

Inhaltsverzeichnis

1 Anwendungen der Modellierung in der Programmierung:

Modeling4Programming	1
1.1 Papyrus Framework zur Modellierung mit SysML/UML	3
1.1.1 UML-Aktivitätsdiagramme mit Eclipse-Papyrus	3
1.1.2 Zustandsdiagramme mit Eclipse-Papyrus	4
1.1.3 Erstellen von Sequenzdiagrammen mit Eclipse-Papyrus	5
1.1.4 Erstellen von Anwendungsfalldiagrammen mit Eclipse-Papyrus	5
1.2 Obeo-UML-Designer	7
1.2.1 Visualisieren der Diagramme auf Basis von UML 2.5	7
1.2.2 Überblicke über UML-Diagramme mit Eclipse-UML-Designer	13
1.2.2.1 Strukturelle Diagramme	14
1.2.2.2 Verhaltenbasierte Diagramme	15
1.2.3 Beispiele von UML-Diagrammen	15
1.2.3.1 Klassendiagramme für Stromversorgungstester	15
1.2.3.2 Komponentendiagramme für die Websystemqualität	16
1.2.3.3 Zustandsdiagramme für die Websystemqualität	16
1.2.3.4 Profildiagramme für Parallelisierungsprozesse für den Asynchronmotor	17
1.2.3.5 Verteilungsdiagramme oder Deployment-Diagramm	18
1.3 UML-SysML-Struktur	20
1.3.1 Blockdefinitionsdiagramme	20
1.3.2 Interne Blockdiagramme	22
1.3.3 Anforderungsdiagramme	23
1.3.4 Zusicherungsdiagramme	23

1.4	IT-Lösungen mit „ <i>Modeling4Programming</i> “	25
1.4.1	Modellierung von IT-Lösungen mit C++.....	26
1.4.1.1	Anwendungen der Funktionalitäten der Elektronik in der Modellierung.....	26
1.4.1.2	Modellierungsaspekte mithilfe der Programmierung	34
1.4.1.2.1	Aktivitätsdiagramm.....	34
1.4.1.2.2	Sequenzdiagramm.....	39
1.4.1.2.3	Kommunikationsdiagramm.....	39
1.4.1.2.4	Zustandsdiagramm	40
1.4.1.2.5	Inneres Klassendiagramm.....	41
1.4.2	Modellierungen von Java-Anwendungen mithilfe von UML	51
1.4.2.1	Modellierung von Anwendungen für den Einsatz von Resonanzelementen Kondensator und Spule mit Eclipse-UML-Designer.....	51
1.4.2.2	Anwendung der Programmierungsperformance in der Berechnung der Kenngröße Wirkungsgrad des Motors.....	57
1.4.2.3	Modellierung der Vermeidung der Kollision zwischen gleichnamigen Methoden aus zwei unterschiedlichen Interfaces	61
1.4.3	Umsetzung der funktionellen Modellierung in der Programmierung.....	67
1.4.3.1	Java für die funktionelle Modellierung.....	67
1.4.3.1.1	Lambda-Ausdrücke zur Modellierung der Berechnungen	68
1.4.3.1.2	Modellierung der funktionellen Berechnung mithilfe von konstanten Eingaben.....	73
1.4.3.2	C++ für die funktionelle Modellierung.....	75
1.4.3.2.1	Modellierung der Entfernung eines gezielten Elements im Vektor	75
1.4.3.2.2	C++-Standardbibliothek mit Funktions-Template zum Modellieren der Funktionalität des Wirkungsgrades.....	77
1.4.3.2.3	„C++-Standardbibliothek“ mit Funktions-„ <i>Template</i> “, Iteratoren und Überladen der „ <i>Operatoren</i> “ zum Modellieren der Orthogonalität zwischen zwei Vektoren	79
1.4.3.2.4	Lambda-Funktion zum Darstellen der Derivation einer Funktion	82
1.4.3.2.5	Anwendungen der C++-Standardbibliothek in der Modellierung der Parallelisierung ...	84

1.4.3.2.6	Anwendungen der C++-Standardbibliothek in der Implementierung der Funktion „ <i>std::tuple()</i> “	89
1.4.3.2.7	Anwendung der „ <i>std::map</i> “-Objekte in der Analyse der Elemente einer Sammlung	91
1.4.3.2.8	Modellierung der Ausnahmebehandlung mithilfe der Ein- und Ausgabemöglichkeit	94
1.4.3.2.9	Modellierung des Überladens der Operatoren < und << mithilfe des „ <i>set</i> “-Containers	96
1.4.3.2.10	Modellierung der Anwendung der Funktion „ <i>evaluate()</i> in der Analyse der Zeiger auf Funktionen“	98
1.4.3.2.11	Modellierung der Funktionalität von „ <i>std::for_each()</i> “ zum Darstellen der Lambda-Ausdrücke-Rolle	100
1.4.3.2.12	Anwendungen der Funktionen „ <i>std::copy()</i> “ und „ <i>std::transform</i> “ zum Darstellen der Parallelisierung	102
1.5	Software-Architektur mit Papyrus und UML-Designer	105
1.5.1	Modellierung eines Klassendiagramms mithilfe vom Open Source Eclipse-Papyrus zum Analysieren eines Testprogramms mit Junit	105
1.5.1.1	Modellierung eines Testsystems für die Energietools	105
1.5.1.2	Modellierung eines Testsystems für WLAN-Systeme	109
1.5.2	Modellierung des Klassendiagramms zum Beschreiben der parametrisierten Systeme mit Eclipse-Ecore-Framework	112
1.5.3	Modellierungen der parallelen Implementierungen von Interface	121
1.5.4	Modellierung der Funktionalitäten der Pattern-Methoden mithilfe des Klassendiagramms von Eclipse-UML-Designer	134
1.5.5	Modellierung der Anwendungen des Interfaces „ <i>Collection</i> “ mit dem Klassendiagramm von Eclipse-UML-Designer	138
1.6	Zusammenfassung	142
	Literatur	144
2	UML-Modellierung mit der Eclipse-Umgebung	145
2.1	Modellierung des Klassendiagramms mit Obeo-UML-Designer	145
2.1.1	Vererbung	151
2.1.2	Eigenschaften der Klassen	154

2.1.3	Modellierung des Klassendiagramm mithilfe der Operationen	156
2.1.4	Praxis-Beispiel: Anwendung der Klassendiagramme in der Modellierung des Durchlassverhaltens des Transistors	158
2.2	Kompositionsstrukturdiagramm von Obeo-Designer-UML	170
2.3	Zustandsdiagramm von Obeo-UML-Designer	180
2.3.1	Überblick über Erstellungstools des Zustandsdiagramms	181
2.3.2	Notationselemente	181
2.3.3	Anwendung des Zustandsdiagramms in der Energietechnik	182
2.4	Komponentendiagramm	183
2.4.1	Komponentenmodell von Java EE	184
2.4.2	Komponenten für Java EE	185
2.4.3	Komponenten für JSF, JPA und CDI	186
2.4.3.1	JavaServer Faces (JSF)	186
2.4.3.2	Java-Persistence-API (JPA).	187
2.4.3.3	Contexts and Dependency Injection (CDI).	187
2.5	Verteilungsdiagramm (Deployment-Diagramm)	188
2.5.1	Device für „Application Server Java EE“	189
2.5.2	Device „Client“	190
2.5.3	Device MySQL-Datenbankserver	190
2.6	Zusammenfassung	190
	Literatur.	192
3	Eclipse-Papyrus-Framework	193
3.1	Erstellung eines UML-Klassendiagramms	193
3.1.1	Struktur des UML-Klassendiagramms.	198
3.1.2	Beispiel: Inneres Klassendiagramm.	202
3.1.2.1	Überblick über Assoziationen	205
3.1.2.2	Überblick über Generalisierung	206
3.1.2.3	„Vererbungskaskade“	206
3.2	Paketdiagramm.	207
3.2.1	Paketdiagramm mit dem Design Pattern Model View Controller	208
3.2.2	Überblick über Java-Code in dem Modell	209
3.3	„Class Tree Table“	216
3.3.1	Struktur der Tabelle	217
3.3.2	Vertikale Position	220
3.3.3	Horizontale Position.	222
3.4	Sequenzdiagramme mit Eclipse-Papyrus	226
3.5	Kommunikationsdiagramm mit Eclipse-Papyrus	234

3.6	Objektdiagramme mit Eclipse-Papyrus	239
3.6.1	Erstellen eines Klassendiagrammes mit Eclipse-Papyrus	240
3.6.2	Erstellen eines Objektdiagramms mit Eclipse-Papyrus	253
3.6.2.1	Elemente des Objektdiagramms	266
3.6.2.2	Grafische Darstellung vom Objektdiagramm	266
3.7	Kompositionsstrukturdiagramm.	267
3.7.1	Komposition.	268
3.7.2	Klassifikator	269
3.8	Komponentendiagramm mit Eclipse-Papyrus	281
3.8.1	Praxisbeispiel: Abhängigkeit zwischen Komponenten und Interface	281
3.8.2	Kapselung von Zustand und Verhalten.	283
3.9	Zusammenfassung	291
	Literatur.	296
4	SysML-Modellierung mit Eclipse-Papyrus	297
4.1	Blockdefinitionsdiagramm.	299
4.1.1	Aufbau von Blockdefinitionsdiagrammen	299
4.1.2	Erstellung von Blockdefinitionsdiagrammen (BDD) mit Eclipse-Papyrus	299
4.1.2.1	Praxisbeispiel: Modellierung der Schaltung vom Schwingkreiswechselrichter	301
4.1.2.2	Überblicke über Merkmale der Blöcke zur SysML-Modellierung	309
4.2	Internes Blockdiagramm (IBD)	317
4.2.1	Modellierung der Schaltung eines Blindleistungsstromrichters mit IBD	318
4.2.1.1	Modellierung der Funktionalität des Blindleistungsstromrichters mithilfe von SysML-Informationsobjektflüssen	318
4.2.1.2	Modellierung der Schaltung des Blindleistungsstromrichters mithilfe der SysML-Objektflussports	322
4.2.2	Modellierung der Leistungssteuerung durch die Spannungsverstellung bei Schwingkreiswechselrichtern mit IBD	326
4.3	Anforderungsdiagramm	332
4.3.1	Anforderungsdiagramm des Schwingkreiswechselrichters mit Papyrus	333
4.3.2	Anforderungstabelle des Solar-Schwingkreiswechselrichters mit Papyrus	339

4.4	Zusicherungsdiagramm (Parametrisierdiagramm)	347
4.4.1	Modellierung von Verlusten in „Insulate Gate Bipolar Transistor“ (IGBT) mithilfe von Sicherungsdiagrammen auf Basis von Eclipse-Papyrus-SysML	348
4.4.2	Modellierung von Blindleistungen mit Zusicherungsdiagrammen	351
4.5	Zusammenfassung	352
	Literatur.	355
5	Parallele Modellierung mit Obeo-UML-Designer	357
5.1	Modellierung mit Zustandsdiagrammen	357
5.1.1	Horizontale Modellierung	358
5.1.2	Vertikale Modellierung	359
5.2	Modellierung mit Aktivitätsdiagrammen	361
5.2.1	Vertikale Integration.	368
5.2.2	Horizontale Integration	370
5.3	Modellierung mit Klassendiagrammen	372
5.3.1	Vererbungshierarchie	372
5.3.2	Modellieren der Strukturen der Klassen	373
5.3.3	Parallelisierung der objektorientierten Modellierung	374
5.4	Modellierung mit Sequenzdiagrammen	375
5.4.1	Darstellung der Parallelisierungsprozesse mit Objekten oder Lebenslinien	377
5.4.2	Darstellung der Parallelisierungsprozesse mit Interaktionen.	378
5.5	Zusammenfassung	380
	Literatur.	382
6	Vom Modellieren zum Programmieren.	383
6.1	Anwendung von Java Swing in der Entwicklung der grafischen Oberfläche [1]	383
6.2	DesignPattern Interface	413
6.3	Codegenerierung aus Ecore-Modellen.	418
6.4	Zusammenfassung	424
	Literatur.	425
	Stichwortverzeichnis.	427