

Contents

Part I Software Product Lines

1	Software Product Lines	3
1.1	From Individualism to Standardization and Back Again	3
1.2	Specialized and Standardized Software	6
1.3	Software Product Lines	7
1.4	Promises of Software Product Lines	9
1.5	Success Stories	10
1.6	A Feature-Oriented Approach	11
1.7	Running Examples	11
1.8	Intended Audience of the Book	13
1.9	How to Read this Book	14
1.10	Further Reading	15
2	A Development Process for Feature-Oriented Product Lines	17
2.1	Features and Products	17
2.2	A Process for Product-Line Development	19
2.2.1	Domain Analysis	22
2.2.2	Requirements Analysis	24
2.2.3	Domain Implementation	25
2.2.4	Product Derivation	26
2.3	Feature Modeling	26
2.3.1	Feature Models	27
2.3.2	Feature Diagrams	28
2.3.3	Formalization in Propositional Logic	31
2.3.4	The Feature Model for the Graph Library	32
2.3.5	Variations and Extensions of Feature Models	34
2.3.6	Feature Modeling in Practice	36
2.3.7	Tooling	39

2.4	Adoption Paths of the Product-Line Approach	39
2.4.1	Proactive Approach.	40
2.4.2	Extractive Approach	40
2.4.3	Reactive Approach	41
2.5	Further Reading	42
	Exercises	42

Part II Variability Implementation

3	Basic Concepts, Classification and Quality Criteria	47
3.1	Dimensions of Variability Implementation.	48
3.1.1	Binding Time.	48
3.1.2	Technology: Language-Based Versus Tool-Based.	49
3.1.3	Representation: Annotation Versus Composition	50
3.2	Quality Criteria	52
3.2.1	Preplanning Effort	52
3.2.2	Feature Traceability	54
3.2.3	Separation of Concerns	55
3.2.4	Information Hiding.	56
3.2.5	Granularity	59
3.2.6	Uniformity.	59
3.3	Structure of Subsequent Chapters	60
3.4	Further Reading	61
	Exercises	62
4	Classic, Language-Based Variability Mechanisms	65
4.1	Parameters.	66
4.1.1	Discussion	66
4.2	Design Patterns	69
4.2.1	Observer Pattern.	70
4.2.2	Template-Method Pattern	71
4.2.3	Strategy Pattern	73
4.2.4	Decorator Pattern	75
4.2.5	Discussion	77
4.3	Frameworks	79
4.3.1	White-Box Frameworks.	80
4.3.2	Black-Box Frameworks.	81
4.3.3	An Implementation Example for Frameworks	82
4.3.4	Loading Plug-Ins	85
4.3.5	Discussion	86
4.4	Components and Services	89
4.4.1	Sizing Components.	91
4.4.2	Composing Components	92

4.4.3	Components Versus Plug-Ins	93
4.5	Further Reading	95
	Exercises	96
5	Classic, Tool-Driven Variability Mechanisms	99
5.1	Version-Control Systems	99
5.1.1	Terminology	100
5.1.2	Building Product Lines with Version-Control Systems.	101
5.1.3	Discussion	103
5.2	Build Systems	105
5.2.1	Variability in Build Scripts	105
5.2.2	Custom Build Scripts	106
5.2.3	Case Study: Build-System Variability in Linux	107
5.2.4	Discussion	108
5.3	Preprocessors	110
5.3.1	The C Preprocessor <code>cpp</code>	110
5.3.2	Implementing Variability with Preprocessors	111
5.3.3	Further Preprocessors	113
5.3.4	Disciplined Annotations	116
5.3.5	Preprocessors in Practice	118
5.3.6	Discussion	120
5.4	Further Reading	124
	Exercises	125
6	Advanced, Language-Based Variability Mechanisms	129
6.1	Feature-Oriented Programming.	130
6.1.1	Collaboration-Based Design.	130
6.1.2	Feature Modules.	132
6.1.3	The Jak Language	133
6.1.4	Models of Feature-Oriented Programming	135
6.1.5	Discussion	138
6.2	Aspect-Oriented Programming	141
6.2.1	Aspects: Separating Crosscutting Concerns	142
6.2.2	The AspectJ Language	145
6.2.3	Aspects for Product Lines	147
6.2.4	Discussion	149
6.3	Aspects and Feature Modules in Concert.	152
6.3.1	Homogeneous and Heterogeneous Crosscutting Concerns	153
6.3.2	Static and Dynamic Crosscutting Concerns	157
6.3.3	Summary of Comparison.	161
6.3.4	Combining Aspects and Feature Modules	161

6.3.5	A Study on Advanced Crosscutting Mechanisms	163
6.3.6	Discussion	164
6.4	Tooling	166
6.5	Practical Relevance.	167
6.6	Further Approaches.	168
6.6.1	Delta-Oriented Programming	168
6.6.2	Refactoring Feature Modules	169
6.6.3	Context-Oriented Programming	170
6.7	Further Reading	171
	Exercises	172
7	Advanced, Tool-Driven Variability Mechanisms	175
7.1	Exploiting Feature Tracing	175
7.1.1	Consistency Checking	177
7.1.2	Visualizing Tracing Information.	178
7.2	Views on Code.	180
7.3	Integrated Product Derivation.	182
7.4	Discussion: Virtual Separation of Concerns	184
7.5	Tooling	186
7.6	Further Reading	186
	Exercises	187
 Part III Advanced Topics		
8	Refactoring of Software Product Lines	193
8.1	Refactoring in General	194
8.2	Refactoring in Software Product Lines	197
8.2.1	Variability Smells in Software Product Lines.	197
8.2.2	Defining Product-Line Refactorings	200
8.2.3	Examples of Product-Line Refactorings.	201
8.3	Refactoring as Path Toward a Product Line.	203
8.3.1	Example: Extraction of Feature Colored of the Graph Library.	203
8.3.2	Case Study: Refactoring of Berkeley DB with AspectJ	207
8.4	Further Reading	210
	Exercises	212
9	Feature Interactions	213
9.1	The Feature-Interaction Problem.	214
9.1.1	Higher Order Interactions	216
9.2	Detecting Feature Interactions	217
9.3	The Optional-Feature Problem	219
9.4	Implementing Feature Interactions	222

9.4.1	Implementation Strategies: Overview and Goals.	223
9.4.2	Change Feature Model	224
9.4.3	Multiple Implementations	225
9.4.4	Moving Code.	226
9.4.5	Conditional Compilation	227
9.4.6	Optional Weaving.	228
9.4.7	Distinct Module for Coordination Code.	230
9.4.8	Comparison of Solutions	232
9.5	Experience.	233
9.5.1	Decomposition of Berkeley DB	234
9.5.2	Design and Implementation of FAME-DBMS	236
9.6	Further Reading	239
	Exercises	240
10	Analysis of Software Product Lines.	243
10.1	Analysis of Feature Models	244
10.1.1	Valid Feature Selection	245
10.1.2	Consistent Feature Models.	247
10.1.3	Testing Facts about Feature Models	248
10.1.4	Dead Features and Mandatory Features	249
10.1.5	Constraint Propagation	250
10.1.6	Number of Valid Feature Selections	251
10.1.7	Comparing Feature Models	252
10.1.8	Other Feature-Model Analyses	254
10.2	Analysis of Feature-to-Code Mappings	254
10.2.1	Dead Code.	255
10.2.2	Abstract Features	257
10.2.3	Determining Presence Conditions	257
10.3	Analysis of Domain Implementations	260
10.3.1	Design Space	262
10.3.2	Sampling Strategies	263
10.3.3	Family-Based Type Checking of Preprocessor-Based Implementations.	264
10.3.4	Family-Based Type Checking for Feature-Oriented Programming	269
10.3.5	Family-Based Analysis with Variability Encoding	271
10.3.6	Feature-Based Analysis Strategies	272
10.3.7	Beyond Type Checking.	273
10.4	Case Studies and Experience	275
10.5	Tooling	276
10.6	Further Reading	277
	Exercises	278

Appendix A: Tool Support. 283

References 293

Index 309