

Contents

Foreword xvii

Preface xix

The KobrA project xx

Target audience xx

Document structure xxi

The library case study xxi

Acknowledgments xxii

Website xxii

Part 1 Introduction 1

1 Background 3

- 1.1 Reuse technologies 4
 - 1.1.1 Component-based development 4
 - 1.1.2 Architectural styles and design patterns 5
 - 1.1.3 Product line engineering 6
- 1.2 Development methods 7
 - 1.2.1 First-generation object-oriented methods 7
 - 1.2.2 Component-oriented methods 12
 - 1.2.3 Product line methods 16
 - 1.2.4 Object-oriented method frameworks 18
 - 1.2.5 Cleanroom techniques 22
- 1.3 Basic goals 25
 - 1.3.1 Simple 25
 - 1.3.2 Systematic 25
 - 1.3.3 Scalable 26
 - 1.3.4 Practical 26

2 Method overview 27

- 2.1 Core concepts 28
 - 2.1.1 Product line engineering 30
 - 2.1.2 Component modeling 31
 - 2.1.3 Component embodiment 35
 - 2.1.4 Project monitoring and control 38
- 2.2 Artifacts 40
 - 2.2.1 Frameworks 40
 - 2.2.2 Application 48
- 2.3 Process 50
 - 2.3.1 Framework engineering 52
 - 2.3.2 Application engineering 56
 - 2.3.3 Incremental development 57
- 2.4 Relationship to other methods 58
- 2.5 Key properties of KobrA 61
- 2.6 Road map 63

Part 2 Component modeling 65

3 The KobrA component model 67

- 3.1 Components 67
 - 3.1.1 Instances versus types 68
 - 3.1.2 Classes versus modules 69
 - 3.1.3 Subsystems 70
- 3.2 Component assembly 72
 - 3.2.1 Composition 72
 - 3.2.2 Clientship 73
 - 3.2.3 Ownership 74
 - 3.2.4 Containment 76
- 3.3 Modeling components 80
 - 3.3.1 The principle of uniformity 81
 - 3.3.2 The principle of locality 81
 - 3.3.3 The principle of parsimony 84
 - 3.3.4 The principle of encapsulation 85
- 3.4 Building component trees 90
 - 3.4.1 Polymorphism 90

- 3.4.2 Visibility rules 92
- 3.4.3 Consistency rules 94
- 3.4.4 Shaping containment trees 94

4 Specification 97

- 4.1 Specification artifacts 98
 - 4.1.1 Structural model 98
 - 4.1.2 Functional model 103
 - 4.1.3 Behavioral model 108
 - 4.1.4 Auxiliary artifacts 114
- 4.2 Specification process 114
 - 4.2.1 Structural modeling 114
 - 4.2.2 Functional modeling 116
 - 4.2.3 Behavioral modeling 117
 - 4.2.4 Quality control 117

5 Realization 119

- 5.1 Realization artifacts 119
 - 5.1.1 Structural model 120
 - 5.1.2 Activity model 123
 - 5.1.3 Interaction model 128
 - 5.1.4 Auxiliary artifacts 130
- 5.2 Realization process 132
 - 5.2.1 Structural modeling 133
 - 5.2.2 Activity modeling 135
 - 5.2.3 Interaction modeling 135
 - 5.2.4 Quality control 136

6 Containment 137

- 6.1 Package structure 138
- 6.2 Artifacts 141
 - 6.2.1 Specification relationships 143
 - 6.2.2 Realization relationships 147
- 6.3 Process 148
 - 6.3.1 The DNA spiral 149
 - 6.3.2 Component identification 151

- 6.3.3 Sub-component creation 152
- 6.3.4 Tree refactoring 153
- 6.3.5 Quality control 154

7 Context realization 155

- 7.1 Context realization artifacts 156
 - 7.1.1 Enterprise model 157
 - 7.1.2 Structural model 159
 - 7.1.3 Activity model 162
 - 7.1.4 Interaction model 167
- 7.2 Context realization process 168
 - 7.2.1 Enterprise modeling 169
 - 7.2.2 Structural modeling 169
 - 7.2.3 Usage modeling 170
 - 7.2.4 Interaction modeling 171
 - 7.2.5 Quality control 172

8 Public containment 173

- 8.1 Public containment versus public composition 174
 - 8.1.1 Public composition 174
 - 8.1.2 Public containment 176
- 8.2 Artifacts 178
 - 8.2.1 Public composition 178
 - 8.2.2 Public containment 184
- 8.3 Process 184
 - 8.3.1 Public composition 184
 - 8.3.2 Public containment 186
 - 8.3.3 Quality control 188

9 Generalization 189

- 9.1 Artifacts 190
 - 9.1.1 Structural model 191
 - 9.1.2 Functional model 193
 - 9.1.3 Behavioral model 195
- 9.2 Process 195
 - 9.2.1 Specialization 196

9.2.2 Generalization 197

9.2.3 Quality control 198

10 Components and patterns 199

10.1 What are patterns? 200

10.1.1 Patterns in UML 201

10.2 Leveraging existing patterns in Kobra 202

10.2.1 Layered architecture pattern 202

10.2.2 Distributed system architectures 209

10.3 Komponent containment patterns 214

10.3.1 Universal visibility pattern 216

10.3.2 Need-to-know pattern 216

10.3.3 Component library pattern 217

Part 3 Embodiment 221

11 Implementation 223

11.1 Separating refinement and translation 224

11.2 UML implementation profiles 227

11.2.1 The Normal Object Form 228

11.2.2 Transformation patterns 229

11.3 Implementation artifacts 233

11.3.1 Implementation structural model 233

11.3.2 Component diagrams 234

11.3.3 Pseudo-code 237

11.3.4 Source code 238

11.3.5 Sample implementations 239

11.4 Implementation process 244

11.4.1 Flattening 245

11.4.2 Refinement 246

11.4.3 Translation 247

11.4.4 Finalization 250

12 Component reuse 251

12.1 The role of reuse 252

12.2 Reuse artifacts 254

12.2.1 Desired and offered specifications 255

- 12.2.2 Conformance map 255
- 12.2.3 Semantic map 258
- 12.3 Reuse process 258
 - 12.3.1 Component selection 258
 - 12.3.2 Containment tree adaptation 260

13 Incremental development 264

- 13.1 System construction and deployment 264
 - 13.1.1 System construction 265
 - 13.1.2 Deployment 266
- 13.2 Activity sequencing 267
 - 13.2.1 Function-oriented versus component-based increments 268
- 13.3 Artifacts 269
 - 13.3.1 Stub specifications 269
 - 13.3.2 Component-based increments 270
- 13.4 Process 271
 - 13.4.1 Component increments 273
 - 13.4.2 Operation increments 273

Part 4 Product line engineering 275

14 Product line concepts 277

- 14.1 Promoting reuse 277
 - 14.1.1 Domain engineering 279
 - 14.1.2 Product line engineering 281
- 14.2 Product line engineering in KobrA 284
 - 14.2.1 Single-system engineering 286

15 Framework engineering 287

- 15.1 The generic component model 287
- 15.2 Artifacts 290
 - 15.2.1 Context realization 290
 - 15.2.2 KobrA component specification 304
 - 15.2.3 KobrA component realization 313
 - 15.2.4 KobrA component embodiment 318
- 15.3 Process 319

- 15.3.1 Variability identification 319
- 15.3.2 Decision modeling 320
- 15.3.3 KobrA component identification 320
- 15.3.4 KobrA component embodiment 321

16 Application engineering 324

- 16.1 Artifacts 325
 - 16.1.1 Decision resolution model 325
- 16.2 Process 325
 - 16.2.1 Context realization instantiation 326
 - 16.2.2 Specification and realization instantiation 327

Part 5 Project monitoring and control 331

17 Maintenance 333

- 17.1 Core principles 334
 - 17.1.1 Framework to application coupling 335
 - 17.1.2 Version orientation versus change set orientation 336
 - 17.1.3 Separation of concerns 336
- 17.2 Artifacts 337
 - 17.2.1 Configuration items 337
 - 17.2.2 Dependencies 341
 - 17.2.3 Evolution graphs 343
 - 17.2.4 Changes 344
- 17.3 Process 352
 - 17.3.1 Change management 352
 - 17.3.2 Configuration management 369

18 Quality assurance 372

- 18.1 Attaining quality 373
 - 18.1.1 What is quality? 373
 - 18.1.2 Non-functional requirements specification 375
 - 18.1.3 Quality assurance techniques 376
 - 18.1.4 Quality policy specification 377
 - 18.1.5 Quality documentation 379
- 18.2 Software testing 379

- 18.2.1 Testing artifacts 380
- 18.2.2 Testing process 382
- 18.3 Software inspection 385
 - 18.3.1 Organization technique 386
 - 18.3.2 Reading technique 387
 - 18.3.3 Inspection artifacts 388
 - 18.3.4 Inspection process 390

19 Quality modeling 393

- 19.1 What is a quality model? 394
- 19.2 Measurement of structural properties 395
 - 19.2.1 Size measurement 395
 - 19.2.2 Coupling measurement 397
 - 19.2.3 Complexity measurement 402
- 19.3 Example quality models 404
 - 19.3.1 Prediction models 404
 - 19.3.2 Quality benchmarks 406
 - 19.3.3 Simple ranking models 407

20 Applying Kobra 409

- 20.1 General characteristics of Kobra 409
- 20.2 Customization 411
 - 20.2.1 Framework and component genericity 412
 - 20.2.2 Context realization adaptation 413
 - 20.2.3 Komponent modeling adaptation 414
 - 20.2.4 Implementation, building, and deployment integration 414
- 20.3 Transfer 415
 - 20.3.1 Transfer planning 415
 - 20.3.2 Training 416
 - 20.3.3 Tool support 417

Appendices

A Metamodel 419

- A.1 Core 420
 - A.1.1 Komponent 423
 - A.1.2 Specification 427

- A.1.3 Realization 431
- A.1.4 Implementation 435
- A.1.5 Reuse 437
- A.1.6 System 438
- A.1.7 Physical Component 440
- A.1.8 Context Realization 442
- A.1.9 Quality assurance 443
- A.1.10 Inspection 444
- A.1.11 Measurement 445
- A.1.12 Testing 446
- A.1.13 Implementation Suite 448
- A.2 Maintenance 449
 - A.2.1 Assets 449
 - A.2.2 Change 452
 - A.2.3 Deliverable 454

B Process 456

- B.1 Process metamodel and notation 456
 - B.1.1 Process metamodel 457
 - B.1.2 Notation 458
 - B.1.3 Background 458
- B.2 Process model 459
 - B.2.1 Activities 459
 - B.2.2 Artifacts 461
 - B.2.3 Product flow 464

Glossary 483

References 489

Index 497

Acclaim for this book 505