Item
CMDB	Configuration Management Database
CMS	Configuration Management System
COM	Conceptual Operational Model
DAP	IBM Deployment Architecture Platform
DSL	Domain Specific Language
EA	Enterprise Architecture
EAF	Enterprise Architecture Framework
EAM	Enterprise Architecture Management
EBS	(Oracle) E-Business Suite
EJB	Enterprise Java Bean
EMF	Eclipse Modeling Framework
FA	Funktionale Anforderung
FTE	Full Time Equivalent
GMF	Graphical Modeling Framework
IaaS	Infrastructure-as-a-Service
IBM	International Business Machines Corporation
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IT	Informationstechnik
ITIL	IT Infrastructure Library
ITO	IT Operations
ITSM	IT-Service-Management
JET	Java Emitter Templates
KM	Konfigurationsmanagement
M2C	Model-to-Code oder Model-to-Configuration
M2M	Model-to-Model
MDA	Model Driven Architecture
MDCM	Model-Driven-Configuration-Management
MDE	Model Driven Engineering
MDSM	Model Driven Software Development
MOF	Meta Object Facility
OCL	Object Constraint Language

PaaS	Platform-as-a-Service
POM	Physical Operational Model
QA	Qualitätsanforderung
QVT	Query/Views/Transformation
RB	Rahmenbedingung
RSA	IBM Rational Software Architect
SaaS	Software-as-a-Service
SAP	Systeme, Anwendungen, Produkte in der Datenverarbeitung
SOM	Specification Level of Operational Model
TCO	Total Cost of Ownership
TOGAF	The Open Group Architecture Framework
UML	Unified Modeling Language
VLBA	Very Large Business Application
XML	Extensible Markup Language
XSD	XML Schema Definition

Abstract

Enterprise architecture management (EAM) is the fundamental base to strategically design and control IT in organisations. IT service management (ITSM) constitutes a special IT domain, which provides IT services by operating an IT system landscape. A basic prerequisite for using IT services is, therefore, to configure the underlying IT system landscape (IT configuration management). While EAM addresses the logical design of an IT system landscape, IT configuration management supports the construction of a planned enterprise architecture for IT operations. Whereas fundamental elements of an enterprise architecture are considered in an aggregated manner at Enterprise Architecture Management (EAM), IT Operations (ITO) models are focused on specific systems or system components and are, therefore, considered in detail. Hence, a semantic gap exists due to different focal points of both disciplines (EAM — strategic; ITSM — operational). A seamless integration seems, thus, to be necessary to transform EAM models into IT system landscape models and finally to implement these models.

In software development model driven engineering approaches (MDE) are used to separate general application knowledge from specific implementation techniques. Here, defined transformers create software products automatically. Although common frameworks for enterprise architecture management (EAM) propose predefined artifacts and methods for this purpose, the transition between the design stage and the operating stage of software takes place manually. Time-consuming discovery procedures capture the current IT configuration after the transition, store it in configuration databases and allow further analysis on the basis of reconstructed models. This manual process inherits the danger that the quality of solutions is directly dependent on technician's individual knowledge. Additionally, due to potentially insufficient integration the resulting IT configuration may not be aligned with business requirements. Furthermore, design is perhaps not transparent and not traceable.

The outcome of this thesis is the concept of model driven configuration management (MDCM). The fundamental idea is to use a MDE approach to seamlessly integrate enterprise architecture management and IT configuration management for IT operations. Hence, the objective is that the resulting IT configuration is aligned with business processes, designed transparently and traceably as well as quality is no longer dependent on individual knowledge.

Model driven configuration management (MDCM) contributes to model driven engineering, configuration management, enterprise architecture management and data center automation as well as IT service management. In detail, the presented research provides the following results: (1) a collection of requirements for integrating enterprise architecture management and configuration management of IT system landscapes and IT infrastructures, (2) a MDE-based process model for the automated configuration of IT system landscapes and IT infrastructures, (3) a validated MDCM tool architecture for the use in EAM and IT operations.

Model Driven Engineering, Configuration Management, Enterprise Architecture Management, IT system landscape, IT infrastructure

Zusammenfassung

Enterprise-Architecture-Management (EAM) stellt die Basis für die strategische Gestaltung und Steuerung der unternehmensweiten IT dar. Das IT-Service-Management (ITSM) dagegen ist eine spezialisierte IT-Domäne, welche IT-Dienstleistungen (IT-Services) durch den Betrieb von IT-Systemlandschaften (IT-Betrieb) zur Verfügung stellt. Bevor IT-Dienstleistungen genutzt werden können, muss die darunter liegende IT-Systemlandschaft konfiguriert werden. Während EAM das logische Design einer IT-Systemlandschaft adressiert, unterstützt das IT-Konfigurationsmanagement im IT-Betrieb die Konstruktion einer geplanten Unternehmensarchitektur. So müssen Beschreibungen von Unternehmensarchitekturen in konkrete Konfigurationen der IT-Systemlandschaft überführt werden. Traditionell ist die IT-Konfiguration im IT-Service-Management positioniert. So entsteht eine semantische Lücke, welche durch die unterschiedlichen Schwerpunkte beider Disziplinen zu erklären ist (EAM – strategisch; ITSM – operativ). Demnach erscheint eine nahtlose Integration notwendig, um Modelle von Unternehmensarchitekturen in Modelle von IT-Systemlandschaften zu überführen und diese letztendlich zu realisieren.

In der Softwareentwicklung werden Model-Driven-Engineering-Ansätze (MDE) verwendet, um allgemeines Anwendungswissen von spezifischen Implementierungstechniken zu trennen. Durch Transformatoren werden so automatisiert Software-Produkte erstellt. Rahmenwerke des Enterprise-Architecture-Managements (EAM) schlagen für die Umsetzung von Unternehmensarchitekturen in IT-Systemlandschaften zwar vordefinierte Artefakte und Vorgehensweisen vor. Die Migration und Transition aus dem Entwurf in den eigentlichen Betrieb erfolgt aber manuell. Erst aufwendige Discovery-Läufe erfassen die aktuelle Konfiguration im Nachhinein, speichern diese in Konfigurationsdatenbanken und lassen weitere Analysen für Änderungen auf Basis rekonstruierter Modelle zu. Zusätzlich ist die finale Umsetzung einer IT-Systemlandschaft damit abhängig vom individuellen Wissen von Personen des IT-Betriebs über die eingesetzten Technologien und deren Nutzung.

In dieser Arbeit wird das Konzept des Model-Driven-Configuration-Managements (MDCM) zur Schließung der oben genannte semantischen Lücke gezeigt. Die grundlegende Idee des vorgestellten MDCM-Konzepts zur nahtlosen Integration von Enterprise-Architecture-Management und IT-Konfigurationsmanagement im IT-Betrieb ist es, einen Model-Driven-Engineering-Ansatz zu entwickeln. Es wird das Ziel verfolgt, dass die resultierende IT-Konfiguration an den Geschäftsprozessen eines Unternehmens ausgerichtet, nachvollziehbar und transparent gestaltet und die Qualität unabhängig von individuellem Wissen ist.

Damit liefert diese Arbeit Beiträge in den Bereichen Model-Driven-Engineering, Konfigurationsmanagement, Enterprise-Architecture-Management, Rechenzentrumsautomatisierung und IT-Service-Management. Es werden insbesondere folgende Arbeitspakete bearbeitet: (1) Sammlung von Anforderungen für die Integration von EAM und Konfigurationsmanagement von IT-Systemlandschaften und IT-Infrastrukturen, (2) Entwicklung eines Vorgehensmodells auf Basis von MDE Techniken für die automatische Konfiguration von IT-Systemlandschaften, (3) Erstellung und Validierung einer MDCM- Werkzeugarchitektur für die Anwendung des MDCM im Bereich des EAM und IT-Betriebs.

Model Driven Engineering, Konfigurationsmanagement, Enterprise Architecture Management, IT-Systemlandschaft, IT-Infrastruktur

1

Kapitel 1

IT-Systemlandschaften im Spannungsfeld zwischen Strategie und Betrieb

Ein integrierter Ansatz zur Abstimmung von Unternehmensstrategie, Geschäftsprozessen und IT ist eine wesentliche Voraussetzung für den erfolgreichen Einsatz von IT in der aktuellen Geschäftswelt [Han11, 66 ff.; Ven94, 73 ff.]. IT wird nicht nur als reiner Kostenfaktor sondern vielmehr als Erfolgsfaktor angesehen und sollte daher an der Unternehmensstrategie ausgerichtet sein [Han11, 66; Zim08, 357; Lan05a, 10 ff.]. Die Bedeutung der IT für den Unternehmenserfolg zeigt sich vor allem dann, wenn IT ausfällt oder mangelhaft funktioniert. So bereitete der Ausfall des Buchungs- und Informationssystem für einen Tag im Rechenzentrum der Deutschen Bahn AG¹ im Jahr 2009 einen finanziellen Schaden von ca. 500.000 EUR [BM09; DT09]. Aber auch eine mit der Unternehmensstrategie und den Geschäftsprozessen schlecht abgestimmte IT-Systemlandschaft hat Auswirkungen auf den Unternehmenserfolg. So meldete die Firma Schiesser im Jahr 2009 Insolvenz an. Grund dafür waren finanzielle Probleme aus dem Lizenzgeschäft für andere Markenhersteller durch zu kleine Stückzahlen und auf Grund einer schlechten Lieferzuverlässigkeit. Letztere wurde allerdings durch massive Logistikprobleme durch die Einführung eines neuen IT-Systems maßgeblich mitverursacht [Sen04; Kes09]. Zudem ist die Komplexität von IT-Systemlandschaften in den letzten Jahren durch die Menge an verarbeiteten Daten, durch den Einsatz unternehmenskritischer Anwendungen und einer stärker an Geschäftsanforderungen ausgerichteten Informationsversorgung gestiegen [Sym10, 8]. Diese Komplexität führt zudem zu hohen Administrationsaufwänden, welche sich ebenfalls auf die Gesamtkosten der IT auswirken können [TSO07b, 94 ff.]. Andererseits wird die Unternehmens-IT

¹ Auch wenn nicht ausdrücklich gekennzeichnet, sind Marken und Namen Eigentum der jeweiligen Organisationen.

ebenfalls als „Enabler“ angesehen, um innovative Unternehmensstrategien und Organisationsstrukturen zu ermöglichen [Ven94, 73 ff.]. Auf Grund dieser gegenseitigen Abhängigkeit ist eine transparente und nachvollziehbare Abstimmung von Unternehmensstrategie, Geschäftsprozessen und der unternehmensweiten IT notwendig und folglich die effiziente Gestaltung der unternehmensweiten IT-Systemlandschaft [Ven94, 32 ff.]. Unter einer unternehmensweiten IT-Systemlandschaft werden sämtliche betrieblichen Anwendungssysteme und die darunter liegende IT-Infrastruktur bezeichnet, die zur Leistungserbringung direkt oder indirekt benötigt werden. Als IT-Infrastruktur werden sämtliche materiellen und immateriellen Güter bezeichnet, die den Betrieb von Anwendungssoftware ermöglichen. Dabei können eine technische Sicht (*IT-Infrastruktur im engeren Sinne*) und die Sicht des Informationsmanagements (*IT-Infrastruktur im weiteren Sinne*) unterschieden werden. Letztere schließt neben den technischen Elementen der technischen Sicht ebenfalls institutionelle und personelle Gegebenheiten mit ein [PZH11]. In dieser Arbeit wird IT-Infrastruktur weiterhin aus der technischen Sicht betrachtet.

Im Bereich der *IT-Governance* (ganzheitliches Management von IT-Strukturen und IT-Prozessen [Pfr08, 489 ff.]) stellt *Enterprise-Architecture-Management (EAM)* die Basis für die strategische Steuerung und Gestaltung der unternehmensweiten IT dar [Nie06, 29]. Auf der anderen Seite ist das *IT-Service-Management (ITSM)* eine spezialisierte IT-Domäne, welche IT-Dienstleistungen (IT-Services) durch den Betrieb von IT-Systemlandschaften (IT-Betrieb) zur Verfügung stellt [Bon07, 16 ff.]. Bevor IT-Dienstleistungen genutzt werden können, muss die darunter liegende IT-Systemlandschaft konfiguriert werden. Während EAM das logische Design einer IT-Systemlandschaft adressiert, unterstützt das IT-Konfigurationsmanagement im IT-Betrieb die Konstruktion einer geplanten Unternehmensarchitektur. So müssen Beschreibungen von Unternehmensarchitekturen in konkrete Konfigurationen der IT-Systemlandschaft überführt werden. Traditionell ist die IT-Konfiguration im IT-Service-Management positioniert. So entsteht eine semantische Lücke, welche durch die unterschiedlichen Schwerpunkte beider Disziplinen zu erklären ist (EAM: strategisch; ITSM: operativ). Demnach erscheint eine nahtlose Integration notwendig, um Modelle von Unternehmensarchitekturen in Modelle von IT-Systemlandschaften zu überführen und diese letztendlich zu realisieren.

Hierfür werden in der Softwareentwicklung *Model-Driven-Engineering-Ansätze (MDE)* verwendet, um allgemeines Anwendungswissen von spezifischen Implementierungstechniken zu trennen. Durch Transformatoren werden so automatisiert Software-Produkte erstellt [CH03, 2 ff.]. Die Migration und Transition aus Unternehmensarchitekturen in den eigentlichen Betrieb von IT-Systemlandschaften erfolgt dagegen manuell. Erst aufwendige Discovery-Läufe erfassen die aktuelle Konfiguration im Nachhinein, speichern diese in Konfigurationsdatenbanken und lassen weitere Analysen für Änderungen auf Basis rekonstruierter Modelle zu [HP11, 1 f.]. Konfigurationsdatenbanken (*Configuration*

Management Database (CMDB) beinhalten Informationen über so genannte Konfigurationselemente von IT-Systemlandschaften [TSO07b, 84–85]. Zusätzlich ist die finale Umsetzung einer IT-Systemlandschaft damit abhängig vom individuellen Wissen von Personen des IT-Betriebs über die eingesetzten Technologien und deren Nutzung. In dieser Arbeit wird das Konzept des Model-Driven-Configuration-Managements (MDCM) gezeigt. Die **grundlegende Idee** des vorgestellten MDCM-Konzepts zur nahtlosen Integration von Enterprise-Architecture-Management und IT-Konfigurationsmanagement im IT-Betrieb ist es, einen Model-Driven-Engineering-Ansatz zu entwickeln. Dabei wird das **Ziel** verfolgt, dass die resultierende IT-Konfiguration an den Geschäftsprozessen eines Unternehmens ausgerichtet, nachvollziehbar und transparent gestaltet und die Qualität unabhängig von individuellem Wissen ist.

Die **Forschungsfrage** für die vorliegende Arbeit lautet demnach: *Wie kann durch die Verwendung eines Model-Driven-Engineering-Ansatzes die identifizierte semantische Lücke zwischen Enterprise-Architecture-Management und IT-Konfigurationsmanagement im IT-Betrieb geschlossen werden, damit die Verteilung von Software in einer Anwendungssystemlandschaft und die Konfiguration der darunterliegenden IT-Infrastruktur automatisiert und nachvollziehbar erfolgen kann?*

Das in dieser Arbeit entwickelte Model-Driven-Configuration-Management liefert daher **Beiträge** in den Bereichen Model-Driven-Engineering, IT-Konfigurationsmanagement, Transition von Unternehmensarchitekturen, Rechenzentrumsautomatisierung und IT-Service-Management.

Beim in dieser Arbeit vorgeschlagenen Model-Driven-Configuration-Management werden Techniken des Model-Driven-Engineerings für die Realisierung von Anforderungen an die Konfiguration von IT-Systemlandschaften verwendet. Das bedeutet, hier wird die Perspektive der Entwicklung und Bereitstellung einer adäquaten IT-Systemlandschaft eingenommen. Auch wenn neuere Entwicklungen einen Trend abzeichnen, dass zukünftig vermehrt IT-Services bei externen Dienstleistern eingekauft werden (z.B. Cloud-Computing), so zeigt sich, dass zur Zeit 80% der Unternehmen diesem Trend nicht folgen und eine Bereitstellung von IT-Systemlandschaften auf dem „klassischen“ Weg vollziehen [Sta11, 24 f.]. Deswegen wird in dieser Arbeit vornehmlich der „klassische“ IT-Betrieb von IT-Systemlandschaften betrachtet. Konsequenzen von Entwicklungen wie z.B. Cloud-Computing für die Anwendung eines Model-Driven-Configuration-Managements werden trotzdem im Ausblick dieser Arbeit benannt.

Um das Ziel dieser Arbeit zu erreichen, sind die Erkenntnisse der Forschungsarbeit durch verschiedene Forschungsmethoden untersucht worden. Somit kann die Validität der Forschungsergebnisse aus den jeweiligen Sichtweisen der Forschungsmethoden sichergestellt

werden. Diese Arbeit unterliegt dem Design–Science–Ansatz und ist daher konstruktionsorientiert aufgebaut [WH07, 28]. Zur Beantwortung der oben genannten Forschungsfrage, wird die vorliegende Arbeit von folgenden *Untersuchungsfragen* (U1–U4 – s. Tabelle 1.1) geleitet.

Tabelle 1.1: *Untersuchungsfragen*

Nr.	<i>Untersuchungsfrage</i>
U1	Welche Anforderungen für eine nahtlose Integration ergeben sich aus den unterschiedlichen Perspektiven des Enterprise–Architecture–Managements und des IT–Konfigurationsmanagements im IT–Betrieb?
U2	Wie können Modelle des EAM automatisiert in IT–Konfigurationsmodelle des IT–Betriebs durch Techniken des Model–Driven–Engineerings transformiert werden?
U3	Wie muss eine unterstützende Entwicklungsplattform für das MDCM aufgebaut sein?
U4	Wie kann eine Entwicklungsplattform für das MDCM so genutzt werden, dass sie für den Umsetzungsprozess von einer strategischen Unternehmensarchitektur hin zur Konfiguration von IT–Systemlandschaften anwendbar ist?

Zunächst interessiert, welche Gründe für die identifizierte semantische Lücke zwischen Enterprise–Architecture–Management und IT–Konfigurationsmanagement im IT–Betrieb ursächlich sind. Durch die Analyse dieser Problemstellung aus beiden Perspektiven ergeben sich Anforderungen an die in der Forschungsfrage gestellte Zielstellung zur Schließung dieser Lücke (U1). Um die potenziellen Vorteile durch Automatisierung im Rahmen eines Model–Driven–Engineering–Ansatzes aus der Softwareentwicklung zu übertragen, ist zudem von Interesse, welche Techniken des MDEs für die nahtlose Integration von Enterprise–Architecture–Management und IT–Konfigurationsmanagement im IT–Betrieb eingesetzt werden können (U2). Um zu zeigen, wie die Möglichkeiten des MDE zur Schließung der semantischen Lücke genutzt werden können, ist zu klären wie eine Werkzeugunterstützung durch eine MDCM–Entwicklungsplattform aufgebaut sein müsste (U3). Zuletzt soll untersucht und gezeigt werden, wie das Konzept des Model–Driven–Configuration–Managements genutzt werden kann, damit die Verteilung von Software in einer Anwendungssystemlandschaft und zur Konfiguration der darunterliegenden IT–Infrastruktur automatisiert und nachvollziehbar erfolgen kann (U4).

Daraus ergeben sich folgende Arbeitspakete, die in den einzelnen Kapiteln dieser Dissertation zur Beantwortung der Untersuchungsfragen bearbeitet werden (*Gang der Arbeit*):

- Im **Kapitel 2 (Unternehmensarchitekturen und IT–Konfigurationen)** werden die theoretischen Grundlagen für eine nahtlose Integration von strategischen Unternehmensarchitekturen und operativen IT–Konfigurationen hergeleitet. Ziel ist es, Anforderungen aus dem Zusammenhang der beiden Disziplinen EAM und IT–Konfigurationsmanagement zu erfassen (U1).
- Im **Kapitel 3 (Empirische Anforderungsanalyse)** wird der Stand der Praxis bei der Integration von EAM und IT–Betrieb mit Hilfe von Experteninterviews und einer qualitativen Inhaltsanalyse erfasst. Daraus werden weitere Rahmenbedingungen sowie

funktionale und qualitative Anforderungen an eine nahtlose Integration von EAM und IT-Konfigurationsmanagement im IT-Betrieb gesammelt (U1).

- Im **Kapitel 4 (Konzept des Model-Driven-Configuration-Managements (MDCM))** wird ein Lösungsansatz für eine nahtlose Integration von EAM und IT-Betrieb auf Basis des Model-Driven-Engineerings vorgestellt. Dieser Lösungsansatz setzt sich aus dem MDCM-Vorgehensmodell sowie der MDCM-Werkzeugarchitektur zusammen und basiert auf den in den vorherigen Kapiteln gesammelten Anforderungen (U2-U4).
- Im **Kapitel 5 (Prototypische Implementierung und Validierung)** wird die Anwendbarkeit des entwickelten MDCM-Vorgehensmodells und der MDCM-Werkzeugarchitektur nachgewiesen. Die Anwendbarkeit des MDCM-Konzepts wird hierbei durch eine prototypische Implementierung sowie dessen Anwendung im Rahmen einer Fallstudie zur Planung und Konfiguration komplexer IT-Systemlandschaften untersucht und bewertet (U2-U4).
- Im **Kapitel 6 (Zusammenfassung und Ausblick)** wird die Arbeit zusammengefasst, der Beitrag dieser Arbeit für das Forschungsgebiet vorgestellt, die Grenzen des Ansatzes aufgezeigt und Ideen für die zukünftige Forschung herausgestellt.

Im Forschungsgebiet des Model-Driven-Configuration-Managements existieren *verwandte Arbeiten* (Stand der Forschung). Diese können anhand folgender Kriterien im Hinblick auf die gestellte Forschungsfrage unterschieden werden:

- Wird eine **Integration von Enterprise-Architecture-Management und IT-Konfigurationsmanagement im IT-Betrieb** angestrebt?
- Wird die gesamte **IT-Systemlandschaft** betrachtet?
- Wird ein Model-Driven-Engineering-Ansatz mit **Metamodell, Transformatoren und Modellierung** vorgeschlagen?
- Wird eine **Werkzeugunterstützung** zur Automatisierung von Abläufen gezeigt?
- Wird ein **Vorgehensmodell** für die Beschreibung von Abläufen gezeigt?
- Wird eine **Implementierung** des jeweiligen Ansatzes gezeigt?
- Werden die Ergebnisse einer **Validierung** unterzogen?
- Sind die Ergebnisse **generalisierbar** und damit unabhängig von Produkten oder Einsatzgebieten?

Eine weitere Kategorie ist: *Wird Cloud-Computing eingesetzt?*, welche nicht direkt aus den Forschungsfragen abgeleitet werden kann, sondern als Unterscheidungsmerkmal aus den Arbeiten selbst identifiziert wurde.

Zunächst zeigt keine dieser Arbeiten eine *Integration von Enterprise-Architecture-Management und IT-Konfigurationsmanagement im IT-Betrieb* auf Basis eines durchgängigen

MDE-Ansatzes. In [Bet07] wird ein *generalisierbares* Entwurfsmuster mit dem Namen *Model-Driven Configuration Management* vorgeschlagen, welches beschreibt, dass Daten für das IT-Konfigurationsmanagement so früh wie möglich erfasst und automatisiert in CMDBs gespeichert werden sollen. Hier wird empfohlen, Beschreibungen von *IT-Systemlandschaften* auf Basis der Unified Modelling Language (UML) als Ausgangsmodell zu verwenden und durch XSLT-Transformatoren in CMDB-Daten zu überführen. Der *MDE-Ansatz* beschreibt zwar *Transformatoren*, lässt aber eine *Implementierung* und *Werkzeugunterstützung* offen. Zudem wird die Anwendbarkeit nicht *validiert* [Bet07, 332 ff.]. IBMs *Deployment-Topology-Model-Ansatz* [CCEK09] erweitert das Konzept der UML-basierten Verteilungsbeschreibung [OMG10, 193 ff.]. Verteilungsdiagramme visualisieren wie Systemkomponenten physikalisch umgesetzt werden und wie diese miteinander kommunizieren. Durch eine semantische Ungenauigkeit von UML-Verteilungsdiagrammen wird eine Automatisierung erschwert². IBMs Ansatz bietet ein semantisch eindeutiges *Metamodell* und einen visuellen Editor als *Werkzeugunterstützung (Implementierung)*, um genaue Verteilungsbeschreibungen für die Automatisierung von Softwareverteilungen in *IT-Systemlandschaften* zu *modellieren*. Diese Lösung ist als Patent angemeldet und daher *validiert*, allerdings zunächst nur auf die IBM-Anwendungsdomäne beschränkt [CCEK09, 4 ff.] (teilweise generalisierbar). Der Standard für verteilte Konfigurationsmanagementdatenbanken (*federated CMDB – CMDBf*) der Distributed Management Task Force (DTMF) stellt Servicedefinitionen zur Verfügung und beschreibt ein Konzept zur Integration heterogener und verteilter Datenrepositorys für das IT-Servicemanagement [DTM09, 7 ff.]. Durch die Verwendung dieses Standards können bestehende CMDBs angebunden werden. Erste prototypische *Implementierungen* und deren *Validierung* wurden in [GSV09] beschrieben und zeigen eine verteilte CMDB für Enterprise-Java-Beans-Server in IT-Systemlandschaften (Anwendungsdomäne (*Metamodell*) beschränkt – *teilweise generalisierbar*). Zusätzlich wird ein modellgetriebener Ansatz (*Transformatoren*) vorgestellt, welcher Soll/Ist-Konfigurationen für Analysen vergleichen kann [GSV09, 61 ff.]. In [LFM⁺11] wird ein Ansatz beschrieben, der darauf zielt, bestehende Anwendungen einer IT-Architektur geeignet in eine *Cloud-Umgebung* zu überführen. Damit wird teilweise das Ziel verfolgt, eine *Integration von EAM und IT-Konfigurationsmanagement im IT-Betrieb* herzustellen. Allerdings ist dieser Ansatz auf Anwendungssoftware beschränkt und deckt somit nicht die gesamte IT-Systemlandschaft ab. Es wird ein *Metamodell* angeboten, welches zur Beschreibung der Anwendungsverteilung mit Hilfe von Cloud-Computing genutzt werden kann. Zudem wird eine *Werkzeugunterstützung* zur *Modellierung* und automatischen Verteilung der Anwendung in Cloud-Umgebungen gezeigt. Der Ansatz sowie die präsentierte Werkzeugarchitektur wurden *validiert* und die *Generalisierbarkeit* in Bezug auf die Anwendungsverteilung gezeigt [LFM⁺11, 307 ff.]. Im Bereich der automatischen Bereitstellung von IT-Infrastrukturen gibt es Forschungsansätze für

² Detaillierte Ausführungen dazu befinden sich im Kapitel 4.1.

Service-Orientierte-Infrastrukturen [TOG09, 9 ff.], Service-Level-Agreement (SLA) getriebene Applikationskomposition [UML09, 369 ff.] oder, kombiniert, SLA-getriebene Service-Orientierte-Infrastrukturen [SLA11; CHA⁺10, 7 ff.]. Diese bieten zwar generalisierbare Ansätze mit Metamodellen für die Beschreibung von Infrastrukturen an, zeigen aber keine Implementierung eines MDE-Ansatzes (Transformatoren, Werkzeuge). Sie bleiben dagegen konzeptuell und ihre Anwendbarkeit ist demnach nur theoretisch validiert. Die vorgestellten Arbeiten zeigen jedoch interessante Ansätze zur Lösung des Problems zur Schließung der semantischen Lücke zwischen Enterprise-Architecture-Management und IT-Konfigurationsmanagement im IT-Betrieb. Sie werden daher beim Aufbau des MDCM-Konzepts wieder aufgegriffen und im Kapitel 6 (s. Abbildung 5.22) dem vorgestellten MDCM-Ansatz gegenüber gestellt.

Abschließend noch ein *Hinweis zur Verwendung von englischen Begriffen in dieser Arbeit*. Die Lingua Franca im Fachgebiet der Informatik und Wirtschaftsinformatik ist Englisch. Es konnte nicht vermieden werden, Begriffe im englischen Original zu verwenden. Hierbei handelt es sich meist um Fachbegriffe deren deutsche Übersetzung nicht gebräuchlich ist. Beschreibungen englischer Abbildungen sind jedoch grundsätzlich auf Deutsch, wobei der jeweilige englische Begriff in Klammern gesetzt wurde, wenn eine deutsche Übersetzung nicht ausreichend erschien.

2

Kapitel 2

Unternehmensarchitekturen und IT-Konfigurationen

2.1 Enterprise-Architecture-Management (EAM)

2.1.1 Unternehmensarchitektur und Management

Einerseits ist mit dem Begriff *Architektur* die Wissenschaft und Technik für das *Entwerfen und Entwickeln* von Systemen gemeint [Lan05a, 10; EHH⁺08, 78]. Andererseits bezeichnet Architektur die *Struktur eines Systems*, also die grundlegenden Systemelemente und deren Beziehungen zueinander und zur Systemumgebung [IEE00, 3; Sch08, 407]. Um Verwechslungen zu vermeiden, wird Architektur in dieser Arbeit als die Struktur eines Systems und als das Ergebnis eines Erstellungsprozesses (*Architektur-Management*) verstanden.

Laut ISO/IEC 42010:2007 (vormals IEEE STD. 1472:2000¹) werden Architekturen mit Hilfe von Modellen beschrieben. Diese Modelle sind durch Abstraktion gebildet und zeigen das Grundgerüst einer Systemstruktur. Mit dem ISO-Standard wird ein formaler und konzeptueller Rahmen für Architekturbeschreibungen zur Verfügung gestellt. Zudem werden damit Grundprinzipien der Architekturgestaltung definiert. In Abbildung 2.1 ist dieser Zusammenhang als Klassendiagramm der Unified Modelling Language (UML) dargestellt.

Ein *System* befindet sich in einer Umwelt (*Environment*) und wird durch diese beeinflusst und abgegrenzt. Dabei besitzt ein System eine oder mehrere Anspruchsgruppen

¹ Der IEEE Standard wurde von der ISO übernommen. Beide Standards werden seit 2007 gemeinsam durch die IEEE und ISO weiterentwickelt [ISO07].

hend von Abbildung 2.1 durch die Benennung und Beschreibung der Beziehungen zwischen *Anspruchsgruppe*, *Zweck*, *erwarteter Inhalt* und *Modellierungstechnik* (s. Tabelle 2.1).

Tabelle 2.1: Elemente einer Sichtweisenexplikation

<i>Element</i>	<i>Inhalt</i>	<i>Fragestellung</i>
Anspruchsgruppe (<i>Viewpoint zu Stakeholder</i>)	Legt den Zweck fest und bestimmt Kommunikationspartner.	<i>Für wen ist diese Sichtweise nützlich?</i>
Zweck (<i>Viewpoint zu Concern</i>)	Interessen (Concerns) werden durch adressierte Anspruchsgruppe determiniert.	<i>Warum soll eine Architekturbeschreibung aus dieser Sichtweise erzeugt werden?</i>
Erwarteter Inhalt (<i>Viewpoint zu Architectural Description</i>)	Verdeutlicht Elemente, über die Aussagen getroffen werden sollen.	<i>Was soll gezeigt werden? Welchen Bereich einer Architekturbeschreibung deckt eine, aus dieser Sichtweise erstellte, Sicht ab?</i>
Modellierungstechnik (<i>Viewpoint zu Model</i>)	Determiniert Art und Weise der Beschreibung.	<i>Wie wird eine Sicht aus dieser Sichtweise erstellt?</i>

Die Benennung der *Anspruchsgruppe* einer Sichtweise ist erforderlich, da sie einerseits den Zweck festlegt und zudem den Kommunikationspartner bestimmt. Der *Zweck* ergibt sich aus den durch die adressierte Anspruchsgruppe determinierten Interessen (*Concerns*). Die Beschreibung des *erwarteten Inhalts* verdeutlicht, über welche Elemente des zu beschreibenden Systems Aussagen getroffen werden sollen. Dies ist vor allem für die Prüfung von Bedeutung, ob eine Architekturbeschreibung sämtliche Bereiche einer Architektur umfasst. Durch die Angabe der *Modellierungstechnik* wird die Art und Weise zur Beschreibung einer Sicht determiniert. Die Beziehung von der Sichtweise zur Sicht (s. Abbildung 2.1) ist nicht Bestandteil einer Sichtweisenexplikation, da eine Sicht eine konkrete Instanz einer abstrakten Sichtweise darstellt.

Ursprünglich für die Beschreibung von Softwarearchitekturen erstellt, kann der ISO/IEC STD. 4210:2007 und dessen Definition einer Architektur auf Grund seines allgemeingültigen Charakters ebenfalls für andere Fachbereiche adaptiert werden [Sch04a, 5; Lan05a, 2ff.]. Durch die Bedeutung der Informationsversorgung in Unternehmen ist die erfolgreiche strategische Abstimmung zwischen der *Unternehmensstrategie*, den *Geschäftsprozessen* und der effizienten Gestaltung der *unternehmensweiten IT-Systemlandschaft* eine wichtige Voraussetzung für den Unternehmenserfolg (Business/IT-Alignment) [Ven94, 73ff.; PB89, 14ff.]. Wenn Unternehmen als System betrachtet werden², dann ist mit dem Begriff *Unternehmensarchitektur* (*Enterprise Architecture*) die Struktur eines gesamten Unternehmens gemeint. Eine Beschreibung der Unternehmensarchitektur stellt dabei die Gesamtsicht auf ein Unternehmen dar und beschreibt dessen grundlegende Elemente, deren Beziehungen zueinander und zur Umwelt, um die Ziele eines Unternehmens zu erreichen (Systemzweck) [ARW08, 292; Lan05a, 2; Sch04b, 13]. Sie ist somit ein Grundgerüst, bestehend

² Unternehmen können als soziotechnische Systeme angesehen werden [Rop09, 58ff.].

aus den *Abstraktionsebenen: Strategie, Organisation, Integration, Anwendungssoftware* und *IT-Infrastruktur* und ihren *Gestaltungsobjekten* (s. Abbildung 2.2) [Win03, 93f.; WF06, 2ff.; FW07, 163ff.; ARW08, 292ff.; Sch08, 400ff.].

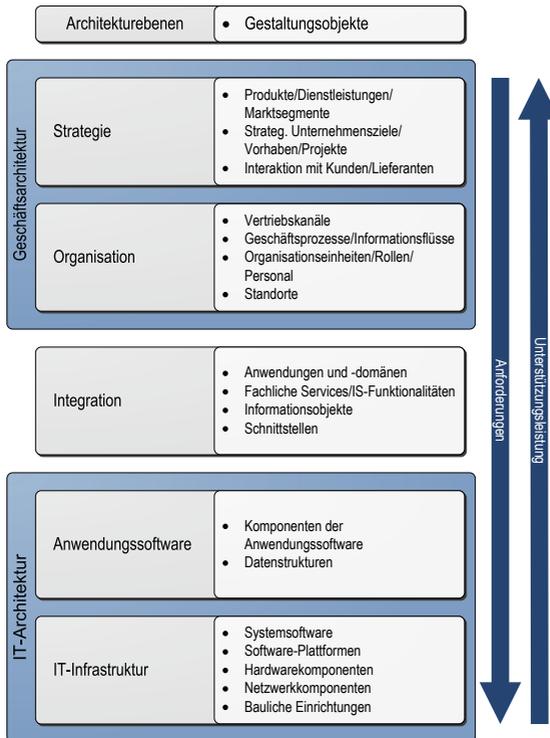


Abbildung 2.2: Gestaltungsebenen und -objekte von Unternehmensarchitekturen [in Anlehnung an ARW08, 293, Wol88, 38, HZ11, 8 und PZH11]

Nachfolgend werden die verschiedenen Gestaltungsebenen von Unternehmensarchitekturen beschrieben:

- **Strategieebene:** beschreibt die strategische Positionierung eines Unternehmens im Hinblick auf dessen Auftreten am Markt (Produkte, Dienstleistungen, Marktsegmente), strategische Unternehmensziele, Vorhaben und Projekte sowie seiner Interaktion mit Kunden und Lieferanten.
- **Organisationssebene:** enthält sämtliche organisatorischen (Organisationseinheiten, Rollen, Personal), funktionellen (Vertriebskanäle), prozessbezogenen (Geschäftsprozesse), geographischen (Standorte) und informatorischen Belange (Informationsflüsse) eines Unternehmens. Diese sind an der Unternehmensstrategie ausgerichtet.

- **Geschäftsebene:** ist die Zusammenfassung der *Strategieebene* und *Organisationsebene* und wird auch als *Geschäftsarchitektur (Business Architecture)* bezeichnet.
- **Integrationssebene:** stellt die Verbindung zwischen organisatorischer und technischer Sicht auf eine Unternehmensarchitektur dar und kapselt vor allem die Funktionalitäten der darunterliegenden Ebenen in Anwendungen und Anwendungsdomänen, Schnittstellen, Informationsobjekte sowie fachlichen Services und Funktionalität von Informationssystemen.
- **Anwendungssoftwareebene:** beschreibt sämtliche Datenstrukturen und Anwendungssoftware, um eine Geschäftsarchitektur zu unterstützen.
- **IT-Infrastruktur:** bildet die Elemente der Anwendungssoftwareebene auf Elemente der IT-Infrastruktur ab und wird auch als *Technology-Architecture* bezeichnet [TOG09, 10].
- **IT-Ebene:** fasst die *Anwendungsebene* und *IT-Infrastrukturebene* zusammen und wird daher auch als *IT-Architektur* bezeichnet.

Die jeweiligen Ebenen stellen dabei Anforderungen an die darunter liegende Ebene. Diese wiederum definieren die Unterstützungsleistungen der unteren Ebene für die darüber liegende Ebene [Wol88, 38]. Durch die Betrachtung eines Unternehmens über sämtliche Bereiche von der Geschäftsarchitektur bis zur IT-Architektur können Zusammenhänge zwischen der fachlichen und technischen Struktur der Informationsversorgung in einem Unternehmen aufgezeigt werden [WF06, 3ff.; ARW08, 292ff.]. Erst diese holistische Sicht auf die Gesamtstruktur eines Unternehmens ermöglicht die globale Ausrichtung der Teilbereiche an der Unternehmensstrategie [Lan05a, 3ff.]. Dabei hat eine Unternehmensarchitektur verschiedene Verwendungszwecke. Zum einen dient sie als *Entwurfs- und Spezifikationsgrundlage* zur Unterstützung von Architekten, um während der Architekturgestaltung Entwürfe und Spezifikationen zu erstellen sowie Designentscheidungen zu dokumentieren. Für die Unterstützung von Entscheidern wird sie als *Entscheidungsgrundlage* verwendet. Sie schafft Transparenz für bereichsübergreifende Teilarchitekturen und ermöglicht Analysen bezüglich ihrer Elemente und Verbindungen. Probleme und Konsequenzen sowie Möglichkeiten zur Veränderung lassen sich so erkennen. Als *Kommunikationsgrundlage* wird sie zur Information von Anspruchsgruppen eingesetzt, um ein generelles Verständnis über Unternehmensstrukturen zu erreichen und Unterstützung für eine geplante Unternehmensarchitektur sowie dessen Konsequenzen zu erlangen. Des Weiteren dient sie als Argumentationsgrundlage, um eventuelle Architekturgegner vom Nutzen einer geplanten Veränderung zu überzeugen [Lan05b, 168ff.; LPW⁺09, 63]. Daher ist die Beschreibung einer Unternehmensarchitektur für das Management eines Unternehmens von Bedeutung.

Ausgehend vom *funktionalen* Begriffsverständnis in der Betriebswirtschaftslehre³, sind mit

³ Im Gegensatz dazu wird vom *institutionellen Managementbegriff* gesprochen, wenn damit sämtliche mit Führungstätigkeiten betrauten Personen gemeint sind [DS03, 342ff.; SS05, 6ff.; Krc05, 23f.].