Table of Contents

Chapter Summary xi

Chapter 1. Model Transformation: A Survey of the State of the Art 1
Tom MENS

1.1. Model-driven engineering 1
1.2. Model transformation 2
1.2.1. Definitions 2
1.2.2. Taxonomy 4
1.3. Model transformation languages 5
1.4. Model transformation activities 8
1.5. Conclusion 14
1.6. Acknowledgements 14
1.7. Bibliography 15

Chapter 2. Model-Based Code Generation 21
Chris RAISTRICK

2.1. Introduction 21
2.2. The model-driven architecture (MDA) process . . 22
2.3. The automated approach to code generation 23
2.4. Domain modeling 25
2.5. The executable UML (xUML) formalism 29
2.6. System generation 31

2.7. Executable UML to code mappings 34
2.8. Conclusions 41
2.9. Bibliography. 42

Chapter 3. Testing Model Transformations: A Case for Test Generation from Input Domain Models 43
Benoit BAUDRY

3.1. Introduction 43
3.2. Challenges for testing systems with large input domains 46
3.2.1. Large set of input data 46
3.2.2. Configurable systems 48
3.2.3. Grammarware and model transformations . . 48
3.2.4. Testing challenges 52
3.3. Selecting test data in large domains 52
3.3.1. Category partition 52
3.3.2. Combinatorial interaction testing 55
3.4. Metamodel-based test input generation. 58
3.4.1. Metamodel coverage criteria 59
3.4.2. Model and object fragments for test adequacy criteria 61
3.4.3. Discussion. 64
3.4.4. Automatic synthesis of test models 65
3.5. Conclusion 67
3.6. Acknowledgements 68
3.7. Bibliography. 68

Chapter 4. Symbolic Execution-Based Techniques for Conformance Testing 73
Christophe GASTON, Pascale LE GALL, Nicolas RAPIN and Assia TOUIL

4.1. Context 73
4.1.1. Conformance testing: an introduction 73
4.1.2. Conformance relation 74
4.1.3. An overview of the approach 78

4.2. Input output symbolic transition systems 79
4.2.1. Data types 79
4.2.2. Input/output symbolic transition systems . . . 80
4.2.3. Semantics 82
4.3. Symbolic execution 84
4.4. Conformance testing for IOSTS 87
4.4.1. Test purposes 88
4.4.2. Preliminary definitions and informal description 89
4.4.3. Inference rules 94
4.5. Concluding remarks 96
4.5.1. Choosing test purposes 96
4.5.2. Implementation issues 101
4.6. Bibliography 101

Chapter 5. Using MARTE and SysML for Modeling Real-Time Embedded Systems 105
Huascar ESPINOZA, Daniela CANCILA, Sébastien GÉRARD and Bran SELIC

5.1. Introduction 105
5.2. Background 108
5.2.1. UML profiling capabilities 108
5.2.2. SysML and MARTE modeling capabilities . . 111
5.3. Scenarios of combined usage 113
5.3.1. Defining architecture frameworks 114
5.3.2. Requirements engineering 115
5.3.3. System-level design integration 117
5.3.4. Engineering/quantitative analysis 120
5.4. Combination Strategies 125
5.4.1. Issues 125
5.4.2. Strategies 128
5.5. Related work 130
5.6. Conclusion 133
5.7. Acknowledgements 134
5.8. Bibliography 134

Chapter 6. Software Model-based Performance Analysis 139
Dorina C. PETRIU

6.1. Introduction 139
6.2. Performance models 142
6.2.1. Queuing network models 144
6.2.2. Layered queuing network model 146
6.3. Software model with performance annotations. . 148
6.3.1. Performance domain model. 148
6.3.2. Source model example 152
6.4. Mapping from software to performance model . . 155
6.5. Using a pivot language: Core Scenario Model (CSM) 158
6.6. Case study performance model. 160
6.7. Conclusions 162
6.8. Acknowledgements 163
6.9. Bibliography. 163

Chapter 7. Model Integration for Formal Qualification of Timing-Aware Software Data Acquisition Components 167
Jean-Philippe BABAU, Philippe DHAUSSY and Pierre-Yves PILLAIN

7.1. Introduction 167
7.2. System modeling. 170
7.2.1. Acquisition system modeling. 170
7.2.2. Case study 172
7.2.3. Formal modeling techniques 174
7.3. Variation points modeling 182
7.3.1. Variation points definition 184
7.3.2. CDL implementation 187
7.4. Experiments and results 189
7.4.1. Tools. 189
7.4.2. Experimentations 191

7.5. Conclusion 194
7.6. Bibliography 195

Chapter 8. SoC/SoPC Development using MDD and MARTE Profile 201
Denis AULAGNIER, Ali KOUDRI, Stéphane LECOMTE, Philippe SOULARD, Joël CHAMPEAU, Jorgiano VIDAL, Gilles PERROUIN and Pierre LERAY

8.1. Introduction 201
8.2. Related works 203
8.3. MOPCOM process and models 206
8.4. Application 210
8.5. System analysis 211
8.5.1. Requirement analysis 211
8.5.2. Functional analysis 212
8.5.3. Action language 213
8.6. Abstract modeling level 214
8.7. Execution modeling level 216
8.7.1. The platform independent model/application model in EML. 217
8.7.2. The platform model in EML 217
8.7.3. The platform specific model/allocation model in EML 218
8.7.4. Analysis model. 219
8.8. Detailed modeling level 220
8.8.1. Platform model 221
8.8.2. Allocation model. 222
8.9. Tooling Support 223
8.9.1. Process validation through metamodeling with Kermeta 223
8.9.2. Model transformation and generation with MDWorkbench platform 224
8.10. HDL Code Generation 225
8.10.1. VHDL code generation 226
8.10.2. Rhapsody integration 227
8.11. Conclusion 228

8.12. Acknowledgements 229
8.13. Bibliography 229

List of Authors 233

Index 237