

Table of Contents

Chapter 1. Introduction	1
1.1. Software product line engineering	2
1.2. Model-driven engineering	3
1.3. Merging model-driven and software product line engineering	5
1.4. The FieSta framework	8
1.5. Book structure	11
Chapter 2. Software Product Line Engineering Basics	17
2.1. Introduction to product line engineering	17
2.2. Brief history	21
2.3. Application example: Smart-Home systems	24
2.3.1. Smart-Home system's domain	24
2.3.2. Requirements of the application example	26
2.4. Software product line engineering	30
2.5. Domain engineering	34
2.5.1. Component-based software engineering	36
2.6. Variability management	37
2.6.1. Feature modeling	40
2.7. Application engineering	43
2.7.1. Product configuration	44
2.7.2. Product derivation	46
2.8. Benefits and drawbacks	48
2.9. Issues in product line	49

2.9.1. Variability management	50
2.9.2. Product derivation	50
2.9.3. Testing	51
2.9.4. Traceability	52
2.9.5. Product line evolution	53
2.9.6. Tool support	55
2.10. Summary	56
Chapter 3. Model-Driven Engineering	59
3.1. Introduction	59
3.2. Models and metamodels	60
3.2.1. The 4-level metamodeling framework	65
3.2.2. The nature of models	67
3.3. UML class diagrams and OCL	68
3.4. Model transformations	74
3.4.1. Scheduling of transformation rules	76
3.4.2. Model transformation patterns	78
3.4.3. Classification of model transformations	79
3.4.4. Vertical model transformations	80
3.4.5. Horizontal model transformations	81
3.4.6. Model composition or model weaving	81
3.5. Modeling framework	83
3.5.1. The eclipse modeling framework	83
3.5.2. The topcased toolkit	86
3.6. Model transformation languages	86
3.6.1. QVT	87
3.6.2. ATL	89
3.6.3. The openArchitectureWare framework	90
3.6.4. The Xtend language	92
3.7. Benefits and challenges for SPLE	96
3.8. Summary	98
Chapter 4. Model-Driven and Software Product Line Engineering	101
4.1. Introduction	102
4.2. Problem space issues	107
4.2.1. Separating points of views	107
4.2.2. Capturing variability and configuring products	108

4.2.3. Relating several points of view	109
4.2.4. Configuring products in a multi-staged process	110
4.3. Solution space issues	111
4.4. Developing core assets	112
4.4.1. Developing decision models and deriving products	112
4.5. Variability expression and product configuration	113
4.5.1. Metamodels	114
4.5.2. Feature models	120
4.6. Core asset development and product derivation	126
4.6.1. Transformation rules in the Smart-Home systems SPL	127
4.6.2. Creating and using decision models	132
4.7. Summary	138
Chapter 5. The FieSta Framework: Fine-Grained Derivation and Configuration	139
5.1. Introduction	139
5.1.1. Coarse-grained and fine-grained variations	140
5.2. Binding models and constraint models	142
5.2.1. Binding models	142
5.2.2. Constraint models	143
5.2.3. The cardinality property	146
5.2.4. The structural dependency property	147
5.2.5. The constraint metamodel and the binding metamodel	148
5.2.6. Validating binding models against constraint models	150
5.3. Deriving products based on constraint models and binding models	152
5.3.1. The extended decision metamodel	155
5.3.2. Creating executable model transformation workflows from decision models and constraint models	156

5.4. Identified limitations	157
5.4.1. Features combinatorial	157
5.4.2. Features interaction	158
5.4.3. Bindings interaction	159
5.5. Summary	160
Chapter 6. Tools Support	161
6.1. Introduction	161
6.2. The FieSta process	162
6.3. The SPL of Smart-Home systems	163
6.4. Variability expression and product configuration	170
6.4.1. MD-SPL project creation	170
6.4.2. Metamodels and feature models creation . . .	170
6.4.3. Constraint models creation	173
6.4.4. Domain models and binding models creation	178
6.5. Completing and running the product derivation	184
6.5.1. Transformation rules creation	184
6.5.2. Decision models creation	186
6.5.3. Generation and execution of model transformation workflows	188
6.6. Summary	190
Chapter 7. A Second Comprehensive Application Example	191
7.1. Domain of the collection manager system	191
7.2. Requirements of the application example	192
7.2.1. Kernel commonalities	193
7.2.2. GUI commonalities	193
7.2.3. Kernel and GUI variability	193
7.3. The overall process	196
7.3.1. Domain engineering	196
7.3.2. Application engineering	197
7.4. Variability expression and product configuration	198
7.4.1. Metamodels	198
7.4.2. The feature model	202

7.4.3. The constraint model	204
7.4.4. Binding models	205
7.5. Core assets development and product derivation	207
7.5.1. Rule transformations in the SPL of the collection manager systems	207
7.5.2. Decision models	209
7.6. Summary	211
Chapter 8. Further Reading	213
8.1. Northop and Clements' book	213
8.2. Pohl, Böckle and Van der Linden's book	214
8.3. Gomaa's book	214
8.4. Van der Linden, Schmid, and Rommes' book	215
8.5. Stahl, Voelter, and Czarnecki book	216
8.6. AMPLE book	216
8.7. Feature modeling notations	218
8.8. Decision models	218
8.9. Model-driven software product lines	220
8.9.1. The Czarnecki and Antkiewicz's approach	222
8.9.2. The Wagelaar's approach	224
8.9.3. Loughran <i>et al.</i> 's approach	229
8.9.4. Voelter and Groher's approach	232
8.9.5. Comparison table	235
8.10. Dynamic variability	236
8.11. Domain specific languages	238
8.12. Additional references	240
8.13. Summary	242
Chapter 9. Conclusion	243
9.1. Book summary	244
9.2. MD-SPL engineering	247
9.2.1. Metamodeling and feature modeling	248
9.2.2. Multi-staged configuration of products	249
9.2.3. Coarse and fine-grained variations and configurations	249
9.2.4. Core assets development and decision models	250

9.2.5. Product derivation	251
9.2.6. Comparison table	251
9.2.7. Perspectives	253
Bibliography	257
Index	271