Uwe Afimann

Invasive Software Composition

With 127 Figures and 9 Tables

Springer
Table of Contents

1 How Can Invasive Software Composition Help You .................. 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

Problems in Composition ....................................................... 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 Metamodelling, 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

From Modular to Composition Systems ..................................... 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 AiV-Calculus .......................................... 95
  3.7.3 Piccola and the 7rl£-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

4 Invasive Software Composition ......................................... 107
  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'sinaComposer? ..................................... 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

5 How To Make Invasive Composition Reliable............ 147
  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
Part III Applications of Invasive Composition with Declared Hooks

6 Generalized Parameterization ..................................................167
   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

7 Architecture As Composition ...................................................189
   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

8 Inheritance As Hook Extension ..................................213
   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
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Start Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1.3</td>
<td>Invasive Extension Between Inheritance and Delegation</td>
<td>223</td>
</tr>
<tr>
<td>8.2</td>
<td>Inheritance and Delegation Facades in Compost</td>
<td>224</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Multiple Inheritance in Compost</td>
<td>226</td>
</tr>
<tr>
<td>8.3</td>
<td>Comparing Inheritance, Generics, and Frameworks</td>
<td>227</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Rules for Invasive Extension</td>
<td>231</td>
</tr>
<tr>
<td>8.4</td>
<td>Epilogue</td>
<td>232</td>
</tr>
<tr>
<td>8.4.1</td>
<td>Related Work</td>
<td>232</td>
</tr>
<tr>
<td>8.4.2</td>
<td>Summary</td>
<td>233</td>
</tr>
<tr>
<td>9</td>
<td>Views with Sound Extensions</td>
<td></td>
</tr>
<tr>
<td>9.1</td>
<td>Sound Extensions of Boxes</td>
<td></td>
</tr>
<tr>
<td>9.1.1</td>
<td>Method Slice Extensions are Sound Extension*</td>
<td>238</td>
</tr>
<tr>
<td>9.1.2</td>
<td>Feature Group Extensions are Sound Extensions</td>
<td>240</td>
</tr>
<tr>
<td>9.1.3</td>
<td>Forward Flow Extensions are Sound Extensions</td>
<td>243</td>
</tr>
<tr>
<td>9.1.4</td>
<td>Query Method Extensions are Sound Extensions</td>
<td>244</td>
</tr>
<tr>
<td>9.2</td>
<td>Intrusive Data Structures as Feature Group Extensions</td>
<td>244</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Intrusive Data Structures in Program Optimization</td>
<td>245</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Intrusive Functors for Intrusive Data Structures</td>
<td>249</td>
</tr>
<tr>
<td>9.3</td>
<td>Solving the Syntactic Fragile Base Class Problem</td>
<td>253</td>
</tr>
<tr>
<td>9.4</td>
<td>Epilogue</td>
<td>255</td>
</tr>
<tr>
<td>9.4.1</td>
<td>Related Work</td>
<td>255</td>
</tr>
<tr>
<td>9.4.2</td>
<td>Summary</td>
<td>257</td>
</tr>
<tr>
<td>10</td>
<td>Aspect Composition As Distribution of Aspect Boxes</td>
<td></td>
</tr>
<tr>
<td>10.1</td>
<td>Composition of Aspects with Distributors</td>
<td></td>
</tr>
<tr>
<td>10.1.1</td>
<td>Weaving on the Program Representation Level</td>
<td>263</td>
</tr>
<tr>
<td>10.1.2</td>
<td>The Upper Level: Weaving on Fragment Boxes and Hooks</td>
<td>264</td>
</tr>
<tr>
<td>10.1.3</td>
<td>Parameterized Weaving</td>
<td>266</td>
</tr>
<tr>
<td>10.1.4</td>
<td>Aspects in Hungarian Notation</td>
<td>267</td>
</tr>
<tr>
<td>10.2</td>
<td>Sound Distributions, Sound Weavings</td>
<td>268</td>
</tr>
<tr>
<td>10.2.1</td>
<td>Examples of Sound Aspects</td>
<td>269</td>
</tr>
<tr>
<td>10.3</td>
<td>The Universe of Aspect and Weaving Languages</td>
<td>270</td>
</tr>
<tr>
<td>10.4</td>
<td>Epilogue</td>
<td>270</td>
</tr>
<tr>
<td>10.4.1</td>
<td>Related Work</td>
<td>270</td>
</tr>
<tr>
<td>10.4.2</td>
<td>Summary</td>
<td>271</td>
</tr>
<tr>
<td>11</td>
<td>The Progress in Invasive Composition</td>
<td></td>
</tr>
<tr>
<td>11.1</td>
<td>What Has Been Achieved?</td>
<td>274</td>
</tr>
<tr>
<td>11.2</td>
<td>Related Work</td>
<td>277</td>
</tr>
<tr>
<td>11.3</td>
<td>Software Composition in the Future</td>
<td>278</td>
</tr>
</tbody>
</table>