

# Table of Contents

<b>1</b>	<b>How Can Invasive Software Composition Help You</b> .....	1
1.1	A Short Overview of the Book .....	4
1.2	The Component Dream .....	6
1.3	CoSy: A Component System for Adaptation and Extension ..	9
1.4	Aspect Separation: New Dimensions of Modularity .....	12
1.5	Requirements for Composition .....	18
1.6	Epilogue.....	20
1.6.1	Remarks .....	20
1.6.2	Goals of Invasive Software Composition .....	20

---

## Part I On the Way to Composition Systems

---

<b>2</b>	<b>Problems in Composition</b> .....	23
2.1	Requirements for Component Models .....	24
2.1.1	Code Reuse Requires Modularity .....	24
2.1.2	Adaptation Requires Parameterization .....	27
2.1.3	Improved Reuse Requires Standardization .....	31
2.2	Requirements for Basic Composition Techniques .....	34
2.2.1	Coupling Requires Connection .....	34
2.2.2	Evolution Requires Extensibility .....	37
2.2.3	Crosscutting Requires Aspect Separation .....	40
2.2.4	Composition Requires Scalability .....	42
2.2.5	Composition Requires Metamodeling, Introspection, and Reflection .....	44
2.3	Requirements for Composition Languages .....	56
2.4	Why Composition Systems Are Nothing Special .....	58
2.5	Epilogue.....	61
2.5.1	Summary .....	61
2.5.2	Related Work .....	61
<b>3</b>	<b>From Modular to Composition Systems</b> .....	63
3.1	Modular Systems .....	64
3.2	Object-Oriented Systems .....	65
3.3	Classical Component Systems .....	67

VI Table of Contents

3.4	Architecture Systems .....	72
3.5	Aspect Systems .....	78
3.6	Systems with Expression-Based Composition Languages ...	82
3.6.1	Subject-Oriented Programming (SOP) .....	83
3.6.2	Hyperspace Programming .....	85
3.6.3	Metaclass Composition .....	88
3.7	Systems with Control-Flow-Based Composition Languages..	90
3.7.1	Composition Filters .....	90
3.7.2	The $\lambda N$ -Calculus .....	95
3.7.3	Piccola and the $\pi\mathcal{L}$ -Calculus .....	99
3.8	Epilogue .....	102
3.8.1	Related Work .....	102
3.8.2	Summary: From Component to Composition Systems .....	103

---

**Part II The Concept of Invasive Composition**

---

<b>4</b>	<b>Invasive Software Composition .....</b>	<b>107</b>
4.1	What Is Invasive Composition? .....	108
4.1.1	What's in a Fragment Box? .....	114
4.1.2	What's in a Hook? .....	116
4.1.3	What's in a Composer? .....	118
4.1.4	Composition Language and Build Process .....	119
4.2	A Library for Invasive Software Composition .....	121
4.2.1	Configuration and Simple Composition Programs ...	122
4.2.2	The Component Model of Compost .....	124
4.2.3	Predefined Hooks in Compost .....	127
4.3	The Basic Composition Technique in Compost .....	128
4.4	The Composition Process in Compost .....	132
4.4.1	Composing Components in Compost .....	132
4.4.2	A Larger Example: Binding Super Classes .....	137
4.5	Epilogue .....	141
4.5.1	Remarks .....	141
4.5.2	Related Work .....	142
4.5.3	Summary .....	144
4.5.4	History .....	144
<b>5</b>	<b>How To Make Invasive Composition Reliable .....</b>	<b>147</b>
5.1	Sound Invasive Composition .....	148
5.1.1	Sound Extensions .....	149
5.2	Information Hiding in Invasive Composition .....	151
5.3	Termination of the Composition Process .....	153
5.4	Composition Process and Variant Selection .....	154
5.4.1	Demand-Driven Recomposition (Lazy System Builds)	156

5.4.2	Metacomposition: Configuring Architectures with Composition Classes .....	159
5.4.3	What a Composition System Can Check .....	162
5.5	Epilogue .....	163
5.5.1	Related Work .....	163
5.5.2	Summary: The Role of Invasive Composition .....	163

---

### Part III Applications of Invasive Composition with Declared Hooks

---

<b>6</b>	<b>Generalized Parameterization .....</b>	<b>167</b>
6.1	Generic Types and Identifiers .....	170
6.1.1	Instantiation Methods for Generic Templates .....	171
6.1.2	An Example: The Generic List Classes of Compost ...	172
6.2	Invasive Frameworks with Generic Statements .....	177
6.2.1	Expanding Code Templates in Ccc .....	178
6.3	Avoiding the Inheritance Anomaly with Generic Modifiers ..	183
6.4	Epilogue .....	185
6.4.1	Related Work .....	185
6.4.2	Summary .....	186
6.4.3	History .....	187
<b>7</b>	<b>Architecture As Composition .....</b>	<b>189</b>
7.1	Ports As Hooks and Connectors As Composers .....	190
7.1.1	A Model for Ports and Connectors .....	190
7.2	Topology and Transfer Aspect of Connections .....	193
7.2.1	Binding Communication Partners in Topological Connections .....	195
7.2.2	Rewriting Topological to Concrete Connections .....	199
7.2.3	Connections Without Explicit Gate Objects .....	202
7.2.4	Connections with Explicit Gate Objects .....	203
7.2.5	Reuse of Architectures .....	205
7.3	Epilogue .....	208
7.3.1	Related Work .....	208
7.3.2	Summary .....	208

---

### Part IV Applications of Invasive Composition with Implicit Hooks

---

<b>8</b>	<b>Inheritance As Hook Extension .....</b>	<b>213</b>
8.1	Simple Feature Extension .....	217
8.1.1	Subclassing by Mixins in a Record Calculus .....	217
8.1.2	Invasive Extension of Feature Hooks Models Mixin Operators .....	219

VIII Table of Contents

8.1.3	Invasive Extension Between Inheritance and Delegation .....	223
8.2	Inheritance and Delegation Facades in Compost .....	224
8.2.1	Multiple Inheritance in Compost .....	226
8.3	Comparing Inheritance, Generics, and Frameworks .....	227
8.3.1	Rules for Invasive Extension .....	231
8.4	Epilogue.....	232
8.4.1	Related Work .....	232
8.4.2	Summary .....	233
<b>9</b>	<b>Views with Sound Extensions .....</b>	<b>235</b>
9.1	Sound Extensions of Boxes .....	238
9.1.1	Method Slice Extensions are Sound Extensions .....	238
9.1.2	Feature Group Extensions are Sound Extensions .....	240
9.1.3	Forward Flow Extensions are Sound Extensions .....	243
9.1.4	Query Method Extensions are Sound Extensions .....	244
9.2	Intrusive Data Structures as Feature Group Extensions .....	244
9.2.1	Intrusive Data Structures in Program Optimization .....	245
9.2.2	Intrusive Functors for Intrusive Data Structures .....	249
9.3	Solving the Syntactic Fragile Base Class Problem .....	253
9.4	Epilogue.....	255
9.4.1	Related Work .....	255
9.4.2	Summary .....	257
<b>10</b>	<b>Aspect Composition As Distribution of Aspect Boxes .....</b>	<b>259</b>
10.1	Composition of Aspects with Distributors .....	263
10.1.1	Weaving on the Program Representation Level .....	263
10.1.2	The Upper Level: Weaving on Fragment Boxes and Hooks.....	264
10.1.3	Parameterized Weaving.....	266
10.1.4	Aspects in Hungarian Notation .....	267
10.2	Sound Distributions, Sound Weavings .....	268
10.2.1	Examples of Sound Aspects .....	269
10.3	The Universe of Aspect and Weaving Languages.....	270
10.4	Epilogue.....	270
10.4.1	Related Work .....	270
10.4.2	Summary .....	271
<b>11</b>	<b>The Progress in Invasive Composition .....</b>	<b>273</b>
11.1	What Has Been Achieved? .....	274
11.2	Related Work.....	277
11.3	Software Composition in the Future.....	278

<b>Appendix A</b>	<b>Programming Languages and Compilers</b> .....	279
<b>Appendix B</b>	<b>The Production Cell</b> .....	285
<b>Appendix C</b>	<b>A Facet Classification of Hooks</b> .....	289
<b>Appendix D</b>	<b>The Structure of Compost in UML</b> .....	295
<b>Appendix E</b>	<b>Legend of the Box Graphics</b> .....	299
<b>Glossary</b>	.....	301
<b>Index</b>	.....	313
<b>References</b>	.....	321