

Contents

Preface	<i>xxi</i>
Organization of This Book	<i>xxii</i>
Who Should Read This Book?	<i>xxiii</i>
How to Use This Book	<i>xxiv</i>
Web Site	<i>xxiv</i>
Acknowledgments	<i>xxiv</i>
Introduction	<i>xxvii</i>
Component-Based Software Engineering	<i>xxix</i>
Challenges of CBSE	<i>xxx</i>
Components in Reliable Systems	<i>xxxiii</i>
References	<i>xxxvi</i>

	Part 1: The Definition and Specification of Components	1
1	Basic Concepts in CBSE	3
	Introduction	3
	Components	4
	Different Component Definitions	5
	Objects and Components	8
	Interfaces	9
	Contracts	10
	Patterns	12
	Frameworks	14
	Relations Between Concepts	16
	Frameworks and Components	17
	Frameworks and Contracts	18
	Frameworks and Patterns	18
	Conclusion	19
	Summary	20
	References	21
2	Specification of Software Components	23
	Introduction	23
	Current Component Specification Techniques	24
	Specifying the Semantics of Components	27
	Specifying Extrafunctional Properties of Components	34
	Summary	37
	References	37

	Part 2: Software Architecture and Components	39
3	Architecting Component-Based Systems	41
	Introduction	41
	Role of Software Architecture	42
	Assessment and Evaluation	42
	Configuration Management	43
	Dynamic Software Architectures	44
	Designing Software Architectures	44
	Architecture Design Process	45
	Architectural Styles	46
	Architecture-Driven Component Development	49
	Custom Components	50
	Preexisting Components	50
	Component-Driven Architecture Development	51
	Summary	53
	References	54
4	Component Models and Technology	57
	Introduction	57
	A Short Historical Perspective	58
	Component Interface and Connections	59
	Performing Services Transparently	60
	Notation and Plan	61
	Acme ADL	62
	Components and Ports	63
	Connectors and Roles	63
	Systems and Attachments	64
	Representations and Bindings	64

Properties, Constraints, Types, and Styles	65
Discussion	65
JavaBeans Component Model	66
Key Features	66
Interface of a Component	67
Implementation of a Component	68
Components Assembly	69
Packaging and Deployment	70
COM, DCOM, MTS, and COM+	71
Interfaces and Assembly	71
Implementation	71
Framework	72
Life Cycle	72
CCM	73
Interface and Assembly	73
Assemblies	75
Implementation of a Component	75
Framework: The Container Approach	76
Life Cycle	77
.NET Component Model	77
Interfaces and Assembly	77
Implementation	78
Framework	78
Life Cycle	79
The OSGI Component Model	79
Two Levels of Components	79
Interface of a Bundle Component	80
Assembly of Bundle Components	80
Implementation of a Bundle Component	83
Summary	83
Interface	83
Assembly	84

	Implementation	84
	Framework	84
	Conclusion	85
	References	86
	Part 3: Developing Software Components	87
5	CBD Process	89
	Introduction	89
	Software Process Models	90
	The Sequential Model	91
	Evolutionary Development	92
	Unified Process	95
	CBD	97
	Reuse Approach	97
	Separation of Development Processes	98
	Component-Based Software Life Cycle	100
	Development with Components	101
	Component Development	108
	Summary	111
	References	111
6	Semantic Integrity in CBD	115
	Introduction	115
	General Issues of Semantic Concern	116
	Specification Levels	117
	Weak and Strong Contracts	117
	Required and Provided Interfaces	118
	Levels of Formalism for Semantic Specifications	118
	An Example	119
	No Semantics	119

	Intuitive Semantics	120
	Structured Semantics	121
	Executable Semantics	122
	Formal Semantics	124
	Phases in a Component's Life	126
	Creation Phase	126
	Use Phase	127
	Maintenance Phase	128
	A Taxonomy for Component Semantics	128
	Summary	130
	References	131
7	<u>Role-Based Component Engineering</u>	135
	Introduction	135
	Encouraging the Use of Roles	138
	Role Technology	142
	Using Roles at the Design Level	142
	Using Roles at the Implementation Level	143
	Frameworks and Roles	145
	Black-Box and White-Box Frameworks	146
	A Model for Frameworks	148
	Dealing with Coupling	150
	Summary	152
	References	152
	<u>Part 4: Using Software Components</u>	155
8	<u>Dispelling the Myth of Component Evaluation</u>	157
	Introduction	157

	Multiple-Criteria Evaluation	160
	Genus: Preference Structure-Based Evaluation	160
	Species: Multiple-Attribute Utility Evaluation	166
	Exploding the Myth of Component Evaluation	167
	Components, Assemblies, Properties, and Determinants	168
	Dispensing with Inconsequential Component Evaluation	169
	Distinguishing Abstract from Concrete Interfaces	170
	Partial Satisfaction of Evaluation Attributes	171
	Fundamental Exemplar Incompleteness	172
	Assembly Evaluation and Search	173
	Multiple-Attribute Evaluation and Beyond	175
	Summary	176
	References	176
9	Component Composition and Integration	179
	Introduction	179
	Component Integration	181
	From Integration to Composition	182
	Predictable Assembly from Certifiable Components	185
	Prediction-Enabled Component Technology	186
	Summary	189
	References	190
10	Predicting System Trustworthiness	193
	Introduction	193
	What Else Can Be Done?	195
	Two Additional Useful Techniques for Predicting Component Interoperability	200

	Summary	202
	References	203
	Part 5: Software Product Lines	205
11	Components in Product Line Architectures	207
	Introduction	207
	From Products to Components	208
	Developing a Single Product	209
	Component Markets	209
	Established Reuse	211
	Product Families and Product Lines	212
	Product Populations	213
	The Lessons	214
	Component Approaches	215
	Libraries	215
	Object-Oriented Frameworks	216
	Component Frameworks	217
	Lego	218
	Koala	219
	Frameworks as Components	219
	Summary	220
	Conclusion	220
	References	220
12	The Koala Component Model	223
	Introduction	223
	Requirements for Koala	223
	Binding and Bind Time	224
	Koala Components	226

Provides and Requires Interfaces	227
Binding, Gluing, and Switches	228
Diversity Interfaces	229
Product-Line Architecture	229
Creating Products	233
Managing Versions and Variants	233
Subtyping	234
Conclusion	235
References	236

Part 6: Real-Time Software Components **237**

13	Components in Real-Time Systems	239
	Introduction	239
	Characteristics and Challenges of Real-Time Systems	241
	Real-Time Component Models	243
	Infrastructure Components	244
	Application-Specific Component Models	246
	Examples of Component Models	246
	Designing Component-Based Real-Time Systems	249
	Top-Level Design	250
	Detailed Design	251
	Architecture Analysis	251
	Scheduling	251
	WCET Verification	252
	Implementation of New Components	253
	System Build and Test	254
	Component Library	254

	Composition of Components	255
	Example: Real-Time Components in Rubus OS	257
	Rubus	257
	Extensions for CBSE	258
	Reuse of Real-Time Components	259
	On-Line Upgrades of Components	260
	Summary	261
	References	262
14	Testing Reusable Software Components in Safety-Critical Real-Time Systems	265
	Introduction	265
	Reuse and Exhaustive Testing	267
	Reuse and Statistical Evidence	270
	Component Reuse, Statistical Evidence, and Failure Behavior	270
	Sequential Systems	272
	Real-Time Systems	274
	Concurrent Real-Time Systems	276
	Summary	278
	References	278
15	Providing Real-Time Services for COTS Components	281
	Introduction	281
	Windows NT as an RTOS	283
	Real-Time Features of Windows NT	285
	Observations and Recommendations	289
	Windows NT in a Real-Time Setting	290
	Design of the Real-Time Application	291

	Experimental Results	293
	Summary of Results	296
	Conclusion	296
	References	298
	Part 7: Case Studies—CBD in Industrial Applications	299
16	Component-Based Embedded Systems	303
	Introduction	303
	Problem Domain	304
	Implications for Component Technology	305
	Contractually Specified Interfaces	306
	Unit of Composition and Independent Deployment	307
	Explicit Context Dependencies	308
	Component Granularity	308
	Reuse	309
	Architecture and Frameworks	310
	Location Transparency	310
	Portability and Platform Independence	310
	Component Wiring	311
	Embedded Systems with Synchronous Components	311
	Synchronous Approach	311
	Synchronous Software Components	312
	Case Study Description	313
	Architecture	314
	Dynamic Behavior	316
	Intercomponent Communication	317
	Prerequisites of Effective Development	319
	Summary	322
	References	322

17	Architectural Support for Reuse: A Case Study in Industrial Automation	325
	Introduction	325
	Industrial Automation Systems	326
	The Motivation for a Platform	329
	The Aspect Integrator Platform	331
	The AspectObject Paradigm and Object Organization	332
	The Architecture of the Aspect Integrator Platform	333
	Developing a Domain-Specific Application	346
	Reuse in AIP-Based Applications	348
	Some Words of Caution	350
	Summary	351
	References	352
18	A Framework for Integrating Business Applications	355
	Introduction	355
	Motivation	356
	Information Organizer: Support for Building Integrated Systems	358
	Information Organizer Basic Parts	358
	BOM	359
	BOF	362
	Modeling and Integration	363
	Structure of Aspects and Objects	364
	Patterns Constructed Using the BOF	365
	BPS	368

	Practical Experience	370
	Summary	371
	References	372
19	Industrial Experience with Dassault Syst�me Component Model	375
	Introduction	375
	The OM Component Model	376
	Defining an OM Component	377
	Using OM Components to Build Applications	377
	Implementing OM Components	378
	Discussion	380
	Different Kinds of Architecture	381
	Lessons Learned	382
	Framework Evolution	382
	Architecture and Components from an Industrial Perspective	383
	Summary	384
	References	385
	Contributors	387
	About the Authors	388
	Index	399
