
Compendium Enterprise Architect from Sparx Systems

Foundational – Advanced – PROFESSIONAL - EXPERT
Edited edition for EA Version 15

We would very much appreciate your suggestions on improving this handbook. Please use the provided email addresses.

Vienna, November 2020

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

©SparxSystems Software GmbH Wien. All rights reserved. No part of this work may be reproduced in any form (photocopy, printing, microfilm or any other process) without the express written permission of the publisher, Sparxsystems Software, or processed, modified or reproduced using electronic systems.

These documents have been prepared and checked with great care. Unfortunately, errors cannot be excluded. The author assumes no responsibility or liability for incorrect information. Most screenshots have been created using Enterprise Architect 15, Build 1513 with “Microsoft® Office 2016” as application look; some differences to other builds may be apparent.

Electronic Book Search: booksearch.sparxsystems.eu

Blog: blog.sparxsystems.de

European Web Page: www.sparxsystems.de (German) , www.sparxsystems.eu (English)

Author



Dr. Horst Kargl was first introduced to object-oriented modeling and programming in 1998. Prior to his career at SparxSystems Software GmbH he was assistant professor at TU Wien (Vienna Technical University) where he worked on a number of research projects involving e-learning, semantic web and model-driven software development. His doctoral thesis examined the automatic integration of modeling languages and model transformation. In 2006 he joined the SparxSystems team in a freelancing role, and by 2008 he was working full-time as a consultant and trainer. His core focus areas are business and software engineering, software architecture, model-based software and systems development and leveraging the customization and extensibility potential of Enterprise Architect (EA).

Mail: Horst.Kargl@sparxsystems.eu

About this edition

This edition has been revised and adapted to EA version 15. Most of the functions described can also be used with older EA versions. The menu items may be in a different position or have a different name. In our blog (blog.sparxsystems.de) you will find articles with further information on translation tables between the different EA versions.

In this book, a distinction has been made between configurations and examples. A new icon for the configuration has been introduced for this. See chapter Conventions on page v. The individual exercise examples are now numbered to make it easier to exchange information about the examples they contain.

The book's index has also been completely revised to make it easy to look up and find the most important topics and terms. In addition to the index in the book, there is also an electronic online search in the book, which enables a full-text search in the book. You can find the book search at: booksearch.sparxsystems.eu

In addition to these additions, we have retained the current structure of the book. EA-specific topics, modeling languages and methods are marked separately. To do justice to the title "Compendium", the individual topics are described as independent information units in order to present the information briefly and concisely to the interested reader. The structure of the individual chapters is designed so that both serial reading of all chapters and specific reading of individual chapters is possible. References to other chapters help to read up missing knowledge.

Important points to be remembered have been summarized and marked. There are practical examples for almost every topic to understand. Since there is not just one truth and personal preferences often play a role, we did not want to withhold this from you and have enriched the book with comments from various people.

The title of the book is Compendium on Enterprise Architect from Sparx Systems, but also looks outside the box of Enterprise Architect and looks at approaches, methods and expansion options that the EA does not offer or not in the form, but which can be implemented with the EA. You will learn how to use the EA as your modeling platform and how to integrate the EA into your tool chain. The goal should be that the EA becomes your tool and supports your approach and processes in order to be able to implement your modeling approach.

This book version is written in \LaTeX . All menu references were generated from an EA model in \LaTeX . To make this possible, I would like to thank the following people for their support: Theresa Lieber, Florian Kalenda, Markus Gründler, Robert Sichert, Franz Ewald Müller, Dietmar Steinpichler, Orsolya Nemeth, Thomas Besorna.

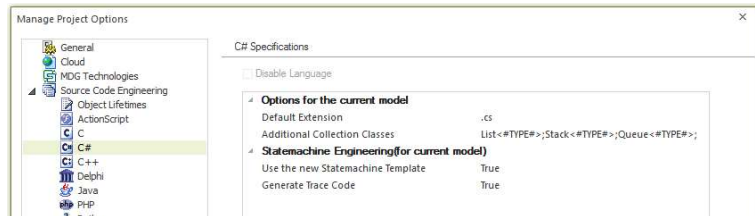
Glossary

Term	Description
(Element) Com- partment	A rectangular model element consists of several sections. These sections are called compartments. In UML::Class there is, for example, a Compartment for Name, Attributes, Operations, Notes, etc. The display can be configured in each case.
Composite Dia- gram	A diagram referenced by a composite element.
Composite-Element	EA model element that contains a hyperlink to a composite diagram.
Diagram	The diagram serves as a drawing surface in which model elements and connectors are visualized. The model is located in the project browser.
Diagram-Model- Element	The Diagram-Model-Element is a model element that does not appear in the Project Browser, but has almost all the properties of a “normal” model element.
Diagram-Element	A diagram element is a model element that is only visible on the diagram but not in the Project Browser and is represented like model elements by a diagram object.
Diagram-Object	The diagram object is the graphical representation of a model element on a diagram. A model element is represented on a diagram by a maximum of one diagram object.
Document- Template	A customizable template for generating documents in Enterprise Architect.
Enterprise Archi- tect (EA)	The modeling tool called “Enterprise Architect” from Sparx Systems.
EA-Repository	The database in which EA models are stored. We distinguish between file-based and database-based EA repositories.
EA-Package	An EA package is a special model element to structure models.
EA-Tag	An EA tag is an area used in the Document Generation Template that defines the context of a model element.
Connector/Link	Describes a relationship between model elements - in other words: a line between model elements. Depending on the modeling language (and diagram type) used, the connectors have different meanings.
Metamodel	A metamodel is a “scheme” or template for creating models. It defines the individual model elements, their semantics and structure.
Model	The model is the collection of all model elements and is located in the EA Project Browser.
(Root) Model	Each EA repository has at least one root model (root in the Project Browser) and can contain any number of them.

Term	Description
Model-Element	A model element designates any symbol of a modeling language. In other words: Anything that is not a line between model elements. Model elements are usually displayed in the Project Browser.
Modelling Approach	The way a modeling language is used to create a model in Enterprise Architect. This includes the dimensions: modeling language, modeling tool, method, experience. A concrete result of a modeling approach is the “reference model”.
Private Model	An EA repository that is used by an EA user alone, but exchanges models (parts) via version control.
Project Browser	The Project Browser is the filing structure for model elements and diagrams. It is similar to the Windows Explorer.
Project Metamodel	The general rules of how a model should be constructed.
Reference Model	The reference model is part of a modeling approach. It contains the model structure as well as examples for the modeling language to be used. Based on the reference model, modeling guidelines can be derived, which can then be implemented as automatically executable validation rules and automatisms to facilitate modeling.
Shared Model	An EA repository that is used and modified by multiple EA users simultaneously.
Shared Repository	An EA repository that shares project-specific information, such as the project glossary, with other EA repositories.
Structured elements	A structured element is a model element that can only exist together with another model element (the container). A port, ActionPin, etc. are such model elements.
View-Package	The View package is a special EA package. It has a view icon.
Quicklink	The little symbol at the upper right corner of a model element to create a connector to another model element. The quicklink menu is a suggestion of possible relationships.
«StereotypName»	A stereotype is a term from the UML language to give a UML element a stronger meaning (semantics).

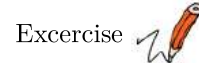
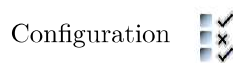
Conventions

In order to do justice to the title of this book and to facilitate quick reference and location of the most important information, we have provided you with some conventions.



Menu entries: References to menu entries are always placed within square brackets. For example, in the main menu: *[Configure > Model > Options > C# > Default Extension]*. Optional menu selections are shown inside angle brackets, separated by a vertical bar (|), as follows: *[Start > Workspace > Preferences > Links > <Routing | Pen Width>]*. If one menu item opens a dialog for continued reference, then this reference is denoted with a “->”. For example: *[Start > Desktop > Preferences -> Diagram > Behavior > Autosave Changes]*.

Symbols: These symbols are placed in the margins to help identify certain types of information.



Key combinations: Like menu entries, key combinations are also shown in square brackets, except that they are italicized. For example: *[Ctrl + Tab]*.

Name of Modelling-Language concepts: If special modeling language elements are mentioned in the text, they are displayed as follows: *Class*, *UseCase Actor*. Primitive data types are represented as well as modeling language elements: *Integer*, *int*, *bool*.

Names used in models: If a model example is referenced by name, the name will be displayed like this in the text: *„Einzugebender Text“*. If you refer to two classes with the name *Person* and *Address*, they are represented as follows: *„Person“*, *„Address“*.

Text to be entered on the user interface: Texts that are entered in the user interface, such as during a model search or a properties dialog of a model element, are highlighted as follows: *„Enter Text in User Interface“*.

User Interface Window Names & EA Features and Properties: If reference is made to a specific function or window of the user interface, the name is displayed as follows: *name of the user interface*. For example, this is the *Project Browser*, the *Traceability* view, or the *Status* property of a model element.

Text blocks that deal with special topics are highlighted.

Personal Comment

Name: Personal opinion and/or experience

Semantics of the modeling language

Modeling language specific explanation (UML, SysML, BPMN, etc.)

Code (Script, C#, SQL, etc.)

Considerations about the modeling method

TEXT DESCRIBES A SPECIAL PROCEDURE OR METHOD.

Contents

Author Information	i
Glossary	iii
Conventions	v
1 Why modeling, and why with EA?	1
1.1 Why do we use a modeling tool?	2
1.2 Enterprise Architect implementation options	3
1.3 Why do we model?	4
1.3.1 Simplified representation of complex systems	5
1.3.2 Basis for communication	5
1.3.3 Generating documents	5
1.3.4 Transforming models	6
1.3.5 Generating code	6
1.3.6 Importing existing code	6
1.3.7 Verification of models	7
1.3.8 Simulation of models	7
1.3.9 Automatic execution of changes	7
1.3.10 Generating test cases	7
1.4 Hurdles when introducing a model-based approach	7
1.4.1 Which modeling language should be used?	9
1.4.2 The selected tool	9
1.4.3 What does the “project metamodel” of the model to be created look like?	10
1.4.4 How do I facilitate the creation of the model that matches the project metamodel?	10
1.4.5 At what level of abstraction should modeling be carried out?	12
1.4.6 How much detail is required?	12
1.4.7 How complete should the model be?	12
1.5 Introducing a modeling approach into the company	12
1.5.1 Possible scenarios	13
1.5.2 An iterative implementation process	14
2 Basic Configuration	17
2.1 Enterprise Architect Editions and Licenses	17
2.2 Initial Installation	18
2.2.1 Floating license “portability”	19
2.2.2 Trial Version expired?	20
2.3 Setup and Basic Structure of Enterprise Architect	20
2.3.1 Visual Style	21
2.3.2 Ribbon Menu	22
2.3.3 Perspective	22
2.3.4 Windows	22
2.3.5 Workspace Layout	23
2.3.6 Toolbar	23
2.3.7 MDG Technology	23
2.3.8 Keyboard Shortcuts to Work Faster)	24
2.4 Settings that Make Working with EA Easier	24

3	First Steps	27
3.1	Creating a New Project	27
3.2	Working with Predefined Templates (The Model Wizard)	28
3.3	Creating Diagrams and Models	28
3.3.1	Creating a Package	28
3.3.2	Creating a Diagram	28
3.3.3	Deleting Diagram Contents and Diagrams	29
3.3.4	Creating a Model Element	30
3.3.5	Creating Relationships	31
3.4	The Difference Between the Model and Diagrams	32
3.5	Diagram Layout	33
3.5.1	Manual Layout	33
3.5.2	Automatic Layout	33
3.6	The Most Important Functions and Windows	34
3.6.1	The Project Browser	34
3.6.2	The Diagram Toolbox	35
3.6.3	The Traceability Window	35
3.6.4	The Element Browser	36
3.6.5	The Element Properties Dialog	38
3.6.6	The Dockable Property Window	41
3.6.7	The Connector Properties Dialog	42
3.6.8	The Diagram Properties Dialog	44
3.6.9	The Diagram Filters & Layers Window	44
3.6.10	The Insert Related Elements Dialog	45
3.6.11	The Notes Window	46
3.6.12	The Dynamic Document Window	47
3.6.13	The Tagged Value Window	48
3.6.14	The Relationships Window	49
3.6.15	The Relationship Matrix	49
3.6.16	Configuration as Project Resources Relationship Matrix	50
3.6.17	Matrix Overlay	51
3.6.18	Matrix Configuration as a Separate Model Element	52
3.6.19	Calling up the Relationship Matrix from the Diagram	52
3.6.20	The Package Table View	53
3.6.21	The Grouping Function of the Table View	53
3.6.22	The Parent & Interface Dialog	54
3.6.23	Arrange Windows and Save as Workspace Layout	55
3.6.24	Inserting Files as Model Elements	55
3.6.25	The Navigator Window	55
3.6.26	The Portals Window	56
3.7	The Ribbon Menu	57
3.7.1	Create Custom Ribbon Set	57
3.7.2	Select a Ribbon Set	57
3.7.3	Enable All Ribbons	57
3.8	EA Perspective	58
3.8.1	Configure Perspectives	58
3.8.2	Perspective - Group Restrictions	59
3.8.3	Activate all perspectives	60
3.9	Basic Modeling Concepts in EA	60
3.9.1	Stereotypes and Tagged Values	60
3.9.2	Type - Instance - Relationships (Classified Element)	62
3.9.3	Namespaces and Naming Conventions	64
3.9.4	Hiding and Displaying the Complete Namespace	65
3.9.5	The Multiplicity of Connectors	65
3.9.6	Hyperlinks	66
3.9.7	Notes	67
3.9.8	Automatic Nesting in the Project Browser	69
3.10	The Basic Data Structure and Operations in EA	69
3.10.1	The Diagram (is not the Model)	70
3.10.2	The Model Element	73
3.10.3	Additional Model Element Properties	73
3.10.4	The Child Element (Sub-Element)	73
3.10.5	The Structured Element	74

3.10.6	The Diagram Element	77
3.10.7	The Connector	78
3.10.8	The Package	80
3.10.9	Restructuring Packages	80
3.10.10	Versioning of Packages	80
3.10.11	XML as a Basic Exchange Format	81
3.10.12	Managing Stereotypes	81
3.10.13	Creating an ad-hoc Stereotype	81
3.10.14	Stereotype via UML Profile	82
3.11	Using Colors in Enterprise Architect	83
3.11.1	Configuring Element Colors	83
3.11.2	Configuring Connector Colors and Routing Style	84
3.11.3	Transferring the Style (Pipette, Brush, Style List)	85
3.11.4	The Diagram Legend for Coloring Model Elements and Connectors	85
3.12	Other Simple Configurations	86
3.12.1	Configuring Key Combinations (Shortcut Key)	87
3.12.2	Customizing Existing Toolbars and Creating Your Own	87
3.12.3	Configuring Dropdown Lists	87
4	The Project Structure	89
4.1	What to consider when structuring the Project Browser	89
4.1.1	How should I structure the model?	89
4.1.2	The Root Package	90
4.1.3	The View Package	91
4.2	Linking Pattern for Navigation in the Diagram	91
4.3	Refinement through Composite Elements and Intermediate Diagram	93
4.4	Traceability Considerations	95
4.4.1	Tracing using keyboard shortcuts	95
4.4.2	Traces using relationship	96
4.4.3	Tracing with Custom Reference	96
4.4.4	Tracing using Tagged Values	97
4.4.5	Tracing via hyperlinks	97
4.5	Examples for Trace Chains	98
4.5.1	Trace chains using realization relationship	98
5	UML with Enterprise Architect	101
5.1	UML Background, Overview and Interrelationships	101
5.1.1	Where does UML come from?	101
5.1.2	Diagram and Model	101
5.1.3	Core UML Concepts	102
5.1.4	The UML Language Family	102
5.1.5	Connection between the individual UML Models	104
5.2	Frequently-used Relationships	104
5.2.1	Association Relationship	104
5.2.2	Part-whole Relationship (Aggregation and Composition)	109
5.2.3	Dependency Relationship	110
5.2.4	Abstraction Relationship	111
5.2.5	The usage Relationship	112
5.2.6	The InformationFlow	112
5.2.7	The Generalization Relationship	113
5.2.8	The Realization Relationship	114
5.3	Use Case Model	115
5.3.1	The Metainformation of the Use Case	116
5.3.2	The Use Case Scenario	116
5.3.3	The Trigger of a Use Case	117
5.3.4	The Include Relationship between Use Cases	117
5.3.5	The Extend Relationship between Use Cases	118
5.3.6	The Extension Point of a Use Case	118
5.3.7	The Pre- and Post-conditions of a Use Case	120
5.4	Modelling Activities (Processes, Workflows and Algorithms)	120
5.4.1	Application Areas of the Activity Diagram	120
5.4.2	The Token Concept in the Activity Model	121
5.4.3	Basic Model Elements in the Activity Diagram	122

5.4.4	Modelling Data Flows	125
5.4.5	Refining process steps (Actions)	128
5.4.6	Concurrency modelling	131
5.4.7	Modelling the Context of an Action	132
5.4.8	Asynchronous Communication	133
5.5	Modelling Classes (Structure and Data)	137
5.5.1	Areas of Application of the Class Diagram	137
5.5.2	The most important Relationships of the Class	138
5.5.3	The Class	138
5.5.4	Classes in Classes (nested Classes)	139
5.5.5	Class Features (Attributes, Operations and Receptions)	139
5.5.6	Redefining Features (Attributes and Operations)	143
5.5.7	Interface	143
5.5.8	The Realization Relationship	143
5.5.9	Creating a Domain Model	144
5.5.10	Discussion of Modelling Variants in the Class Diagram	144
5.6	Modelling Objects (Instance Specification)	146
5.6.1	Setting Concrete Values for Instances (Set Run State)	146
5.6.2	Differences among Instance, Property and Port, etc.	147
5.7	Modelling Components	148
5.7.1	The Basic Component	148
5.7.2	Relationships between Components	153
5.7.3	The Packaging Component	154
5.8	Composite Structure Diagram	154
5.8.1	Using Parts (Property Elements)	155
5.8.2	Property vs. Instance	156
5.8.3	Nesting Components within Components	156
5.8.4	Connecting Ports inside Components	157
5.9	The Deployment Diagram	158
5.9.1	The Artifact	158
5.9.2	Modelling the Physical World (Node, Device, Execution Environment)	159
5.9.3	The Link between logical and physical World	159
5.10	Model interactions	159
5.10.1	Modelling Sequences	159
5.11	State Machine Design	166
5.11.1	Simple State Machine	166
5.11.2	Example Traffic Light	173
6	SysML with Enterprise Architect	175
6.1	SysML: SysML background, overview and relationship to UML	175
6.2	Important SysML Relationships	175
6.2.1	Allocation Relationship	175
6.3	SysML-specific Configurations in Enterprise Architect	176
6.3.1	Configuring SysML as the Primary Technology	176
6.3.2	Migrating SysML Versions	176
6.4	The Block Definition diagram (BDD)	176
6.4.1	SysML Ports	177
6.4.2	The SysML FlowProperty	179
6.4.3	Port Examples	179
6.4.4	<i>Aggregation</i> and <i>Composition</i> Relationships in SysML	180
6.4.5	Multiplicity vs. several Relationships	181
6.5	The Internal Block Definition Diagram (IBD)	181
6.5.1	Example 1: Simple Wiring in the IBD	182
6.5.2	Example 2: Wiring with ports and properties	183
6.5.3	Modelling of Variants with Bound Reference	183
6.6	The SysML Requirements Model	183
6.6.1	The Requirement Element (<i>SysML Requirement</i>)	184
6.6.2	The Test Case Element (<i>Test Case</i>)	184
6.6.3	SysML Requirement Relationships	184
6.7	The Parametric Model	186
6.7.1	The Constraint Block	186
6.7.2	The Constraint Model	186
6.7.3	Simulation of Constraint Models	187

6.8	SysML Extensions of the UML Activity Model	187
7	Requirements Engineering with Enterprise Architect	189
7.1	EA Requirements	189
7.2	Traceability of Requirements	190
7.3	The Specification Manager	190
7.3.1	Configuration of the Specification Manager	191
7.4	Requirements Management Tool Integration	191
8	Project Management and Team Collaboration	193
8.1	Estimating Effort (The Use Case Point Method)	193
8.2	The Project Glossary	193
8.2.1	Creating and Managing Glossary Entries	194
8.2.2	Exchanging the Glossary between EA Repositories	194
8.2.3	One Glossary for Multiple EA Repositories	194
8.2.4	Addressing the Glossary Programmatically	195
8.3	Ways to Handle Team Discussions	195
8.3.1	The Team Review	195
8.3.2	Model Element Meta-Information	196
8.3.3	Discussion on a Specific Model Element	197
8.3.4	Model Chat	197
8.3.5	Model Mail	197
8.3.6	The Project Calendar	198
8.3.7	The Model View Dialog	198
8.4	Working with Resources	199
8.4.1	Configuring Resources	199
8.4.2	Allocating Resources	200
8.4.3	The Gantt View	200
9	Working with the Code	201
9.1	Code Generation (Forward Engineering)	201
9.1.1	Basic Settings for Code Generation	201
9.1.2	Mapping between UML Classes and Code File	204
9.1.3	Generating Code for a Selectable Class	205
9.1.4	Generating Code for an Entire Package	207
9.1.5	Generation of Attributes and their Properties	207
9.1.6	Generation of Relationships and their Properties	209
9.1.7	Generating Operations and their Properties	212
9.1.8	Generating Namespaces	213
9.1.9	Generating from #include, Using, Import in Case of Dependencies	214
9.2	Generating Code from Behavioral Models	214
9.2.1	Generating Code from Activity Diagrams	215
9.2.2	Generating Code from Sequence (Interaction) Diagrams	216
9.2.3	Generating Code from State Machines	217
9.3	Importing code (Reverse Engineering)	218
9.3.1	Importing a Single Source File	218
9.3.2	Importing an Entire Source Code Directory	218
9.3.3	Import of Binaries	219
9.3.4	Handling Notes (Comments) during Roundtrip Engineering	219
9.4	Generating Code in a Team Environment	221
9.5	The Code Generation Framework	222
9.5.1	Structure of the Code Generation Template	223
9.5.2	The Substitution Macros	224
9.5.3	Writing Your Own Macros	225
9.5.4	Customization Example for the Code Generation Framework	226
9.5.5	The Resulting Code	228
10	Model Transformations	229
10.1	Built-In Model Transformation	230
10.2	The Model Transformation Framework	232
10.2.1	The Intermediate File Format	232
10.2.2	The Model Transformation Template Mechanism	234

11 Model Simulation	237
11.1 The Most Important Windows for Simulation	237
11.1.1 The Simulator Window	237
11.1.2 The Triggers Window	238
11.1.3 The Locals Window	238
11.1.4 Das Simulation Breakpoints Fenster	238
11.2 Interpretierte Simulation	239
11.2.1 Interpretierte Simulation starten	239
11.2.2 The Simulation Code	239
11.2.3 Simulate Activities	240
11.2.4 Simulate State Machines	240
11.2.5 Incorporate User Interfaces	241
11.2.6 The most important at a glance	241
12 Modeling Databases	243
12.1 Working with EA Legacy Database Tools	243
12.1.1 Creating a Logical Database Model in EA	243
12.1.2 Transforming a Logical Model into a Physical Database Model	246
12.1.3 Creating a Physical Database Model in EA	246
12.1.4 Generating DDL Statements out of EA	249
12.1.5 Creating a Physical Database Model via an ODBC Data Source	250
12.2 Working with the Database Builder	250
12.2.1 Establishing a connection to a DBMS	250
12.2.2 Creating Tables with the Database Builder	251
12.2.3 Generating and Running DDL Scripts	251
12.2.4 Calculating the Difference between Model and Database	252
13 Modeling XML Schemas	255
13.1 XML Schema Model	255
13.1.1 Importing Existing XSD Files	255
13.1.2 Creating Your Own XML Schema Model	256
13.2 The Schema Composer	257
13.2.1 Creating a Model Variant	257
13.2.2 Create a Model Variant	258
13.2.3 Generating a Schema	259
13.2.4 Model Transformation	260
14 Model Validation	263
14.1 Built-In Validation Rules	264
14.1.1 Object Constraint Language (OCL)	264
14.1.2 Automatically activate/deactivate rule categories (Validation Configuration)	265
14.2 Validation by search	265
14.3 Validation by Model Simulation	266
14.4 Write your own Validation Rules	266
14.4.1 Writing Validation Rules as Script	266
14.4.2 Write Validation Rule as Add-In	267
15 Generating Documents	269
15.1 The Document Template Mechanism	270
15.1.1 Other Kinds of Templates	272
15.1.2 Edit Style...	273
15.1.3 Defining a New List	273
16 Version Control for Models	277
16.1 Working with Baselines (internal Versioning)	277
16.2 Connection to external Version Control Systems	279
16.3 Branching of Models	279
16.3.1 Horizontal Branching	279
16.3.2 Vertical Branching	282
16.4 Auditing	284
16.4.1 Storage Requirements	285
16.4.2 Managing Audit Logs	285
16.4.3 In-Audit Search	285

17 Teamwork & Model Management	287
17.1 Shared Model	288
17.2 Private Model	289
17.2.1 Private Model using Packages under Version Control (VCS)	289
17.2.2 Distribution of Private Models for use with VCS Configuration	291
17.3 Combination of Shared and Private Model	291
17.3.1 Shared Model with External Version Control	292
17.3.2 Combining Multiple Shared Models with External Version Control	292
17.4 Shared Model with XMI Export/Import and Project Transfer	293
17.5 EA Cloud and Pro Cloud	293
17.5.1 EA Cloud Service Configuration	294
17.5.2 Activating the Pro Cloud Server	295
17.5.3 Establishing a Connection to EA Cloud	295
17.6 Versioning with LemonTree	295
17.7 WebEA	296
17.7.1 Set up WebEA	296
17.8 Replica Model	297
17.8.1 Creating a new Master Project	297
17.8.2 Creating a New Replica	297
17.8.3 Synchronizing Replicas with the Master Project	297
17.8.4 Tips for the Practical Use of Replicas	298
17.9 Exchanging Models via XMI	298
17.9.1 XMI-Export	299
17.9.2 XMI-Import	300
17.9.3 Automatic Merging on XMI Import	301
17.10 CSV Import/Export	305
17.10.1 Define a Import/Export Specification	305
17.10.2 Execute CSV Import/Export	305
17.11 Cross Package References	306
17.11.1 Handling Cross Package References	307
17.11.2 Working with Activated Cross Package Reference Tracking	308
17.11.3 Possible Usages	309
17.12 Project Transfer	309
17.13 Managing Submodels in Different Repositories	310
17.13.1 Placing Model Branches under Version Control	311
17.13.2 Importing a Model Branch into another EA Repository	311
17.14 EA Security	312
17.14.1 Configuring User and Group Rights	312
17.14.2 Using Windows Authentication	313
17.14.3 Activating Require User Lock to Edit	313
17.14.4 Locking the Model	313
17.14.5 Locking in default Locking Mode	314
17.15 Handling Project-Specific Configurations	315
18 Extend and Customize Enterprise Architect	317
18.1 Configurations	317
18.1.1 Tool-Specific Configurations	317
18.1.2 Project-specific Configurations	319
18.2 MDG Technologies	323
18.2.1 What can all be integrated into an MDG Technology?	323
18.2.2 Create Custom MDG Technologies	323
18.3 (UML-)Profiles	325
18.3.1 What is a UML profile? (OMG Metamodel Layer)	325
18.3.2 Create MDG-Technology-Project	326
18.3.3 Language Profiles (UML Profiles)	326
18.3.4 Create Quicklink Configuration	327
18.3.5 Toolbox Profiles	327
18.3.6 Diagram Profiles	329
18.3.7 Metamodel Rule Profile (Metamodel Constraints)	330
18.3.8 View Specification Profile	331
18.3.9 Configure MDG Technology as Active Technology	331
18.4 Create Design Pattern	332
18.4.1 Save Own Pattern	332

18.4.2	Use Pattern	332
18.5	Template Packages (Small Profile)	333
18.6	Model Search	333
18.6.1	Use Existing Searches	334
18.6.2	Create and Edit Query Builder Searches	334
18.6.3	The EA Database Schema	336
18.6.4	CreateSQL Search	339
18.6.5	Create Add-In Search	340
18.6.6	SQL example search	340
18.6.7	Search - Simple Locked Elements	340
18.6.8	Search - Orphaned Composite Elements	340
18.7	The EA Object Model	341
18.7.1	The Repository Class	341
18.7.2	The Package Class	342
18.7.3	The Element Class	343
18.7.4	The EA.Collection	343
18.7.5	The Connector Class	344
18.7.6	The ConnectorEnd Class	345
18.7.7	The DiagramObject Class	345
18.7.8	The DiagramLink Class	346
18.7.9	The Project Interface	346
18.8	Scripts	346
18.8.1	Normal Script (Normal Group)	347
18.8.2	Diagram Script (Diagram Group)	347
18.8.3	Project Browser Script (Project Browser Group)	347
18.8.4	Workflow Script (Workflow Group)	347
18.8.5	Search Script (Search Group)	348
18.8.6	Find in Project Script (Model Search Group)	348
18.8.7	Diagram Filter Script (Diagram Filter Group)	348
18.8.8	Link Script	348
18.8.9	Package Script	348
18.8.10	Element Script	349
18.9	Hybrid Scripts	349
18.10	External EA Automation	349
18.11	EA Add-In Development	350
18.11.1	What does an Add-In offer	350
18.11.2	Steps necessary to create an Add-in	351
18.11.3	Add-In Debugging	352
18.11.4	Write Add-In Model Search	352
18.12	The Modeling Assistants	352
	Index	353
	Literature	356