

CONTENTS

Introduction	11
Jean-Philippe Babau, Joël Champeau and Sébastien Gérard	
Chapter 1. On Metamodels and Language Engineering	13
Pierre-Alain Muller	
1.1. Introduction	13
1.2. Modeling abstract syntax	14
1.3. Modeling operational semantics	16
1.4. Modeling concrete syntax	18
1.5. Related works	21
1.6. Conclusion	21
1.7. References	22
Chapter 2. Using Directives to Implement Model Transformations	25
Devon Simmonds, Robert France and Sudipto Gosh	
2.1. Introduction	25
2.2. Model Transformation Using Embedded Directives	26
2.3. Transformations directives	27
2.3.1. The source and rename Directives	27
2.3.2. The redefine Directive	29
2.3.3. The new and exclude Directives	30
2.4. Transformation schemas	31
2.5. Class Model transformation – Illustration Example	32
2.5.1. Server Distribution Aspect Class Model	32
2.5.2. COBRA Distribution Class Diagram Transformation Schema	33

6	From MDD concepts to experiments and illustrations	
	2.5.3. Processing Transformation Directives	35
	2.6. Discussion and Conclusion	37
	2.6.1. Model Transformation Using QVT	37
	2.7. References.	41
	Chapter 3. Rationale of the UML profile for Marte	43
	Sébastien Gérard and Huascar Espinoza	
	3.1. Introduction.	43
	3.2. Outlines of Marte	45
	3.2.1. Marte and other OMG standards related RT/E	45
	3.2.2. A foundation for model driven techniques	46
	3.2.3. How should the specification be used?	47
	3.3. Profile architecture	51
	3.4. References.	52
	Chapter 4. From UML to Performance Analysis Models by Abstraction-raising Transformation	53
	Dorina Petriu and Antonino Sabetta	
	4.1. Introduction.	53
	4.2. Conceptual Approach for Abstracting-raising Transformation	55
	4.3. Two-step abstraction-raising transformation	57
	4.3.1. Description of the Source Model	57
	4.3.2. Description of the Target Model	58
	4.3.3. Mapping Approach.	59
	4.4. Two-step abstraction-raising transformation	59
	4.4.1. Proposed Approach	59
	4.4.2. Graph Grammar used for Abstraction Raising.	61
	4.4.3. Mapping from the Extended Source Model to LQN.	63
	4.5. Application of the proposed transformation	64
	4.5.1. Parsing	64
	4.5.2. Generating the LQN relational mapping.	66
	4.6. Conclusion	68
	4.7. References.	69
	Chapter 5. Component-Based Software Engineering for Embedded Systems.	71
	Ivica Crnkovic	
	5.1. Embedded Systems	71
	5.2. Specific requirement and aspects of Embedded Systems	72

5.3. Component-based Basic Concepts for Embedded Systems.	74
5.4. Specific Demands on Component-based Software Engineering	75
5.4.1. Component Interface	76
5.4.2. Component deployment and composition	76
5.5. State of the CBSE practice and experience for Embedded Systems.	77
5.5.1. Automotive Industry.	78
5.5.2. Industrial Automation.	81
5.5.3. Consumer Electronics.	82
5.5.4. Other domains	84
5.6. Work on standardization.	84
5.6.1. The Unified Modelling Language (UML).	84
5.6.2. Real-time CORBA	86
5.6.3. Programmable Logic Controllers: the IEC 61131-3 standard	86
5.6.4. Other standards and de-facto standards.	87
5.7. The needs and priorities in research	88
5.8. References.	89
Chapter 6. Model Driven Engineering for System-on-Chip Design.	91
Pierre Boulet, Cédric Dumoulin and Antoine Honoré	
6.1. Introduction.	91
6.2. SoC Design Challenges and Model Driven Engineering	92
6.2.1. Cost	92
6.2.2. Silicon complexity	93
6.2.3. Productivity.	93
6.2.4. Model Driven Engineering Assets	95
6.3. UML Profiles for SoC Design.	95
6.3.1. Embedded System Modeling and Analysis	95
6.3.2. Electronic System Level Modeling	96
6.4. MDE Approach to SoC Co-Modeling	97
6.4.1. Multiple Models in SoC	98
6.4.2. Metamodels for the “Y” Design.	98
6.4.3. From High Level Models	99
6.4.4. To Technology Models.	100
6.5. Gaspard2 Development Environment.	102
6.5.1. Simplify the work with good tools	103
6.5.2. Transformation Engine: ModTransf.	103
6.5.3. From UML2 Modelers to the Gaspard2 Environment	104
6.5.4. Model Refactoring and Deployment Metmodel.	105
6.5.5. Example of Concept Transformation	106
6.5.6. Evolution of our environment	107
6.6. Conclusion	107
6.7. References.	108

8 From MDD concepts to experiments and illustrations

Chapter 7. Schedulability Analysis and MDD	111
Samuel Rouxel, Guy Gogniat, Jean-Philippe Diguët, Jean-Luc Philippe and Christophe Moy	
7.1. Introduction	111
7.2. Related Work	113
7.3. Global Approach	114
7.3.1. Application Specification (1 st step)	114
7.3.2. Platform Specification (2 nd step)	116
7.3.3. Application – Platform Mapping (3 rd step)	116
7.3.3. Analysis results (4 th step)	117
7.4. UML Modeling	118
7.4.1. Attributes identification	118
7.4.2. Analysis details	120
7.5. Real time analysis tool (RTDT)	121
7.5.1. Real time scheduling strategy	121
7.5.2. Design space exploration for HW/SW partitioning	122
7.6. UMTS FDD Case Study	126
7.7. Conclusion	128
7.8. Acknowledgements	129
7.9. References	129
Chapter 8. Model Driven Testing of Time Sensitive Distributed Systems . .	131
Borislav Gajanovic, Hans Grönniger and Bernhard Rumpe	
8.1. Model Driven Testing	131
8.2. Asynchronous Communication in Distributed Systems	133
8.3. The Alternative Bit Protocol	135
8.3.1. Informal Description of the ABP Components	135
8.3.2. Stream-Based Specification	137
8.3.3. A Mapping to Haskell	139
8.3.4. Executing the Model	141
8.4. Strategies for Testing Distributed, Asynchronously Communicating Systems . .	141
8.4.1. Rules for Testing of Distributed Functionally Specified Models . .	142
8.5. Implementing Tests in Haskell	144
8.5.1. Test Infrastructure	144
8.5.2. Tests for the ABP Components	145
8.6. Discussion of Results	146
8.7. References	147

Chapter 9. Model Management for Formal Validation	149
Joël Champeau, Philippe Dhaussy, François Mekerke and Jean Charles Roger	
9.1. Introduction	149
9.2. System modeling framework	151
9.2.1. Separation of concerns	151
9.2.2. Domain modeling	151
9.2.3. Model Management	152
9.2.4. MDD Implementation	153
9.2.5. System modeling framework conclusion	161
9.3. Building models for formal verification	162
9.3.1. Functionalities of the environment under development	163
9.3.2. Observer and context-based model checking	164
9.3.3. Verification contexts	164
9.3.4. Model transformation techniques	165
9.3.5. A language to specific contexts	165
9.3.6. Translation of CxUCC to observers and concrete contexts	168
9.3.7. Translation of CxUCC to an a-context and an observer set	168
9.3.8. IF-2 implementation	171
9.4. Conclusion and future work	172
9.5. References	173
Chapter 10. The Design of Space Systems	175
David Chemouil	
10.1. Introduction	175
10.1.1. Context	175
10.1.2. Outline	176
10.1.3. Notice	176
10.2. Space Systems	177
10.2.1. Applications	177
10.2.2. Two Views on the Architecture of Space Systems	177
10.3. Design	182
10.3.1. Process	182
10.3.2. By the way, what is so special about Space Systems?	186
10.3.3. On-Board Software	188
10.4. Modelling	190
10.4.1. Current Possibilities	190
10.4.2. Trends and Projects	190
10.5. Conclusion	192
10.6. References	193

10 From MDD concepts to experiments and illustrations

Chapter 11. TOPCASED – An Open Source Development Environment for Embedded Systems.	195
Patrick Farail, Pierre Gauffillet, Agusti Canals, Christophe Le Camus, David Sciamma, Pierre Michel, Xavier Crégul and Marc Pantel	
11.1. Introduction	195
11.2. Requirements and TOPCASED Architecture	198
11.3. Model Driven Engineering and meta-modeling	200
11.4. Generating model editors	201
11.5. Acknowledgment	204
11.6. References	205
11.7. Glossary	206
Chapter 12. Facing Industrial Challenges: A Return on an Experiment on Model-driven Engineering.	209
Jean-Luc Voirin	
12.1. Introduction	209
12.2. A quick overview of our understanding of MDE	211
12.3. Expected Benefits of Model-driven Engineering	212
12.4. Applying MDE Concepts in an industrial Context	214
12.5. Return of Experiment and Findings on MDE Use	218
12.6. Conclusion: so what about MDE?	222
Index of Authors	223