
Kompendium zu Enterprise Architect von Sparx Systems

Foundational - Advanced - Professional - Expert
Überarbeitete Auflage für die EA Version 13.5

Wir freuen uns auch über Ihre Verbesserungsvorschläge zum Handbuch. Bitte nutzen Sie dazu die angegebenen E-Mail-Adressen.

Wien, im Juni 2017

ISBN-13: 978-3-9503784-0-5

© Sparxsystems Software GmbH Wien. Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Fotokopie, Druck, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Herausgebers Sparxsystems Software GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, geändert vervielfältigt oder verbreitet werden.

Diese Unterlagen wurden mit großer Sorgfalt erstellt und geprüft. Leider können Fehler nicht ausgeschlossen werden. Der Autor übernimmt keine Verantwortung oder Haftung für fehlerhafte Angaben. Die Screenshots wurden größtenteils mit Enterprise Architect 13.5, Build 1350 mit „Microsoft® Office 2016“ als Application Look erstellt, bei Verwendung anderer Builds können sich in den Abbildungen Unterschiede ergeben.

Internet: sparxsystems.de , sparxsystems.eu

Autor



Dr. Horst Kargl beschäftigt sich seit 1998 mit objektorientierter Modellierung und Programmierung. Bevor er 2008 zu SparxSystems wechselte, war er an der TU Wien als wissenschaftlicher Mitarbeiter in der Lehre und Forschung tätig. Er forschte in mehreren Projekten an den Themen. E-Learning, Semantic Web sowie modellgetriebener Software Entwicklung. Zu letzterem dissertierte er und hat sich mit der automatischen Integration von Modellierungssprachen und Modelltransformationen beschäftigt.

2006 war er bereits als freiberuflicher Mitarbeiter für SparxSystems tätig, bevor er 2008 fix zu SparxSystems Software GmbH - Central Europa wechselte und dort als Berater und Trainer arbeitet. Seine Schwerpunkte sind Business und Software Engineering, Software Architektur, modellgetriebene Software- und System-Entwicklung, sowie die Anpassungs- und Erweiterungsmöglichkeiten von Enterprise Architect (EA).

Über diese Auflage

Mit dieser Auflage haben wir das Buch neu strukturiert und umgestaltet. Es werden drei große Schwerpunkte unterschieden, EA spezifische Themen, Modellierungssprachen und Methoden. Um den Titel „Kompendium“ gerecht zu werden, sind die einzelnen Themen als unabhängige Informationseinheiten beschrieben, um den interessierten Leser punktgenau die Informationen kurz und knapp aufbereitet zu präsentieren. Der Aufbau der einzelnen Kapitel ist so gestaltet, dass ein serielles Lesen aller Kapitel, als auch ein gezieltes Lesen einzelner Kapitel möglich ist. Referenzen zu anderen Kapitel, helfen fehlendes Wissen nachzulesen.

Wichtige und zu merkende Punkte wurden zusammengefasst und gekennzeichnet. Zu nahezu jedem Thema gibt es auch praktische Beispiele zum Nachvollziehen. Da es nicht nur eine Wahrheit gibt und oft persönliche Vorlieben eine Rolle spielen, wollten wir Ihnen diese nicht vorenthalten und haben das Buch mit Kommentaren unterschiedlicher Personen gespickt.

Der Buchtitel heißt Kompendium zu Enterprise Architect von Sparx Systems, schaut aber auch über den Tellerrand des Enterprise Architect hinaus und betrachtet auch Ansätze, Methoden und Erweiterungsmöglichkeiten, welche der EA nicht oder nicht in der Form bietet, aber mit dem EA umsetzbar sind. So lernen Sie, wie Sie den EA als Ihre Modellierungs-Plattform einsetzen können und den EA in Ihre Werkzeugkette einbauen können. Das Ziel sollte sein, dass der EA Ihr Werkzeug wird und Ihre Vorgehensweise und Prozesse unterstützt, um Ihren Modellierungsansatz umsetzen zu können.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Lesen und viel Erfolg mit Ihrem Modellierungsansatz.

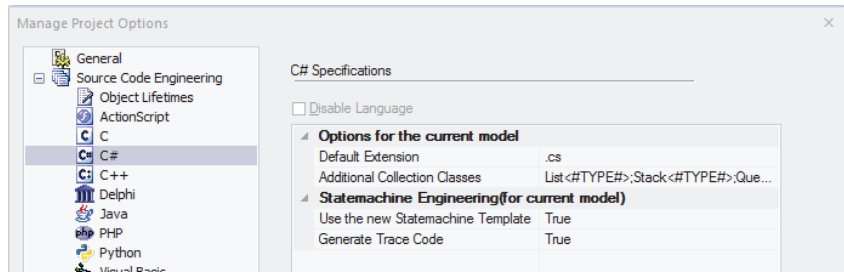
Glossar

(Element) Compartment	Ein rechteckiges Modell-Element besteht aus mehreren Abschnitten. Diese Abschnitte werden Compartments genannt. In der UML::Klasse gibt es z. B. ein Compartment für Name, Attribute, Operationen, Notiz, etc. Die Darstellung kann jeweils konfiguriert werden.
Composite Diagram	Ein Diagramm, welches von einem Composite-Element referenziert wird.
Composite-Element	Ein Modell-Element, das einen Hyperlink zu einem Composite Diagramm enthält.
Diagramm	Das Diagramm dient als Zeichenfläche in der Modell-Elemente und Konnektoren visualisiert werden. Das Modell liegt im Projekt Browser.
Diagramm-Modell-Element	Das Diagramm-Modell-Element ist ein Modell-Element, welches im Project Browser nicht aufscheint, aber fast alle Eigenschaften eines „normalen“ Modell-Elements besitzt.
Diagramm-Element	Ein Diagramm-Element ist ein Modell-Element, das nur im Diagramm, jedoch nicht im Project Browser sichtbar ist und wie Modell-Elemente durch ein Diagram-Object dargestellt wird.
Diagramm-Objekt (engl.: Diagram-Object)	Das Diagram-Object ist die grafische Darstellung eines Modell-Elements in einem Diagramm. Ein Modell-Element wird in einem Diagramm durch maximal ein Diagram-Object dargestellt.
Dokument-Template	Ein anpassbares Template zum Generieren von Dokumenten im Enterprise Architect.
Enterprise Architect (EA)	Das Modellierungswerkzeug mit dem Namen „Enterprise Architect“ der Firma SparxSystems.
EA-Repositorium	Die Datenbank in der EA Modelle abgelegt sind. Wir unterscheiden zwischen datei-basierten EA-Repositorien und datenbank-basierten.
EA-Paket	Ein EA-Paket ist ein spezielles Modell-Element, um Modelle zu strukturieren.
EA-Tag	Ein EA-Tag ist ein Bereich, der im Dokument-Generierungs-Template verwendet wird und den Kontext eines Modell-Elements definiert.
Konnektor/Link	Beschreibt eine Beziehung zwischen Modell-Elementen - mit anderen Worten: ein Strich zwischen Modell-Elementen. Je nach verwendeter Modellierungssprache (und Diagrammart) haben die Konnektoren unterschiedliche Bedeutungen.
Metamodell	Ein Metamodell ist ein „Schema“ bzw. Vorlage, wie Modelle erstellt werden können. Modelle. Es definiert die einzelnen Modell-Elemente, deren Semantik und Struktur.
Modell	Das Modell ist die Sammlung aller Modell-Elemente und liegt im EA Project Browser.
(Root) Modell	Jedes EA-Repositorium hat mindestens ein Root-Model (Wurzel im Project Browser) und kann beliebig viele davon enthalten.
Modell-Element	Ein Modell-Element bezeichnet ein beliebiges Symbol einer Modellierungssprache. Mit anderen Worten: Alles, was kein Strich zwischen Modell-Elementen ist. Modell-Elemente werden i. d. R. im Project Browser angezeigt.
Modellierungsansatz	Die Art und Weise wie mit einer Modellierungssprache ein Modell im Enterprise Architect erstellt wird. Dazu gehören die Dimensionen: Modellierungssprache, Modellierungswerkzeug, Methode, Erfahrung. Ein konkretes Ergebnis eines Modellierungsansatzes ist das „Referenzmodell“.
Private Model	Ein EA-Repositorium, das von einem EA-Benutzer alleine verwendet wird, aber über Versionskontrolle Modelle(-teile) austauscht.
Project Browser	Der Project Browser ist die Ablagestruktur der Modell-Elemente und Diagramme. Er ist vergleichbar mit dem Windows-Explorer.
Projekt-Metamodell	Die allgemeinen Regeln wie ein Modell aufgebaut sein soll.
Referenzmodell	Das Referenzmodell ist Teil eines Modellierungsansatzes. Es enthält: die Modellstruktur, sowie Beispiele für die zu verwendende

	Modellierungssprache. Auf dem Referenzmodell basierend können Modellierungsrichtlinien abgeleitet werden, die anschließend als automatisch ausführbare Validierungsregeln und Automatismen zur Erleichterung des Modellierens umgesetzt werden können.
Shared Model	Ein EA-Repository, das von mehreren EA-Benutzern gleichzeitig verwendet und modifiziert wird.
Shared Repository	Ein EA-Repository, das projektspezifische Informationen, wie z. B. das Projekt-Glossar, mit anderen EA-Repositories teilt.
Strukturierte-Elemente (<i>Structured elements</i>)	Ein Strukturiertes-Element ist ein Modell-Element, das nur gemeinsam mit einem anderen Modell-Elemente (dem Container) existieren kann. Ein Port, ActionPin, etc. sind solche Modell-Elemente.
View-Paket	Das View-Paket ist ein spezielles EA-Paket. Es hat ein View Icon.
Quicklink	Schnellzeichenpfeil an der oberen rechten Ecke eines Modell-Elementes, um einen Konnektor zu einem anderen Modell-Element zu erstellen. Das Quicklink-Menü ist ein Vorschlag von möglichen Beziehungen.
«StereotypName»	Ein Stereotyp ist ein Begriff aus der Sprache UML, um einem UML Element eine stärkere Bedeutung (Semantik) zu verleihen.

Konventionen

Um den Titel des Buches gerecht zu werden und ein schnelles Nachschlagen und Auffinden der wichtigsten Informationen zu erleichtern, hier einige Konventionen.



Menüeinträge: Verweise auf

Menüeinträge werden immer mit eckigen Klammern notiert. Zum Beispiel: im Hauptmenü **[Configure > Model > Options > C# > Default Extension]**. Mehrere Optionale Auswahlmöglichkeiten werden in spitzen Klammern mit einem | getrennt geschrieben, z.B. **[Start > Workspace > Preferences > Links > <Routing | Pen Width>]**. Wird durch einen Menüpunkt ein Dialog geöffnet in dem weiter verwiesen wird, wird der Verweis in den Dialog mit einem "->" dargestellt. Zum Beispiel: **[Start > Workspace > Preferences -> Diagram > Behavior > Autosave Changes]**.

Symbole: Zum einfachen Finden bestimmter Informationen, werden diese Symbole als Marginalien verwendet.

 Hinweis	Merken	 Übung/Konfiguration	 Tipp
---	--------	---	--

Tastenkombination: Tastenkombinationen werden wie Menüeinträge in eckigen Klammern geschrieben. Zum Beispiel: **[Strg + Tab]**.

Eigennamen: Wird im Text auf ein Beispiel Bezug genommen oder spezielle Sprach-Elemente hervorgehoben, so wird der Name kursiv geschrieben. Beispiele sind der Typ *Integer* oder *int* bzw. *UseCase* oder *Person* als Name einer Klasse, sowie EA Fensternamen, z. B. *Project Browser* oder *Properties*-Dialog. Damit die kursive Darstellung nicht überstrapaziert wird und für ein angenehmer Lesefluss, wird der Begriff nicht immer kursiv geschrieben, sondern lediglich, wenn dessen Bedeutung in Kontext des Satzes hervorgehoben werden soll.

Textblöcke, welche spezielle Themen behandeln, werden hervorgehoben.

Text beschreibt ein spezielles Vorgehen oder eine Methode.

Name: Persönliche Meinung und/oder Erfahrung

Code (Script, SQL, etc.)

Modellierungssprachen spezifische Erklärung (UML, SysML, BPMN, etc.)

Inhaltsverzeichnis

1. WARUM MODELLIEREN UND WARUM MIT DEM EA?	17
1.1. WARUM VERWENDEN WIR EIN MODELLIERUNGSWERKZEUG?	18
1.2. EINSATZMÖGLICHKEITEN DES ENTERPRISE ARCHITECT	19
1.3. WARUM MODELLIEREN WIR ÜBERHAUPT?	21
1.3.1. VEREINFACHTE DARSTELLUNG KOMPLEXER SYSTEME	21
1.3.2. KOMMUNIKATIONSGRUNDLAGE	21
1.3.3. GENERIEREN VON DOKUMENTEN	21
1.3.4. MODELLE TRANSFORMIEREN	22
1.3.5. CODE GENERIEREN	22
1.3.6. VORHANDENEN CODE EINLESEN	23
1.3.7. VERIFIKATION DER MODELLE	23
1.3.8. SIMULATION VON MODELLEN	23
1.3.9. ÄNDERUNGEN AUTOMATISCH DURCHFÜHREN	24
1.3.10. TESTFÄLLE GENERIEREN	24
1.4. DIE HÜRDEN BEIM EINFÜHREN EINES MODELLIERUNGSANSATZES	24
1.4.1. WELCHE MODELLIERUNGSSPRACHE SOLL VERWENDET WERDEN?	25
1.4.2. DAS AUSGEWÄHLTE WERKZEUG	26
1.4.3. WIE SIEHT DAS „PROJEKT-METAMODELL“ DES ZU ERSTELLENDEN MODELLS AUS?	27
1.4.4. WIE KANN ICH DAS ERSTELLEN DES MODELLS, PASSEND ZUM PROJEKT-METAMODELL, ERLEICHTERN?	27
1.4.5. WIE ABSTRAKT SOLL MODELLIERT WERDEN?	29
1.4.6. WIE VIELE DETAILS SIND ERFORDERLICH?	29
1.4.7. WIE VOLLSTÄNDIG SOLL DAS MODELL SEIN?	29
1.5. EINEN MODELLIERUNGSANSATZ IM UNTERNEHMEN EINFÜHREN	30
1.5.1. MÖGLICHE SZENARIEN	30
1.5.2. EIN ITERATIVER EINFÜHRUNGSPROZESS	31
2. GRUNDLEGENDE KONFIGURATION	32
2.1. ENTERPRISE ARCHITECT EDITIONEN UND LIZENZEN	32
2.2. ERSTINSTALLATION	34
2.3. AUFBAU UND GRUNDLEGENDE STRUKTUR VON ENTERPRISE ARCHITECT	37
2.4. EINSTELLUNGEN, DIE DAS ARBEITEN ERLEICHTERN	39
3. ERSTE SCHRITTE	41
3.1. NEUES PROJEKT ANLEGEN	41
3.2. ARBEITEN MIT VORDEFINIERTEN TEMPLATES (DER MODEL WIZARD)	42
3.3. DIAGRAMME UND MODELLE ERSTELLEN	42
3.3.1. DIAGRAMM ERSTELLEN	42
3.3.2. DIAGRAMMINHALTE UND DIAGRAMME LÖSCHEN	43
3.3.3. MODELL-ELEMENT ERSTELLEN	44
3.3.4. BEZIEHUNGEN ERSTELLEN	45
3.4. DER UNTERSCHIED ZWISCHEN DEM MODELL UND DIAGRAMMEN	46
3.5. DIAGRAMME LAYOUTEN	47
3.5.1. MANUELLES LAYOUTEN	47
3.5.2. AUTOMATISCHES LAYOUTEN	48
3.6. DIE WICHTIGSTEN FUNKTIONEN UND FENSTER	48
3.6.1. DER PROJECT BROWSER	48

3.6.2.	DIE DIAGRAMM-TOOLBOX	49
3.6.3.	DER TRACE VIEW	50
3.6.4.	DER ELEMENT-BROWSER	51
3.6.5.	DER ELEMENT-PROPERTY DIALOG	52
3.6.6.	DAS PROPERTIES FENSTER	56
3.6.7.	DER KONNEKTOR-PROPERTY DIALOG	57
3.6.8.	DER DIAGRAMM-PROPERTY DIALOG	58
3.6.9.	DAS DIAGRAMM-FILTER FENSTER	59
3.6.10.	DER <i>INSERT RELATED ELEMENT</i> DIALOG	59
3.6.11.	DAS NOTIZ FENSTER	60
3.6.12.	DAS TAGGED VALUE FENSTER	61
3.6.13.	DAS RELATIONSHIPS FENSTER	62
3.6.14.	DIE RELATIONSHIP MATRIX	63
3.6.15.	DIE TABELLENANSICHT EINES PAKETES	66
3.6.16.	DER PARENT DIALOG	68
3.6.17.	FENSTER ANORDNEN UND ALS WORKSPACE LAYOUT SPEICHERN	68
3.6.18.	DATEIEN ALS MODELL-ELEMENTE EINFÜGEN	69
3.7.	DAS PORTAL-SYSTEM IN ENTERPRISE ARCHITECT	69
3.7.1.	DAS NAVIGATOR FENSTER	70
3.7.2.	DAS PORTALS FENSTER	70
3.8.	GRUNDLEGENDE MODELLIERUNGSKONZEPTE IM EA	71
3.8.1.	STEREOTYPEN UND TAGGED VALUES	71
3.8.2.	TYP – INSTANZ – BEZIEHUNGEN (CLASSIFIED ELEMENT)	73
3.8.1.	NAMENSRÄUME UND NAMENSKONVENTION	75
3.8.2.	DIE MULTIPLIZITÄT AN KONNEKTOREN	75
3.8.3.	HYPERLINKS	76
3.8.4.	DIE NOTIZEN	77
3.8.5.	AUTOMATISCHES VERSCHACHTELN IM PROJECT BROWSER	79
3.9.	DIE GRUNDLEGENDE DATENSTRUKTUR UND ARBEITSWEISE IM EA	79
3.9.1.	DAS DIAGRAMM (IST NICHT DAS MODELL)	80
3.9.2.	DAS MODELL-ELEMENT	83
3.9.3.	ZUSÄTZLICHE MODELL-ELEMENT EIGENSCHAFTEN (ADDITIONAL PROPERTIES)	83
3.9.4.	DAS CHILD ELEMENT (SUB-ELEMENT)	83
3.9.5.	DAS STRUKTURIERTE-ELEMENT (<i>STRUCTURED ELEMENT</i>)	84
3.9.6.	DAS DIAGRAMM-ELEMENT	87
3.9.7.	DER KONNEKTOR	88
3.9.8.	DAS PAKET	90
3.9.9.	XMI ALS GRUNDLEGENDES AUSTAUSCHFORMAT	91
3.9.10.	STEREOTYPE VERWALTEN	91
3.10.	DAS FARBENSPIEL IN ENTERPRISE ARCHITECT	93
3.10.1.	KONFIGURIEREN DER FARBEN VON ELEMENTEN	93
3.10.2.	KONFIGURIEREN DER FARBEN UND DES ROUTINGS VON VERBINDUNGEN	94
3.10.3.	ÜBERTRAGEN DES STYLES (PIPETTE, PINSEL, STYLE LIST)	95
3.10.4.	DIE DIAGRAMM-LEGENDE ZUM EINFÄRBN VON MODELL-ELEMENTEN UND KONNEKTOREN	95
3.11.	WEITERE EINFACHE KONFIGURATIONEN	97
3.11.1.	TASTENKOMBINATIONEN KONFIGURIEREN (SHORT KEY)	97
3.11.2.	VORHANDENE TOOLBARS ANPASSEN UND EIGENE ERSTELLEN	97
3.11.3.	DROPDOWN-LISTEN KONFIGURIEREN	97
4.	<u>DIE PROJEKTSTRUKTUR (DAS MODELL)</u>	98
4.1.	GRUNDLEGENDE GEDANKEN ZUM AUFBAU DER PROJECT BROWSER STRUKTUR	99

4.1.1.	DAS ROOT PACKAGE	100
4.1.2.	DAS VIEW PACKAGE	101
4.2.	VERLINKUNGSMUSTER FÜR DIE NAVIGATION IM DIAGRAMM	101
4.3.	VERFEINERUNG DURCH COMPOSITE-ELEMENTE UND ZWISCHENDIAGRAMM	102
4.4.	ÜBERLEGUNGEN ZUR TRACEABILITY	104
4.4.1.	TRACEN MITTELS TASTENKOMBINATIONEN	104
4.4.2.	TRACES MITTELS BEZIEHUNG	105
4.4.3.	TRACEN MITTELS <i>CUSTOME REFERENCE</i>	105
4.4.1.	TRACEN MITTELS TAGGED VALUES	106
4.4.1.	TRACEN MITTELS HYPERLINK	106
4.5.	BEISPIELE FÜR TRACE-KETTEN	107
4.5.1.	TRACE-KETTEN MITTELS REALIZATION BEZIEHUNG	107
5.	UML MIT ENTERPRISE ARCHITECT	109
5.1.	UML HINTERGRUND, ÜBERBLICK UND ZUSAMMENHÄNGE	109
5.1.1.	WOHER KOMMT UML?	110
5.1.2.	DAS DIAGRAMM UND DAS MODELL	110
5.1.3.	KERN-KONZEPT DER UML	110
5.1.4.	DIE UML-SPRACHFAMILIE	110
5.1.5.	ZUSAMMENHANG ZWISCHEN DEN EINZELNEN UML-MODELLEN	112
5.2.	HÄUFIG VERWENDETE BEZIEHUNGEN	113
5.2.1.	ASSOCIATION	113
5.2.2.	TEIL-GANZES-BEZIEHUNG (AGGREGATION UND COMPOSITION)	117
5.2.3.	DEPENDENCY	118
5.2.4.	ABSTRACTION	119
5.2.5.	DIE USAGE BEZIEHUNG	120
5.2.6.	DER INFORMATIONFLOW	120
5.2.7.	DIE GENERALISIERUNG (<i>GENERALISATION</i>)	121
5.2.8.	REALIZATION	123
5.3.	USE CASES MODELLIERUNG	123
5.3.1.	DIE META-INFORMATIONEN EINES USE CASES	124
5.3.2.	DAS USE CASE SZENARIO	124
5.3.3.	DER AUSLÖSER (TRIGGER) EINES USE CASES	125
5.3.4.	DIE INCLUDE-BEZIEHUNG ZWISCHEN USE CASES	125
5.3.5.	DIE EXTEND-BEZIEHUNG ZWISCHEN USE CASES	126
5.3.6.	DER ERWEITERUNGSPUNKT (EXTENSION-POINT) EINES USE CASES	126
5.3.7.	DIE VOR- UND NACHBEDINGUNGEN EINES USE CASES	127
5.4.	AKTIVITÄTEN-MODELLIEREN (PROZESSE, WORKFLOWS UND ALGORITHMEN)	127
5.4.1.	ANWENDUNGSBEREICHE DES AKTIVITÄTSDIAGRAMMS	128
5.4.2.	DAS TOKEN-KONZEPT IM AKTIVITÄTSMODELL	129
5.4.3.	GRUNDLEGENDE MODELL-ELEMENTE IM AKTIVITÄTSDIAGRAMM	129
5.4.1.	DATENFLÜSSE MODELLIEREN	133
5.4.2.	PROZESSSCHRITTE (AKTIONEN) VERFEINERN	137
5.4.3.	NEBENLÄUFIGKEIT MODELLIEREN	139
5.4.4.	DEN KONTEXT EINER AKTION MODELLIEREN	140
5.4.5.	ASYNCHRONE KOMMUNIKATION	142
5.5.	KLASSEN-MODELLIERUNG (STRUKTUREN UND DATEN)	146
5.5.1.	ANWENDUNGSBEREICHE DES KLASSENDIAGRAMMS	146
5.5.1.	DIE WICHTIGSTEN BEZIEHUNGEN DER KLASSE	147
5.5.2.	DIE KLASSE	147
5.5.3.	KLASSEN IN KLASSEN (NESTED CLASSES)	148

5.5.4.	DIE FEATURES DER KLASSE (ATTRIBUTE UND OPERATIONEN)	149
5.5.5.	REDEFINIEREN VON FEATURES (ATTRIBUTEN UND OPERATIONEN)	152
5.5.6.	SCHNITTSTELLEN (INTERFACE)	153
5.5.7.	DIE REALISIERUNGSBEZIEHUNG (REALIZATION)	153
5.5.8.	ERSTELLEN EINES DOMÄNENMODELLS	153
5.5.9.	DISKUSSION VON MODELLIERUNGS-VARIANTEN IM KLASSENDIAGRAMM	154
5.6.	OBJEKTE MODELLIEREN (<i>INSTANCE SPECIFICATION</i>)	156
5.6.1.	KONKRETE WERTE FÜR INSTANZEN SETZEN (SET RUN STATE)	156
5.6.2.	UNTERSCHIED INSTANZ, PROPERTY UND PORT, ETC.	157
5.7.	KOMPONENTEN-MODELLIERUNG	157
5.7.1.	EINFACHE KOMPONENTE (BASIC COMPONENT)	158
5.7.2.	VERBINDUNGEN VON KOMPONENTEN	163
5.7.3.	PACKAGING COMPONENT	163
5.8.	KOMPOSITIONS-STRUKTUR DIAGRAMM (COMPOSITE STRUCTURE DIAGRAM)	164
5.8.1.	VERWENDUNG VON PARTS (<i>PROPERTY</i> ELEMENTE)	165
5.8.2.	PROPERTY VS. INSTANZ	166
5.8.3.	KOMPONENTEN IN KOMPONENTEN SCHACHTELN	166
5.8.4.	VERBINDUNG VON PORTS IN KOMPONENTEN	167
5.9.	DEPLOYMENT DIAGRAMM	168
5.9.1.	DAS ARTEFAKT (<i>ARTIFACT</i>)	169
5.9.1.	DIE PHYSISCHE WELT MODELLIEREN (NODE, DEVICE, EXECUTION ENVIRONMENT)	169
5.9.2.	DER LINK ZWISCHEN LOGISCHER UND PHYSISCHER WELT	170
5.10.	INTERAKTIONEN MODELLIEREN	170
5.10.1.	SEQUENZ-MODELLIERUNG	170
5.11.	ZUSTANDSAUTOMATEN (STATE MACHINE) MODELLIEREN	177
5.11.1.	EINFACHE ZUSTANDSAUTOMATEN	177
6.	<u>SYSML MIT ENTERPRISE ARCHITECT</u>	185
6.1.	SYSML HINTERGRUND, ÜBERBLICK UND ZUSAMMENHANG MIT UML	185
6.2.	WICHTIGE SYSML BEZIEHUNGEN	185
6.2.1.	ALLOCATION BEZIEHUNG	185
6.3.	SYSML SPEZIFISCHE KONFIGURATIONEN IM ENTERPRISE ARCHITECT	186
6.3.1.	SYSML ALS PRIMÄRE TECHNOLOGIE KONFIGURIEREN	186
6.3.2.	SYSML VERSIONEN MIGRIEREN	186
6.4.	DAS BLOCK DEFINITION DIAGRAMM (BDD)	186
6.4.1.	SYSML PORTS	187
6.4.2.	DAS SYSML FLOWPROPERTY	188
6.4.3.	PORT BEISPIELE	188
6.4.4.	AGGREGATIONS- UND KOMPOSITIONS-BEZIEHUNGEN IN SYSML	189
6.5.	DAS INTERNE BLOCK DEFINITION DIAGRAMM (IBD)	190
6.5.1.	BEISPIEL 1: EINFACHE VERDRAHTUNG IM IBD	191
6.5.2.	BEISPIEL 2: VERDRAHTUNG MIT PORTS UND PROPERTIES	192
6.6.	DAS SYSML ANFORDERUNGSMODELL	192
6.6.1.	DAS ANFORDERUNGS-ELEMENT (SYSML REQUIREMENT)	193
6.6.2.	DAS TEST-FALL ELEMENT (TEST CASE)	193
6.6.3.	SYSML ANFORDERUNGS-BEZIEHUNGEN	193
6.7.	DAS PARAMETRIC-MODELL	194
6.7.1.	DER CONSTRAINT-BLOCK	195
6.7.2.	DAS CONSTRAINT-MODEL	195
6.7.3.	SIMULATION VON CONSTRAINT-MODELS	196
6.8.	SYSML ERWEITERUNGEN DES UML-AKTIVITÄTS-MODELL	196

7. REQUIREMENT-ENGINEERING MIT ENTERPRISE ARCHITECT	196
7.1. EA REQUIREMENTS	197
7.2. TRACEABILITY VON ANFORDERUNGEN	197
7.3. DER SPECIFICATION MANAGER	197
7.3.1. KONFIGURATION DES SPECIFICATION MANAGERS	198
7.4. REQUIREMENT MANAGEMENT TOOL INTEGRATION	199
8. PROJEKT-MANAGEMENT UND TEAM-KOLLABORATION MIT ENTERPRISE ARCHITECT	199
8.1. AUFWANDSABSCHÄTZUNG (DIE USE CASE POINT METHODE)	199
8.2. DAS PROJEKT-GLOSSAR	200
8.2.1. GLOSSAREINTRÄGE ERSTELLEN UND VERWALTEN	200
8.2.2. DAS GLOSSAR ZWISCHEN EA-REPOSITORIEN AUSTAUSCHEN	201
8.2.3. EIN GLOSSAR FÜR MEHRERE EA-REPOSITORIEN	201
8.2.4. DAS GLOSSAR PROGRAMMATISCH ANSPRECHEN	201
8.3. DISKUSSIONSMÖGLICHKEITEN IM TEAM	202
8.3.1. DER TEAM REVIEW	202
8.3.2. DISKUSSION AM MODELL-ELEMENT	203
8.3.3. MODEL CHAT	204
8.3.4. MODEL MAIL	204
8.3.5. DER PROJEKTKALENDER	204
8.3.6. DER MODEL VIEW DIALOG	204
8.4. ARBEITEN MIT RESSOURCEN	206
8.4.1. RESSOURCEN KONFIGURIEREN	206
8.4.2. RESSOURCEN ZUWEISEN	206
8.4.3. DER GANTT VIEW	207
9. ARBEITEN MIT DEM CODE	207
9.1. CODE-GENERIERUNG (FORWARD ENGINEERING)	207
9.1.1. GRUNDLEGENDE EINSTELLUNGEN BEIM CODE-GENERIEREN	208
9.1.1. MAPPING ZWISCHEN UML KLASSEN UND CODE-FILE	210
9.1.2. GENERIEREN VON CODE FÜR EINE SELEKTIERTE KLASSE	211
9.1.3. GENERIEREN VON CODE FÜR EIN GESAMTES PAKET	213
9.1.4. GENERIEREN VON ATTRIBUTEN UND DEREN EIGENSCHAFTEN	214
9.1.5. GENERIEREN VON BEZIEHUNGEN UND DEREN EIGENSCHAFTEN	215
9.1.6. GENERIEREN VON OPERATIONEN UND DEREN EIGENSCHAFTEN	219
9.1.7. GENERIEREN VON NAMENSRÄUMEN	220
9.1.8. GENERIEREN VON #INCLUDE, USING, IMPORT BEI DEPENDENCIES	221
9.2. GENERIEREN VON CODE AUS VERHALTENSMODELLEN	222
9.2.1. GENERIEREN VON CODE AUS AKTIVITÄTS-DIAGRAMMEN (ACTIVITY DIAGRAM)	222
9.2.2. GENERIEREN VON CODE AUS INTERAKTIONSDIAGRAMMEN (SEQUENCE DIAGRAM)	223
9.2.3. GENERIEREN VON CODE AUS ZUSTANDSAUTOMATEN (STATE MACHINE)	224
9.3. EINLESEN VON CODE (REVERSE ENGINEERING)	225
9.3.1. EINLESEN EINES EINZELNEN SOURCE-FILES	225
9.3.1. EINLESEN EINES GESAMTEN VERZEICHNIS MIT SOURCE-FILES	225
9.3.2. EINLESEN VON BINARIES	226
9.3.1. UMGANG MIT KOMMENTAREN BEIM ROUNDTRIP-ENGINEERING	226
9.4. CODE-GENERIERUNG IM TEAM	229
9.5. DAS CODEGENERIERUNGSFRAMEWORK	230

9.5.2.	ANPASSUNGSBEISPIEL FÜR DAS CODE GENERATION FRAMEWORK	235
10.	MODELLTRANSFORMATIONEN	238
10.1.	EINGebaute Modelltransformationen	239
10.2.	Das Modelltransformations-Framework	241
10.2.1.	Das Format des Intermediate-File	241
10.2.2.	Der Templatemechanismus der Modelltransformation	244
11.	MODELLSIMULATION	247
11.1.	Die wichtigsten Fenster zum Simulieren	247
11.1.1.	Das Simulator Fenster	247
11.1.2.	Das Triggers Fenster	247
11.2.	Simulationsarten	248
12.	MODELLIERUNG VON DATENBANKEN	248
12.1.	Arbeiten mit den EA Legacy Datenbank Werkzeugen	248
12.1.1.	Logisches Datenbank-Modell im EA erstellen	248
12.1.2.	Logisches Modell in ein physisches Datenbank-Modell transformieren	251
12.1.3.	Physisches Datenbank-Modell im EA erstellen	252
12.1.4.	DDL Statements aus mit dem EA generieren	254
12.1.5.	Ein physisches Datenbank-Modell über eine ODBC Datenquelle erstellen	254
12.2.	Arbeiten mit dem Database Builder	255
12.2.1.	Verbindung zu einem DBMS aufbauen	256
12.2.2.	Tabellen mit dem Database Builder erstellen	256
12.2.3.	DDL Skripte generieren und ausführen	256
12.2.4.	Differenz zwischen Modell und Datenbank berechnen	257
13.	MODELLIERUNG VON XML SCHEMATA	258
13.1.	XML Schema Modell	259
13.1.1.	Vorhandene XSD Files importieren	259
13.1.2.	Ein eigenes XML Schema Modell erstellen	260
13.2.	Der Schema Composer	260
13.2.1.	Schema Profile anlegen	261
13.2.2.	Eine Modell-Variante erstellen	261
13.2.3.	Schema generieren	262
13.2.4.	Modell transformieren	263
14.	MODELLVALIDIERUNG	264
14.1.	Eingebaute Validierungsregeln	265
14.1.1.	Object Constraint Language (OCL)	265
14.1.2.	Regel-Kategorien automatisch aktivieren/deaktivieren (Validation Configuration)	266
14.2.	Validierung durch Suchen	266
14.3.	Validierung mittels Modell-Simulation	267
14.4.	Eigene Validierungsregeln schreiben	267

14.4.1.	VALIDIERUNGS-REGELN ALS SKRIPT SCHREIBEN	267
14.4.2.	VALIDIERUNGS-REGEL ALS ADD-IN SCHREIBEN	268
15.	DOKUMENTE GENERIEREN	269
15.1.	DER DOKUMENT-TEMPLATE MECHANISMUS	271
15.1.1.	WEITERE ARTEN VON TEMPLATES	273
15.1.2.	FORMATVORLAGEN ANPASSEN (EDIT STYLE...)	273
15.1.3.	ÜBERSCHRIFTEN-NUMMERIERUNG MITTELS LISTEN KONFIGURIEREN	274
16.	VERSIONIEREN VON MODELLEN	276
16.1.	ARBEITEN MIT BASELINES (INTERNE VERSIONIERUNG)	277
16.2.	ANBINDUNG AN EXTERNE VERSIONSKONTROLLSYSTEME	278
16.3.	BRANCHING VON MODELLEN	279
16.3.1.	HORIZONTALES BRANCHING	279
16.3.2.	VERTIKALES BRANCHING	281
16.4.	AUDITING VON MODELLEN	284
16.4.1.	SPEICHERBEDARF	284
16.4.2.	AUDIT-LOGS VERWALTEN	285
16.4.3.	IM AUDIT SUCHEN	285
17.	TEAMARBEIT (MANAGEN VON EA-REPOSITORIEN)	285
17.1.	SHARED MODEL	286
17.2.	PRIVATE MODEL	287
17.2.1.	VERTEILEN VON PRIVATEN-MODELLEN	289
17.3.	KOMBINATION AUS SHARED UND PRIVATE MODEL	290
17.3.1.	SHARED MODEL MIT EXTERNER VERSIONSKONTROLLE	290
17.3.2.	KOMBINATION MEHRERER SHARED MODELS MIT EXTERNER VERSIONSKONTROLLE	290
17.3.3.	SHARED MODEL MIT XMI EXPORT/IMPORT UND PROJECT TRANSFER	291
17.4.	EA CLOUD UND PRO CLOUD	292
17.4.1.	EA CLOUD SERVICE KONFIGURATION	292
17.4.2.	AKTIVIEREN DES PRO CLOUD SERVERS	293
17.4.3.	VERBINDUNG MIT DER EA CLOUD AUFBAUEN	294
17.5.	WEB EA	294
17.5.1.	WEB EA EINRICHTEN	294
17.6.	MODELL REPLIKAT	295
17.6.1.	ERSTELLEN EINES NEUEN MASTER-PROJEKTES	295
17.6.2.	ERSTELLEN EINES NEUEN REPLIKATS	295
17.6.3.	SYNCHRONISIEREN VON REPLIKATEN MIT DEM MASTER-PROJEKT	295
17.6.4.	HINWEISE ZUM PRAKTISCHEN EINSATZ VON REPLIKATEN	296
17.7.	MODELLE PER XMI AUSTAUSCHEN	296
17.7.1.	XMI-EXPORT	297
17.7.2.	XMI-IMPORT	298
17.7.3.	AUTOMATISCHES ZUSAMMENFÜHREN (MERGEN) BEIM XMI-IMPORT	299
17.8.	CSV EXPORT / IMPORT	304
17.9.	PAKETÜBERGREIFENDE REFERENZEN	305
17.9.1.	BEHANDLUNG VON CROSS PACKAGE REFERENCES	307
17.9.2.	ARBEITSWEISE MIT AKTIVIERTER CROSS PACKAGE REFERENCE VERFOLGUNG	308
17.9.3.	ANWENDUNGSMÖGLICHKEITEN	309

17.10. PROJEKT TRANSFER	310
17.11. TEILMODELLE IN UNTERSCHIEDLICHEN REPOSITORYN VERWALTEN	310
17.11.1. MODELLZWEIGE UNTER VERSIONSKONTROLLE STELLEN	311
17.11.2. IMPORT EINES MODELLZWEIGES IN EIN ANDERES EA-REPOSITORY	311
17.11.3. ARBEITSSZENARIEN	312
17.12. EA-SECURITY	312
17.12.1. USER- UND GRUPPEN-RECHTE KONFIGURIEREN	312
17.12.2. WINDOWS AUTHENTIFIZIERUNG VERWENDEN	313
17.12.3. REQUIRE USER LOCK TO EDIT MODUS AKTIVIEREN	314
17.12.4. DAS MODELL SPERREN	314
17.13. UMGANG MIT PROJEKTSPEZIFISCHEN KONFIGURATIONEN	315
18. ENTERPRISE ARCHITECT ERWEITERN UND ANPASSEN	315
18.1. KONFIGURATIONEN	315
18.1.1. WERKZEUG SPEZIFISCHE KONFIGURATIONEN	316
18.1.2. PROJECT SPEZIFISCHE KONFIGURATIONEN	318
18.2. MDG TECHNOLOGIES	321
18.2.1. WAS KANN ALLES IN EINE MDG-TECHNOLOGIE INTEGRIERT WERDEN?	321
18.2.2. EIGENE MDG TECHNOLOGIEN ERSTELLEN	322
18.2.3. MDG-TECHNOLOGIEN VERTEILEN	322
18.3. (UML-) PROFILE	323
18.3.1. WAS IST EIN UML-PROFIL? (OMG METAMODEL LAYER)	323
18.3.2. MDG-TECHNOLOGY PROJEKT ERSTELLEN	324
18.3.3. SPRACHPROFILE (UML-PROFILE)	324
18.3.4. QUICKLINK KONFIGURATION ERSTELLEN	325
18.3.5. TOOLBOX-PROFILE	325
18.3.6. DIAGRAMM-PROFILE	327
18.3.7. MDG-TECHNOLOGY ALS AKTIVE TECHNOLOGIE KONFIGURIEREN	328
18.4. DESIGN PATTERN ERSTELLEN	328
18.4.1. EIGENES PATTERN SPEICHERN	328
18.4.2. VERWENDEN VON PATTERN	328
18.5. TEMPLATE PAKETE (KLEINES PROFIL)	329
18.6. MODELLSUCHE	329
18.6.1. VORHANDENE SUCHEN VERWENDEN	330
18.6.2. QUERY BUILDER SUCHEN ERSTELLEN UND EDITIEREN	330
18.6.3. DAS EA-DATENBANKSCHEMA	332
18.6.4. SQL SUCHEN ERSTELLEN	334
18.6.5. ADD-IN SUCHEN ERSTELLEN	335
18.6.6. BEISPIEL SUCHEN	336
18.7. DAS EA-OBJEKTMODELL	336
18.7.1. DIE REPOSITORY-KLASSE	337
18.7.1. DIE PAKET-KLASSE	338
18.7.2. DIE ELEMENT-KLASSE	339
18.7.3. DIE EA.COLLECTION	339
18.7.4. DIE CONNECTOR-KLASSE	340
18.7.5. DIE CONNECTOREND-KLASSE	341
18.7.6. DIE DIAGRAMOBJECT-KLASSE	341
18.7.7. DIE DIAGRAMLINK-KLASSE	341
18.7.8. DAS PROJECT INTERFACE	342
18.8. SKRIPTEN	342
18.8.1. NORMALES SKRIPT (NORMAL GROUP)	342

18.8.2.	DIAGRAMM SKRIPT (DIAGRAM GROUP)	343
18.8.3.	PROJECT BROWSER SKRIPT (PROJECT BROWSER GROUP)	343
18.8.4.	WORKFLOW SKRIPT (WORKFLOW GROUP)	343
18.8.5.	SUCH SKRIPT (SEARCH GROUP)	343
18.8.6.	FIND IN PROJECT SKRIPT (MODEL SEARCH GROUP)	344
18.8.7.	DIAGRAM FILTER SKRIPT (DIAGRAM FILTER GROUP)	344
18.8.8.	LINK SKRIPT	344
18.8.9.	PACKAGE SKRIPT	344
18.8.10.	ELEMENT SKRIPT	344
18.9.	HYBRID SCRIPTS	344
18.10.	EXTERNE EA AUTOMATISIERUNG	345
18.11.	EA ADD-IN ENTWICKLUNG	346
18.11.1.	WAS BIETET MIR EIN ADD-IN	346
18.11.2.	NOTWENDIGE SCHRITTE ZUM ERSTELLEN EINES ADD-IN	347
18.11.3.	ADD-IN DEBUGGEN	347
18.11.4.	ADD-IN MODELL-SUCHEN SCHREIBEN	347
19.	INDEX	348
20.	LITERATUR	352
