Table of Contents

Chapter 1. Metamodeling in Software Architectures 1
Adel SMEDA and Mourad Chabane OUSSALAH

1.1. Introduction. 1
1.2. Metamodeling, why? 3
1.3. Software architecture metamodeling 3
1.4. MADL: a meta-architecture description language 5
1.4.1. Four levels of modeling in software architectures. 5
1.4.2. MADL: reflexive core dedicated to the meta-meta-architecture 7
1.4.3. MADL structure 8
1.4.4. MADL instantiation: example of the ADL Acme 11
1.4.5. Comparison of MADL and MDA/MOF 13
1.5. Mapping of ADLs to UML 17
1.5.1. Why to map an ADL to UML? 18
1.5.2. ADL mapping to UML 19
1.6. A mapping example: the case of the Acme language 31
1.7. Some remarks on the mapping of ADL concepts to UML 32
1.7.1. UML 2.0 as an ADL 32
1.7.2. Mapping strategies 33
1.8. Conclusion 34
1.9. Bibliography 34

Chapter 2. Architecture Constraints 37
Chouki TIBERMACINE

2.1. Introduction. 38
2.2. State of the art 40
2.2.1. Expression of architecture constraints in the design phase. 40

2.2.2. Expression of architecture constraints in the implementation phase 49
2.3. Architecture constraints on object-oriented applications 57
2.3.1. Architecture constraints in the design phase 57
2.3.2. Architecture constraints in the implementation phase 61
2.4. Architecture constraints on component-based applications 68
2.4.1. Architecture constraints in the design phase 69
2.4.2. Architecture constraints in the implementation phase 75
2.5. Architecture constraints on service-oriented applications 79
2.6. Conclusion 85
2.7. Bibliography 86

Chapter 3. Software Architectures and Multiple Variability 91
Mathieu ACHER, Philippe COLLET and Philippe LAHIRE

3.1. Introduction 91
3.2. Variability: foundations and principles 95
3.2.1. Variability and product lines 95
3.2.2. Feature models 97
3.3. Framework of studies and connected work 99
3.3.1. From multiplicity to variability 100
3.3.2. Extraction and evolution of architectural variability 101
3.4. Video surveillance component architecture 102
3.4.1. Case study 102
3.4.2. Accounting for multiple variability 104
3.4.3. Results 108
3.5. SOA for scientific workflows 110
3.5.1. Case study 110
3.5.2. Accounting for multiple variability 112
3.5.3. Results 114
3.6. Reverse engineering plugin-based architecture 116
3.6.1. Case study 116
3.6.2. Accounting for multiple variability 118
3.6.3. Results 120
3.7. Evaluation 122
3.7.1. The necessity of tooling 122
3.7.2. Summary of case studies 123
3.8. Conclusion 125
3.9. Bibliography 126

Chapter 4. Architecture and Quality of Software Systems 133
Nicole LÉVY, Francisca LOSAVIO and Yann POLLET

4.1. Introduction 133

4.2. Quality approach. 135
4.2.1. ISO 25010 quality 135
4.2.2. Quality reference 137
4.2.3. Quality model of a system. 138
4.2.4. Functional quality model 139
4.2.5. Quality model of the architecture 140
4.3. Approach for architecture development of a domain 142
4.3.1. General principles 142
4.3.2. Functional quality model 145
4.3.3. Architectural quality model 145
4.3.4. Reference architecture 145
4.3.5. Transition from domain level to system level 147
4.4. Development of the reference architecture in a functional domain . . . 148
4.4.1. Example of functional domain 148
4.4.2. Functional refinement 148
4.4.3. Development of the FQM 150
4.4.4. Definition of the preliminary architecture 151
4.4.5. Development of architectural quality model 152
4.4.6. Integration of the reference architecture of the domain. 152
4.5. Architectures at system level 156
4.5.1. Functional refinement 156
4.5.2. Functional quality model 157
4.5.3. Basic architecture. 158
4.5.4. Architectural quality model 158
4.5.5. Architecture of the Dopamine and Samarkand systems 159
4.6. Related work 161
4.7. Conclusion 166
4.8. Bibliography 167

Chapter 5. Software Architectures and Multiagent Systems 171
Jean-Paul ARCANGELI, Victor NOËL and Frédéric MIGEON

5.1. Introduction. 172
5.2. MAS and agent-oriented software engineering 172
5.2.1. Agent 173
5.2.2. System and interactions 174
5.2.3. MAS 175
5.2.4. Examples of MAS 177
5.2.5. Agent-oriented software engineering 178
5.3. MAS as an architectural style 183
5.3.1. Positioning the “MAS” style 183
5.3.2. Characteristics in terms of abstraction. 184
5.3.3. Characteristics in terms of (de)composition 188

5.3.4. Link with the requirements 190
5.3.5. A family of architectural styles 194
5.4. The architectural gap 195
5.4.1. State of the practice 196
5.4.2. Analysis from an architectural point of view 197
5.4.3. Assessment 200
5.5. How to fill the architectural gap 200
5.5.1. Limitations of existing solutions 200
5.5.2. Realization of the microarchitecture 201
5.6. Conclusion 204
5.7. Bibliography 205

Chapter 6. Software Architectures and Software Processes 209
Fadila AOUSSAT, Mourad Chabane OUSSALAH and
Mohamed AHMED-NACER

6.1. Introduction 209
6.2. Software process architectures 211
6.2.1. Software process models: definition 211
6.2.2. Modeling software architecture-based software processes 213
6.3. Comparison framework for SA-based SP model reuse solutions 214
6.3.1. The software process axis evaluation criteria 217
6.3.2. The software architecture axis evaluation criteria 220
6.3.3. The quality axis evaluation criteria 223
6.4. Evaluation of SA-based SP modeling and execution approaches 225
6.4.1. SP axis evaluation of SA-based SP reuse approaches 225
6.4.2. SA axis evaluation of SA-based SP reuse approaches 229
6.4.3. Quality axis evaluation of SA-based SP reuse approaches 232
6.4.4. Assessment and discussions 234
6.5. Conclusion 235
6.6. Bibliography 236

List of Authors 241

Index 243