

# Contents

	<b>Foreword</b> .....	<b>xi</b>
	<b>Acknowledgments</b> .....	<b>xv</b>
<b>1</b>	<b>Introduction</b> .....	<b>1</b>
	1.1 Purpose and Use of Models .....	1
	1.2 Modeling for Software Development .....	2
	1.3 How to Read this Book .....	3
<b>2</b>	<b>MDSE Principles</b> .....	<b>7</b>
	2.1 MDSE Basics .....	7
	2.2 Lost in Acronyms: The MD* Jungle .....	9
	2.3 Overview of the MDSE Methodology .....	9
	2.3.1 Overall Vision .....	10
	2.3.2 Target of MDSE: Domains, Platforms, Technical Spaces, and Scenarios .	11
	2.3.3 Modeling Languages .....	13
	2.3.4 Metamodeling .....	13
	2.3.5 Transformations .....	16
	2.3.6 Model Classification .....	17
	2.4 MDSE Adoption in Industry .....	18
	2.5 Tool Support .....	20
	2.5.1 Drawing Tools vs. Modeling Tools .....	20
	2.5.2 Model-Based vs. Programming-Based MDSE Tools .....	22
	2.5.3 Eclipse and EMF .....	22
	2.6 Criticisms of MDSE .....	23
<b>3</b>	<b>MDSE Use Cases</b> .....	<b>25</b>
	3.1 Automating Software Development .....	26
	3.1.1 Code Generation .....	28
	3.1.2 Model Interpretation .....	31
	3.1.3 Combining Code Generation and Model Interpretation .....	32
	3.2 System Interoperability .....	33
	3.3 Reverse Engineering .....	36

<b>4</b>	<b>Model-Driven Architecture (MDA)</b> .....	<b>39</b>
4.1	MDA Definitions and Assumptions .....	40
4.2	The Modeling Levels: CIM, PIM, PSM .....	40
4.3	Mappings .....	43
4.4	General-Purpose and Domain-Specific Languages in MDA .....	44
4.5	Architecture-Driven Modernization .....	45
<b>5</b>	<b>Integration of MDSE in your Development Process</b> .....	<b>49</b>
5.1	Introducing MDSE in your Software Development Process .....	49
5.1.1	Pains and Gains of Software Modeling .....	50
5.1.2	Socio-Technical Congruence of the Development Process .....	50
5.2	Traditional Development Processes and MDSE .....	51
5.3	Agile and MDSE .....	51
5.4	Domain-Driven Design and MDSE .....	53
5.5	Test-Driven Development and MDSE .....	54
5.5.1	Model-Driven Testing .....	54
5.5.2	Test-Driven Modeling .....	55
<b>6</b>	<b>Modeling Languages at a Glance</b> .....	<b>57</b>
6.1	Anatomy of Modeling Languages .....	57
6.2	General-Purpose vs. Domain-Specific Modeling Languages .....	59
6.3	General-Purpose Modeling: The Case of UML .....	60
6.3.1	Design Practices .....	62
6.3.2	Structure Diagrams (or Static Diagrams) .....	62
6.3.3	Behavior Diagrams (or Dynamic Diagrams) .....	64
6.3.4	UML Tools .....	66
6.3.5	Criticisms and Evolution of UML .....	67
6.4	UML Extensibility: The Middle Way Between GPL and DSL .....	67
6.4.1	Stereotypes .....	68
6.4.2	Predicates .....	68
6.4.3	Tagged Values .....	68
6.4.4	UML Profiling .....	69
6.5	Overview on DSLs .....	70
6.5.1	Principles of DSLs .....	70
6.5.2	Some Examples of DSLs .....	72
6.6	Defining Modeling Constraints (OCL) .....	72

<b>7</b>	<b>Developing your Own Modeling Language</b> .....	<b>77</b>
7.1	Metamodel-Centric Language Design .....	77
7.1.1	Abstract Syntax .....	77
7.1.2	Concrete Syntax .....	78
7.1.3	Language Ingredients at a Glance .....	79
7.2	Example DSML: sWML .....	80
7.3	Abstract Syntax Development .....	81
7.3.1	Metamodel Development Process .....	83
7.3.2	Metamodeling in Eclipse .....	92
7.4	Concrete Syntax Development .....	94
7.4.1	Graphical Concrete Syntax (GCS) .....	95
7.4.2	Textual Concrete Syntax (TCS) .....	100
<b>8</b>	<b>Model-to-Model Transformations</b> .....	<b>107</b>
8.1	Model Transformations and their Classification .....	107
8.2	Exogenous, Out-Place Transformations .....	109
8.3	Endogenous, In-Place Transformations .....	116
8.4	Mastering Model Transformations .....	121
8.4.1	Divide and Conquer: Model Transformation Chains .....	121
8.4.2	HOT: Everything is a Model, Even Transformations! .....	122
8.4.3	Beyond Batch: Incremental and Lazy Transformations .....	122
8.4.4	Bi-Directional Model Transformations .....	123
<b>9</b>	<b>Model-to-Text Transformations</b> .....	<b>125</b>
9.1	Basics of Model-Driven Code Generation .....	125
9.2	Code Generation Through Programming Languages .....	127
9.3	Code Generation Through M2T Transformation Languages .....	131
9.3.1	Benefits of M2T Transformation Languages .....	131
9.3.2	Template-Based Transformation Languages: an Overview .....	132
9.3.3	Acceleo: An Implementation of the M2T Transformation Standard ....	133
9.4	Mastering Code Generation .....	136
9.5	Excursus: Code Generation Through M2M Transformations and TCS .....	138
<b>10</b>	<b>Managing Models</b> .....	<b>141</b>
10.1	Model Interchange .....	141
10.2	Model Persistence .....	144

10.3	Model Comparison .....	145
10.4	Model Versioning .....	146
10.5	Model Co-Evolution .....	148
10.6	Global Model Management .....	150
10.7	Model Quality .....	152
	10.7.1 Verifying Models .....	153
	10.7.2 Testing and Validating Models .....	154
10.8	Collaborative Modeling .....	154
<b>11</b>	<b>Summary .....</b>	<b>157</b>
	<b>Bibliography .....</b>	<b>159</b>
	<b>Authors' Biographies .....</b>	<b>165</b>