Kompendium zu Enterprise Architect von Sparx Systems

Foundational – Advanced – Professional - Expert

Überarbeitete Auflage für die EA Version 15

SparxSystemes Software GmbH - Central Europe

Wir freuen uns auch über Ihre Verbesserungsvorschläge zu diesem Buch. Bitte nutzen Sie dazu die angegebenen E-Mail-Adressen.

Wien, im November 2019

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

Elektronische Buchsuche: booksearch.sparxsystems.eu
Blog: blog.sparxsystems.de
Europäische Webseite: www.sparxsystems.de (Deutsch) , www.sparxsystems.eu (Englisch)

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 Anpassung und Erweiterung von Enterprise Architect (EA).

Mail: *Horst.Kargl@sparxsystems.eu*

Über diese Auflage

Diese Auflage wurde überarbeitet und auf die EA Version 15 angepasst. Die meisten beschriebenen Funktionalitäten können auch mit älteren EA-Versionen verwendet werden. Eventuell befinden sich die Menüeinträge an einer anderen Position oder haben einen anderen Namen. In unserem Blog (blog.sparxsystems.de) finden Sie Artikel mit weiterführenden Informationen zu Übersetzungstabellen zwischen den unterschiedlichen EA Versionen.

In diesem Buch wurde zwischen Konfigurationen und Beispielen unterschieden. Dafür wurde ein neues Icon für die Konfiguration eingeführt. Siehe Kapitel Konventionen auf Seite vii.

Die einzelnen Übungs-Beispiele sind nun durchnummeriert, um sich einfacher über die enthaltenen Beispiele austauschen zu können.

Auch der Index des Buches wurde vollständig überarbeitet, um ein einfaches Nachschlagen und Auffinden der wichtigsten Themen und Begriffe zu ermöglichen. Neben dem Index im Buch gibt es auch eine elektronische Online-Suche, welche eine Volltextsuche im Buch ermöglicht. Die Buch-Suche finden Sie unter: booksearch.sparxsystems.eu

Neben diesen Erweiterungen haben wir die aktuelle Struktur des Buches beibehalten. EA spezifische Themen, Modellierungssprachen und Methoden werden gesondert ausgezeichnet. Um dem 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 sowohl ein serielles Lesen aller Kapitel, als auch ein gezieltes Lesen einzelner Kapitel möglich ist. Referenzen zu anderen Kapiteln 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 angereichert.

Der Buchtitel heißt Kompendium zu Enterprise Architect von Sparx Systems, schaut aber auch über den Tellerrand des Enterprise Architect hinaus und betrachtet Ansätze, Methoden und Erweiterungsmöglichkeiten, welche der EA nicht oder nicht in der Form bietet, die aber mit dem EA umsetzbar sind. So lernen Sie, wie Sie den EA als Ihre Modellierungs-Plattform einsetzen 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.

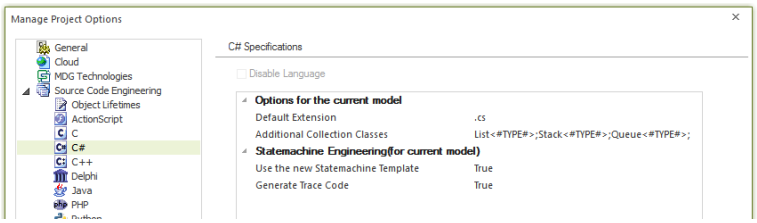
Glossar

Begriff	Erklärung
(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, das von einem Composite-Element referenziert wird.
Composite-Element	Ein Modell-Element, das einen Hyperlink zu einem Composite Diagram enthält.
Diagramm	Das Diagramm dient als Zeichenfläche, in der Modell-Elemente und Konnektoren visualisiert werden. Das Modell liegt im Browser.
Diagramm-Modell-Element	Das Diagramm-Modell-Element ist ein Modell-Element, welches im 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 Browser sichtbar ist und wie Modell-Elemente durch ein Diagramm-Objekt dargestellt wird.
Diagramm-Objekt (engl.: Diagram-Object)	Das Diagramm-Objekt ist die grafische Darstellung eines Modell-Elements in einem Diagramm. Ein Modell-Element wird in einem Diagramm durch maximal ein Diagramm-Objekt 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 Sparx Systems.
EA-Repositorium	Die Datenbank, in der EA Modelle abgelegt sind. Wir unterscheiden zwischen datei-basierten und datenbank-basierten EA Repositorien.
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. eine Vorlage, wie Modelle erstellt werden können. Es definiert die einzelnen Modell-Elemente, deren Semantik und Struktur.
Modell	Das Modell ist die Sammlung aller Modell-Elemente und liegt im EA Browser.

Begriff	Erklärung
(Root) Modell	Jedes EA-Repositorium hat mindestens ein Root-Model (Wurzel im 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 in der Regel im 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 die Versionskontrolle Modelle(-teile) austauscht.
Browser	Der 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-Repositorium, das von mehreren EA-Benutzern gleichzeitig verwendet und modifiziert wird.
Shared Repository	Ein EA-Repositorium, das projektspezifische Informationen, wie z. B. das Projekt-Glossar mit anderen EA-Repositorien teilt.
Strukturierte-Elemente (Structured elements)	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-Elements, um einen Konnektor zu einem anderen Modell-Element zu erstellen. Das Quicklink-Menü ist ein Vorschlag möglicher Beziehungen.
«StereotypName»	Ein Stereotyp ist ein Begriff aus der Sprache UML, um einem UML Element eine stärkere Bedeutung (Semantik) zu verleihen.

Konventionen

Um dem 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 mit spitzen Klammern mit einem | getrennt geschrieben, z. B. *[New Child Diagram ➢ <Activity | State Machine | Interaction, etc.>]*. Wird durch einen Menüpunkt ein Dialog geöffnet in dem weiter verwiesen wird, wird der Verweis in dem Dialog mit einem "→" dargestellt. Zum Beispiel: *[Start ➢ Desktop ➢ Workspaces → Save New Layout...]*.

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

Hinweis  | Merken  | Konfiguration  | Übung  | Tipp 

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

Namen von Modellierungssprach-Elementen: Werden im Text spezielle Modellelierungssprach-Elemente erwähnt, werden diese wie folgt dargestellt: Class, UseCase Actor. Primitive Datentypen werden ebenso wie Modellierungssprach-Elemente dargestellt: Integer, int, bool.

Namen die im Modell verwendet werden: Wird in einem Text ein Beispiel eines Modells beschrieben, werden die Namen der erstellten Modell-Elemente wie folgt geschrieben: *„Einzugebender Text“*. Bezieht man sich auf zwei Klassen mit dem Namen Person und Adresse, werden diese wie folgt dargestellt: *„Person“*, *„Adresse“*.

Einzugebender Text an der Benutzeroberfläche: Texte, die in der Benutzeroberfläche eingegeben werden, wie z. B. bei der Modell-Suche oder einem Eigenschaften-Dialog eines Modell-Elements, werden folgendermaßen hervorgehoben: *„Eingabe-Text im User Interface“*.

Namen von User Interface Fenster & EA Features und Eigenschaften: Wird Bezug auf eine bestimmte Funktion oder Fenster der Benutzeroberfläche genommen, wird der Name wie folgt dargestellt: *Name der Benutzeroberfläche*. Dies ist z. B. der *Browser*, der *Traceability* View oder die Eigenschaft *Status* eines Modell-Elements.

Textblöcke, die spezielle Themen behandeln, werden hervorgehoben.

Persönlicher Kommentar

Name: Persönliche Meinung und/oder Erfahrung

Semantik der Modellierungssprache

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

Code

```
Code (Script, SQL, etc.)
```

Überlegungen zur Modellierungsmethode

Text beschreibt ein spezielles Vorgehen oder eine Methode.

Inhaltsverzeichnis

Autor Information iii

Glossar v

Konventionen vii

1 Warum Modellieren und warum mit dem EA? 1
1.1 Warum verwenden wir ein Modellierungswerkzeug? . . . 2
1.2 Einsatzmöglichkeiten des Enterprise Architect . . . 3
1.3 Warum modellieren wir überhaupt? . . . 5
1.3.1 Vereinfachte Darstellung komplexer Systeme . . . 5
1.3.2 Kommunikationsgrundlage . . . 6
1.3.3 Generierung von Dokumenten . . . 6
1.3.4 Modelle transformieren . . . 6
1.3.5 Code generieren . . . 7
1.3.6 Vorhandenen Code einlesen . . . 7
1.3.7 Verifikation der Modelle . . . 8
1.3.8 Simulation von Modellen . . . 8
1.3.9 Änderungen automatisch durchführen . . . 8
1.3.10 Testfälle generieren . . . 8
1.4 Die Hürden bei der Einführung eines Modellierungsansatzes . . . 9
1.4.1 Welche Modellierungssprache soll verwendet werden? . . . 10
1.4.2 Das ausgewählte Werkzeug . . . 11
1.4.3 Wie sieht das „Projekt-Metamodell“ des zu erstellenden Modells aus? . . . 11
1.4.4 Wie erleichtere ich die Erstellung des Modells, passend zum Projekt-Metamodell? . . . 12
1.4.5 Wie abstrakt soll modelliert werden? . . . 13
1.4.6 Wie viele Details sind erforderlich? . . . 14
1.4.7 Wie vollständig soll das Modell sein? . . . 14
1.5 Einen Modellierungsansatz im Unternehmen einführen . . . 14
1.5.1 Mögliche Szenarien . . . 15
1.5.2 Ein Iterativer Einführungsprozess . . . 16

2 Grundlegende Konfiguration 19
2.1 Enterprise Architect: Editionen und Lizenzen . . . 19
2.2 Erstinstallation . . . 20
2.2.1 Floating License „mitnehmen“ . . . 21
2.3 Aufbau und grundlegende Struktur von Enterprise Architect . . . 22
2.3.1 Visual Style . . . 24
2.3.2 Ribbon Menu . . . 24
2.3.3 Perspective . . . 24
2.3.4 Fenster (Windows) . . . 25
2.3.5 Fenster Anordnung (Workspace Layout) . . . 25
2.3.6 Funktions-Knöpfe (Toolbar) . . . 25
2.3.7 MDG Technology . . . 26
2.3.8 Tastenkürzel (Command Short-Cuts) . . . 26
2.4 Einstellungen, die das Arbeiten erleichtern . . . 27

3 Erste Schritte **29**
3.1 Neues Projekt anlegen . . . 29
3.2 Arbeiten mit vordefinierten Templates (Model Pattern) . . . 30
3.3 Diagramme und Modelle erstellen . . . 30
3.3.1 Paket erstellen . . . 30
3.3.2 Diagramm erstellen . . . 31
3.3.3 Diagramminhalte und Diagramme löschen . . . 31
3.3.4 Modell-Element erstellen . . . 33
3.3.5 Beziehungen erstellen . . . 34
3.4 Der Unterschied zwischen Modell und Diagrammen . . . 35
3.5 Diagramme Layouten . . . 35
3.5.1 Manuelles Layouten . . . 36
3.5.2 Automatisches Layouten . . . 36
3.6 Die wichtigsten Funktionen und Fenster . . . 37
3.6.1 Der Project Browser . . . 37
3.6.2 Die Diagramm-Toolbox . . . 38
3.6.3 Der Traceability View . . . 38
3.6.4 Der Element-Browser . . . 39
3.6.5 Der modale Element-Property Dialog . . . 41
3.6.6 Das Dockable Property Fenster . . . 45
3.6.7 Der Konnektor-Property Dialog . . . 46
3.6.8 Der Diagramm-Property Dialog . . . 48
3.6.9 Das Diagramm-Filters Fenster . . . 49
3.6.10 Der Insert Related Elements Dialog . . . 49
3.6.11 Das Notiz Fenster . . . 50
3.6.12 Das Dynamic Document Fenster . . . 51
3.6.13 Tagged Values (Tags) im Property Fenster . . . 52
3.6.14 Das Relationships Fenster . . . 53
3.6.15 Die Relationship Matrix . . . 53
3.6.16 Relationship Matrix als Projekt Ressource konfigurieren . . . 54
3.6.17 Matrix Overlay . . . 55
3.6.18 Matrix Konfiguration als eigenes Modell-Element . . . 56
3.6.19 Aufruf der Relationship Matrix aus dem Diagramm . . . 56
3.6.20 Die Tabellenansicht eines Pakets . . . 56
3.6.21 Die Gruppierungs-Funktion der Tabellenansicht . . . 58
3.6.22 Der Parent Dialog . . . 58
3.6.23 Fenster anordnen und als Workspace Layout speichern . . . 59
3.6.24 Dateien als Modell-Elemente einfügen . . . 59
3.6.25 Das Navigator Fenster . . . 60
3.6.26 Das Portals Menü . . . 61
3.7 Das Ribbon Menü . . . 62
3.7.1 Eigene Ribbon Konfigurationen erstellen . . . 62
3.7.2 Ein Ribbon Set wählen . . . 62
3.7.3 Alle Ribbons aktivieren . . . 62
3.8 EA Perspective . . . 62
3.8.1 Eigene Perspektiven erstellen . . . 63
3.8.2 Alle Perspektiven aktivieren . . . 63
3.9 Grundlegende Modellierungskonzepte im EA . . . 63
3.9.1 Stereotypen und Tagged Values . . . 63
3.9.2 Typ – Instanz – Beziehungen (Classified Element) . . . 65
3.9.3 Namensräume und Namenskonventionen . . . 68
3.9.4 Ausblenden und Anzeigen des vollständigen Namensraumes . . . 68
3.9.5 Die Multiplizität an Konnektoren . . . 69
3.9.6 Hyperlinks . . . 69
3.9.7 Die Notizen . . . 71
3.9.8 Automatisches Verschachteln im Project Browser . . . 73
3.10 Die grundlegende Datenstruktur und Arbeitsweise im EA . . . 73
3.10.1 Das Diagramm (ist nicht das Modell) . . . 74
3.10.2 Das Modell-Element . . . 77

3.10.3 Zusätzliche Modell-Element Eigenschaften (additional properties) . . . 77
3.10.4 Das Child Element (Sub-Element) . . . 78
3.10.5 Das Strukturierte-Element (Structured Element) . . . 78
3.10.6 Das Diagramm-Element . . . 81
3.10.7 Der Konnektor . . . 83
3.10.8 Das Paket . . . 85
3.10.9 Pakete umstrukturieren . . . 86
3.10.10 Versionierung von Paketen . . . 86
3.10.11 XMI als grundlegendes Austauschformat . . . 86
3.10.12 Stereotypen verwalten . . . 86
3.10.13 Ad hoc Stereotyp anlegen . . . 87
3.10.14 Stereotyp per UML-Profil . . . 87
3.11 Das Farbenspiel in Enterprise Architect . . . 88
3.11.1 Konfiguration der Farben von Elementen . . . 89
3.11.2 Konfiguriation der Farben und des Routings von Verbindungen . . . 90
3.11.3 Übertragen des Styles (Pipette, Pinsel, Style List) . . . 90
3.11.4 Die Diagramm-Legende zur Einfärbung von Modell-Elementen und Konnektoren . . . 91
3.12 Weitere einfache Konfigurationen . . . 92
3.12.1 Tastenkombinationen konfigurieren (Short Key) . . . 92
3.12.2 Vorhandene Toolbars anpassen und eigene erstellen . . . 93
3.12.3 DropDown-Listen konfigurieren . . . 93

4 Die Projektstruktur 95
4.1 Grundlegende Gedanken zum Aufbau der Project Browser Struktur . . . 95
4.1.1 Wie strukturiere ich das Modell? . . . 95
4.1.2 Das Root Package . . . 97
4.1.3 Das View Package . . . 98
4.2 Verlinkungsmuster für die Navigation im Diagramm . . . 98
4.3 Verfeinerung durch Composite-Elemente und Zwischendiagramm . . . 99
4.4 Überlegungen zu Traceability . . . 101
4.4.1 Tracen mittels Tastenkombinationen . . . 102
4.4.2 Traces mittels Beziehung . . . 102
4.4.3 Tracen mittels Custom Reference . . . 103
4.4.4 Tracen mittels Tagged Values . . . 103
4.4.5 Tracen mittels Hyperlinks . . . 104
4.5 Beispiele für Trace-Ketten . . . 105
4.5.1 Trace-Ketten mittels Realization Beziehung . . . 105

5 UML mit Enterprise Architect 109
5.1 UML - Hintergrund, Überblick und Zusammenhänge . . . 109
5.1.1 Woher kommt die UML? . . . 109
5.1.2 Diagramm und 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) . . . 118
5.2.3 Dependency . . . 119
5.2.4 Abstraction . . . 120
5.2.5 Die Usage Beziehung . . . 121
5.2.6 Der InformationFlow . . . 121
5.2.7 Die Generalisierung (Generalisation) . . . 122
5.2.8 Die Realization Beziehung . . . 123
5.3 Use Cases Modellierung . . . 124
5.3.1 Die Metainformationen eines Use Case . . . 125
5.3.2 Das Use Case Szenario . . . 125
5.3.3 Der Auslöser (Trigger) eines Use Case . . . 126
5.3.4 Die Include-Beziehung zwischen Use Cases . . . 126

5.3.5 Die Extend-Beziehung zwischen Use Cases . . . 127
5.3.6 Der Erweiterungspunkt (Extension-Point) eines Use Case . . . 128
5.3.7 Die Vor- und Nachbedingungen eines Use Case . . . 129
5.4 Aktivitäten-Modellierung (Prozesse, Workflows und Algorithmen) . . . 130
5.4.1 Anwendungsbereiche des Aktivitätsdiagramms . . . 130
5.4.2 Das Token-Konzept im Aktivitätsmodell . . . 131
5.4.3 Grundlegende Modell-Elemente im Aktivitätsdiagramm . . . 132
5.4.4 Datenflüsse modellieren . . . 135
5.4.5 Prozessschritte (Aktionen) verfeinern . . . 139
5.4.6 Nebenläufigkeit modellieren . . . 142
5.4.7 Den Kontext einer **Aktion** modellieren . . . 142
5.4.8 Asynchrone Kommunikation . . . 144
5.5 Klassen-Modellierung (Strukturen und Daten) . . . 148
5.5.1 Anwendungsbereiche des Klassendiagramms . . . 148
5.5.2 Die wichtigsten Beziehungen der Klasse . . . 149
5.5.3 Die Klasse . . . 149
5.5.4 Klassen in Klassen (nested Classes) . . . 150
5.5.5 Die Features der Klasse (Attribute, Operationen und Receptions) . . . 151
5.5.6 Redefinieren von Features (Attribute und Operationen) . . . 154
5.5.7 Schnittstellen (Interface) . . . 154
5.5.8 Die Realisierungsbeziehung (Realization) . . . 155
5.5.9 Erstellen eines Domänenmodells . . . 155
5.5.10 Diskussion von Modellierungs-Varianten im Klassendiagramm . . . 156
5.6 Objekte modellieren (Instance Specification) . . . 158
5.6.1 Unterschied Instanz, Property und Port, etc. . . . 158
5.7 Komponenten-Modellierung . . . 159
5.7.1 Die Einfache Komponente (Basic Component) . . . 160
5.7.2 Beziehungen zwischen Komponenten . . . 165
5.7.3 Die Packaging Component . . . 166
5.8 Kompositions-Struktur Diagramm (Composite Structure Diagram) . . . 166
5.8.1 Verwendung von Parts (Property Elemente) . . . 167
5.8.2 Property vs. Instanz . . . 168
5.8.3 Komponenten in Komponenten schachteln . . . 169
5.8.4 Verbindung von Ports in Komponenten . . . 170
5.9 Das Deployment Diagramm . . . 171
5.9.1 Das Artefakt (Artifact) . . . 171
5.9.2 Die physische Welt modellieren (Node, Device, Execution Environment) . . . 172
5.9.3 Der Link zwischen logischer und physischer Welt . . . 172
5.10 Interaktionen modellieren . . . 172
5.10.1 Das Sequenz-Modellierung . . . 172
5.11 Zustandsautomaten (State Machine) modellieren . . . 179
5.11.1 Einfache Zustandsautomaten . . . 179
5.11.2 Beispiel Verkehrsampel . . . 187

6 SysML mit Enterprise Architect 189
6.1 SysML: Hintergrund, Überblick und Zusammenhang mit UML . . . 189
6.2 Wichtige SysML Beziehungen . . . 189
6.2.1 Allocation Beziehung . . . 189
6.3 SysML spezifische Konfigurationen im Enterprise Architect . . . 190
6.3.1 SysML als primäre Technologie konfigurieren . . . 190
6.3.2 SysML Versionen migrieren . . . 190
6.4 Das Block Definition Diagramm (BDD) . . . 190
6.4.1 SysML Ports . . . 191
6.4.2 Das SysML FlowProperty . . . 193
6.4.3 Port Beispiele . . . 193
6.4.4 *Aggregations*- und *Composition*-Beziehungen in SysML . . . 194
6.4.5 Multiplizität vs. mehrere Beziehungen . . . 196
6.5 Das Interne Block Definition Diagramm (IBD) . . . 196
6.5.1 Beispiel 1: Einfache Verdrahtung im IBD . . . 196

6.5.2 Beispiel 2: Verdrahtung mit Ports und Properties . . . 197
6.5.3 Modellieren von Varianten mit der Bound Reference . . . 198
6.6 Strukturelle System-Sichten . . . 198
6.6.1 Das Kontext-Diagramm . . . 198
6.6.2 Funktionale/Logische Dekomposition . . . 198
6.6.3 Physikalische Dekomposition . . . 198
6.6.4 Wirkketten (Event Chain) . . . 198
6.7 Das SysML Anforderungsmodell . . . 198
6.7.1 Das Anforderungs-Element (SysML *Requirement*) . . . 199
6.7.2 Das Testfall Element (*Test Case*) . . . 199
6.7.3 SysML Anforderungs-Beziehungen . . . 199
6.8 Das Parametric-Modell . . . 201
6.8.1 Der Constraint-Block . . . 201
6.8.2 Das Constraint-Model . . . 202
6.8.3 Simulation von Constraint-Modellen . . . 203
6.9 SysML Erweiterungen des UML-Aktivitäts-Modells . . . 203

7 Requirements-Engineering mit Enterprise Architect 205
7.1 EA Requirements . . . 205
7.1.1 Unterschiede im Property-Dialog . . . 205
7.2 Traceability von Anforderungen . . . 206
7.3 Der Specification Manager . . . 206
7.3.1 Konfiguration des Specification Managers . . . 207
7.4 Requirements Management Tool Integration . . . 207

8 Projekt-Management und Team-Kollaboration mit Enterprise Architect 209
8.1 Aufwandsabschätzung (Die Use Case Point Methode) . . . 209
8.2 Das Projekt-Glossar . . . 210
8.2.1 Glossareinträge erstellen und verwalten . . . 210
8.2.2 Das Glossar zwischen EA-Repositorien austauschen . . . 211
8.2.3 Ein Glossar für mehrere EA-Repositorien . . . 211
8.2.4 Das Glossar programmatisch ansprechen . . . 211
8.3 Diskussionsmöglichkeiten im Team . . . 212
8.3.1 Die Team Library (EA Forum) . . . 212
8.3.2 Metainformationen des Modell-Elements . . . 213
8.3.3 Diskussion am Modell-Element . . . 213
8.3.4 Model Chat . . . 213
8.3.5 Model Mail . . . 214
8.3.6 Der Projektkalender . . . 214
8.3.7 Das Model View Fenster . . . 214
8.4 Arbeit mit Ressourcen . . . 216
8.4.1 Ressourcen konfigurieren . . . 216
8.4.2 Ressourcen zuweisen . . . 216
8.4.3 Der Gantt View . . . 216

9 Arbeiten mit dem Code 219
9.1 Code-Generierung (Forward Engineering) . . . 219
9.1.1 Grundlegende Einstellungen bei der Codegenerierung . . . 219
9.1.2 Mapping zwischen UML Klasse und Code-File . . . 222
9.1.3 Generieren von Code für eine selektierte Klasse . . . 223
9.1.4 Generieren von Code für ein gesamtes Paket . . . 225
9.1.5 Generieren von Attributen und deren Eigenschaften . . . 226
9.1.6 Generieren von Beziehungen und deren Eigenschaften . . . 227
9.1.7 Generieren von **Operationen** und deren Eigenschaften . . . 231
9.1.8 Erzeugen von Namensräumen . . . 232
9.1.9 Generieren von #include, using, import bei Dependencies . . . 233
9.2 Erstellen von Code aus Verhaltensmodellen . . . 233
9.2.1 Generieren von Code aus Aktivitäts-Diagrammen (Activity Diagram) . . . 234
9.2.2 Generieren von Code aus Interaktions-Diagrammen (Sequence Diagram) . . . 235

9.2.3 Generieren von Code aus Zustandsautomaten (State Machine) . . . 235
9.3 Einlesen von Code (Reverse Engineering) . . . 237
9.3.1 Einlesen eines einzelnen Source-Files . . . 237
9.3.2 Einlesen eines gesamten Source-Code Verzeichnisses . . . 237
9.3.3 Einlesen von Binaries . . . 238
9.3.4 Umgang mit Kommentaren beim Roundtrip-Engineering . . . 239
9.4 Code-Generierung im Team . . . 240
9.5 Das Codegenerierungs-Framework . . . 241
9.5.1 Struktur der Codegenerierungs-Templates . . . 243
9.5.2 Die Substitutionsmakros . . . 244
9.5.3 Eigene „Makros“ schreiben . . . 245
9.5.4 Anpassungsbeispiel für das Code Generation Framework . . . 245
9.5.5 Der resultierende Code . . . 247

10 Modelltransformation 249
10.1 Eingebaute Modelltransformationen . . . 250
10.2 Das Modelltransformations-Framework . . . 252
10.2.1 Das Format des Intermediate-File . . . 252
10.2.2 Der Templatemechanismus der Modelltransformation . . . 254

11 Modellsimulation 259
11.1 Die wichtigsten Fenster für die Simulation . . . 259
11.1.1 Das Simulator Fenster . . . 259
11.1.2 Das Triggers Fenster . . . 260
11.1.3 Das Locals Fenster . . . 260
11.1.4 Das Simulation Breakpoints Fenster . . . 260
11.2 Interpretierte Simulation . . . 260
11.2.1 Interpretierte Simulation starten . . . 261
11.2.2 Der Simulation Code . . . 261
11.2.3 Aktivitäten Simulieren . . . 261
11.2.4 Zustandsautomaten simulieren . . . 261
11.2.5 Einbinden von User Interfaces . . . 261
11.2.6 Simulation mittels Execution Analyzer Script starten . . . 261
11.2.7 Simulation mittels EA API ausführen . . . 261
11.2.8 Verwendung von COM Objekten in der Simulation . . . 261
11.3 Executable Simulation . . . 261
11.3.1 Zustandsautomaten simulieren . . . 261
11.3.2 Einbinden von User Interfaces . . . 261
11.3.3 Code Generation Templates für die Executable Simulation . . . 261

12 Modellieren von Datenbanken 263
12.1 Arbeit mit den EA Legacy Datenbank Werkzeugen . . . 263
12.1.1 Logisches Datenbank-Modell im EA erstellen . . . 263
12.1.2 Logisches Modell in ein physisches Datenbank-Modell transformieren . . . 266
12.1.3 Physisches Datenbank-Modell im EA erstellen . . . 267
12.1.4 DDL Statements mit dem EA generieren . . . 269
12.1.5 Ein physisches Datenbank-Modell über eine ODBC Datenquelle erstellen . . . 270
12.2 Arbeiten mit dem Database Builder . . . 271
12.2.1 Verbindung zu einem DBMS aufbauen . . . 271
12.2.2 Tabellen mit dem Database Builder erstellen . . . 272
12.2.3 DDL Skripte generieren und ausführen . . . 272
12.2.4 Differenz zwischen Modell und Datenbank berechnen . . . 273

13 Modellieren von XML Schemata 275
13.1 XML Schema Modell . . . 275
13.1.1 Vorhandene XSD Files importieren . . . 275
13.1.2 Ein eigenes XML Schema Modell erstellen . . . 276
13.2 Der Schema Composer . . . 276
13.2.1 Schema Profile anlegen . . . 277

13.2.2 Eine Modellvariante erstellen . . . 278
13.2.3 Schema generieren . . . 279
13.2.4 Modell transformieren . . . 280

14 Modellvalidierung 283
14.1 Eingebaute Validierungsregeln . . . 284
14.1.1 Object Constraint Language (OCL) . . . 284
14.1.2 Regel-Kategorien automatisch aktivieren/deaktivieren (Validation Configuration) . . . 285
14.2 Validierung durch Suchen . . . 285
14.3 Validierung mittels Modell-Simulation . . . 286
14.4 Validierung durch Metamodell-Regeln . . . 286
14.5 Eigene Validierungsregeln schreiben . . . 286
14.5.1 Validierungs-Regeln als Skript schreiben . . . 287
14.5.2 Validierungs-Regel als Add-In schreiben . . . 287

15 Generieren von Dokumenten 291
15.1 Der RTF-Template Mechanismus . . . 292
15.1.1 Weitere Arten von Templates . . . 295
15.1.2 Formatvorlagen anpassen (Edit Style...) . . . 295
15.1.3 Überschriften-Nummerierung mittels Listen konfigurieren . . . 295

16 Versionieren von Modellen 299
16.1 Arbeit mit Baselines (interne Versionierung) . . . 299
16.2 Anbindung an externe Versionskontrollsysteme . . . 301
16.3 Branching von Modellen . . . 302
16.3.1 Horizontales Branching . . . 302
16.4 Auditing von Modellen . . . 307
16.4.1 Speicherbedarf . . . 308
16.4.2 Audit-Logs verwalten . . . 308
16.4.3 Im Audit suchen . . . 308

17 Teamarbeit - Configuration Management 309
17.1 Shared Model . . . 310
17.2 Private Model . . . 311
17.2.1 Private Model mittels Paketen unter Versionskontrolle (VCS) . . . 312
17.3 Kombination aus Shared und Private Model . . . 314
17.3.1 Shared Model mit externer Versionkontrolle . . . 314
17.3.2 Kombination mehrerer Shared Models mit externer Versionskontrolle . . . 314
17.4 Shared Model mit XMI Export/Import und Projekt Transfer . . . 315
17.5 EA Cloud und Pro Cloud . . . 316
17.5.1 EA Cloud Service Konfiguration . . . 316
17.5.2 Aktivieren des Pro Cloud Servers . . . 318
17.5.3 Verbindung mit der EA Cloud aufbauen . . . 318
17.6 Versionieren mit LemonTree . . . 318
17.7 WebEA . . . 318
17.7.1 WebEA einrichten . . . 319
17.8 Modell Replikat . . . 319
17.8.1 Erstellen eines neuen Master-Projektes . . . 320
17.8.2 Erstellen eines neuen Replikats . . . 320
17.8.3 Synchronisieren von Replikaten mit dem Master-Projekt . . . 320
17.8.4 Hinweise zum praktischen Einsatz von Replikaten . . . 321
17.9 Modelle per XMI austauschen . . . 321
17.9.1 XMI-Export . . . 323
17.9.2 XMI-Import . . . 324
17.9.3 Automatisches Zusammenführen (Mergen) beim XMI-Import . . . 324
17.10 CSV Export / Import . . . 329
17.10.1 Import/Export Specification definieren . . . 329
17.10.2 CSV Import/Export durchführen . . . 329
17.11 Paketübergreifende Referenzen . . . 330