



Formes Circulaires by Robert Delaunay

SOFTWARE ENGINEERING SERIES

*A UML*

*Pattern Language*

Suranaree University of Technology



3106100052715

## Table of Contents

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>Introduction</b>                                         | <b>1</b>  |
| Patterns and the UML .....                                  | 2         |
| Levels and Shared Idioms .....                              | 5         |
| Using This Book .....                                       | 7         |
| Resources, Sources, and References .....                    | 8         |
| <b>Part I Getting Started</b>                               | <b>11</b> |
| <b>1 Pattern Essentials</b>                                 | <b>13</b> |
| 1.1 Patterns and Paradigms .....                            | 14        |
| 1.1.1 The Idea of a Pattern .....                           | 15        |
| 1.2 Elements of Patterns .....                              | 17        |
| 1.2.1 A Simple Example .....                                | 18        |
| 1.3 Interpreting the Patterns in This Book .....            | 21        |
| 1.3.1 This Book's Pattern Format .....                      | 22        |
| <b>2 The Unified Modeling Language</b>                      | <b>25</b> |
| 2.1 The UML, Briefly Put .....                              | 26        |
| 2.2 Roots .....                                             | 27        |
| 2.2.1 Key Players .....                                     | 29        |
| 2.3 Understanding the UML .....                             | 33        |
| 2.4 Unification: The Methods Wars Are Over .....            | 34        |
| 2.4.1 Best Practices: In the Eye of the Beholder .....      | 35        |
| 2.4.2 An Independent-Minded Modeling Language .....         | 36        |
| <b>3 UML Essentials, Elements, and Artifacts</b>            | <b>39</b> |
| 3.1 Elements, Viewpoints, and Views .....                   | 40        |
| 3.1.1 Models and Model Elements .....                       | 42        |
| 3.1.2 Diagrams .....                                        | 43        |
| 3.2 Packages .....                                          | 44        |
| 3.2.1 Models: Packages of Views .....                       | 47        |
| 3.2.2 Subsystems: Packages of Behavior and Operations ..... | 47        |
| 3.2.3 Frameworks: Packages of Patterns .....                | 48        |
| 3.3 Extensions .....                                        | 49        |
| 3.3.1 Tagged Values .....                                   | 50        |
| 3.3.2 Constraints .....                                     | 50        |
| 3.3.3 Stereotypes .....                                     | 50        |
| 3.3.4 Profiles .....                                        | 51        |
| 3.4 Symbols .....                                           | 51        |
| 3.4.1 Actor .....                                           | 52        |
| 3.4.2 Use Case/Collaboration .....                          | 52        |
| 3.4.3 Class/Object/Type/Active Class .....                  | 53        |
| 3.4.4 Interface .....                                       | 54        |

|                                                       |           |
|-------------------------------------------------------|-----------|
| 3.4.5 Component .....                                 | 54        |
| 3.4.6 Node .....                                      | 54        |
| 3.4.7 Package .....                                   | 55        |
| 3.4.8 State .....                                     | 55        |
| 3.4.9 Note .....                                      | 56        |
| 3.5 Lines .....                                       | 56        |
| 3.5.1 Messages .....                                  | 56        |
| 3.5.2 Relationships in General .....                  | 57        |
| 3.5.3 Relationships: Some Types of Associations ..... | 58        |
| 3.5.4 Relationships: Some Uses of Dependency .....    | 59        |
| 3.5.5 Abstraction: Other Uses of Dependency .....     | 60        |
| 3.6 Diagrams .....                                    | 61        |
| 3.6.1 Class Diagram .....                             | 61        |
| 3.6.2 Use Case Diagram .....                          | 62        |
| 3.6.3 Interaction Diagrams .....                      | 63        |
| 3.6.4 State Diagrams .....                            | 65        |
| 3.6.5 Activity Diagrams .....                         | 66        |
| 3.6.6 Implementation Diagrams .....                   | 68        |
| 3.7 Further Reading .....                             | 69        |
| <b>Part II The Pattern Language</b> .....             | <b>71</b> |
| <b>4 Patterns of Style</b> .....                      | <b>73</b> |
| Catalogue .....                                       | 73        |
| Context .....                                         | 74        |
| Common Forces .....                                   | 74        |
| Discussion .....                                      | 75        |
| 4.1 Attributes as Compositions to Types .....         | 75        |
| 4.2 Providing Focus .....                             | 79        |
| 4.3 Explicit Elision .....                            | 81        |
| 4.4 Tree Routing .....                                | 83        |
| 4.5 Tombstone Packages .....                          | 85        |
| 4.6 Inheritance Goes Up .....                         | 87        |
| 4.7 Rotated Text .....                                | 88        |
| 4.8 Dual Associations .....                           | 89        |
| 4.9 Billboard Packages .....                          | 91        |
| 4.10 Text Workarounds .....                           | 93        |
| 4.11 Seven Plus or Minus Two .....                    | 96        |
| Summary .....                                         | 97        |

|                                                  |            |
|--------------------------------------------------|------------|
| <b>5 Patterns of Substance</b>                   | <b>99</b>  |
| Catalogue.....                                   | 99         |
| Context .....                                    | 100        |
| Common Forces .....                              | 100        |
| Discussion .....                                 | 101        |
| 5.1 Standard Diagrams.....                       | 101        |
| 5.2 Implementation or Representation .....       | 102        |
| 5.3 Digestible Chunks .....                      | 103        |
| 5.4 Attach the Actor.....                        | 104        |
| 5.5 Business Rules Invariably Constrain.....     | 105        |
| 5.6 Dynamic Object Types .....                   | 107        |
| 5.7 Many-to-Many Class Trio.....                 | 109        |
| 5.8 Model the Seams .....                        | 111        |
| 5.9 Packaging Partitions.....                    | 113        |
| 5.10 Let the Tools Do the Work.....              | 115        |
| 5.11 Opaque Packages .....                       | 117        |
| Summary .....                                    | 119        |
| <b>6 Domain Patterns</b>                         | <b>121</b> |
| Catalogue.....                                   | 121        |
| Context .....                                    | 122        |
| Common Forces .....                              | 122        |
| Discussion .....                                 | 123        |
| 6.1 Domain Model Is Essential .....              | 125        |
| 6.2 Actors Play Essential Roles .....            | 126        |
| 6.3 Factor the Actor .....                       | 127        |
| 6.4 Essential Actions.....                       | 128        |
| 6.5 Essential Vocabulary .....                   | 129        |
| 6.6 Objectify Internal Roles .....               | 130        |
| 6.7 ToBe Model.....                              | 131        |
| 6.8 AsIs Model.....                              | 132        |
| Summary .....                                    | 134        |
| <b>7 Product Patterns</b>                        | <b>137</b> |
| Catalogue.....                                   | 137        |
| Context .....                                    | 138        |
| Forces .....                                     | 138        |
| Discussion .....                                 | 139        |
| 7.1 Manageable Product .....                     | 139        |
| 7.2 Product Stakeholders Are Model Clients ..... | 141        |
| 7.3 Product Events in Context .....              | 142        |
| 7.4 Use Cases Represent Requirements.....        | 144        |
| 7.5 Boundary-Control-Entity (BCE).....           | 145        |
| 7.6 Product Chunks Digest Easily.....            | 148        |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| 7.7 Product Traces Support Robustness .....                         | 149        |
| 7.8 Use Cases: Work as Packages .....                               | 150        |
| 7.9 Tests Need Models .....                                         | 151        |
| 7.10 Configuration Management Model .....                           | 152        |
| Summary .....                                                       | 154        |
| <b>8 Component Patterns</b> .....                                   | <b>155</b> |
| Catalogue .....                                                     | 155        |
| Context .....                                                       | 156        |
| Discussion .....                                                    | 156        |
| 8.1 Separation of Concerns .....                                    | 157        |
| 8.2 Whole Components .....                                          | 159        |
| 8.3 Icons Clarify Components .....                                  | 160        |
| 8.4 Icons Identify Nodes .....                                      | 162        |
| 8.5 Specification Backplane .....                                   | 164        |
| 8.6 Components Manage Change .....                                  | 165        |
| 8.7 Configured and Released Packages .....                          | 166        |
| 8.8 Model for Maintenance .....                                     | 167        |
| Summary .....                                                       | 169        |
| <b>Part III Another Starting Point</b> .....                        | <b>171</b> |
| <b>9 Patterns in Context</b> .....                                  | <b>173</b> |
| 9.1 A Little Starting Context .....                                 | 175        |
| 9.1.1 Force 1: Structuring Abstraction, Abstracting Structure ..... | 175        |
| 9.1.2 Force 2: Guiding Creativity, Creative Guidance .....          | 176        |
| 9.1.3 Force 3: The Search for Quality and Reuse .....               | 177        |
| 9.1.4 Broader Cultural and Professional Forces .....                | 178        |
| 9.2 The Pattern Idea .....                                          | 179        |
| 9.2.1 First Hints .....                                             | 179        |
| 9.2.2 The Early Years .....                                         | 180        |
| 9.2.3 The Idea Emerges .....                                        | 181        |
| 9.2.4 The Beginnings of PLoP .....                                  | 183        |
| 9.2.5 The Gang of Four and After .....                              | 184        |
| 9.3 Patterns as Literature .....                                    | 185        |
| 9.4 Types of Software Patterns .....                                | 188        |
| 9.4.1 CoplienForm .....                                             | 190        |
| 9.4.2 GammaForm .....                                               | 193        |
| 9.5 The Roots: Alexander on Patterns and Pattern Languages .....    | 195        |
| 9.6 A Note on This Language .....                                   | 198        |
| 9.7 The Importance of Patterns .....                                | 199        |
| 9.8 Where Is It All Going? .....                                    | 202        |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| <b>10 The UML in Context</b>                                        | <b>205</b> |
| 10.1 Why Make System Models?                                        | 205        |
| 10.1.1 What Use Is a Model?                                         | 207        |
| 10.2 Every Picture Tells a Story: The UML as a Modeling Language    | 208        |
| 10.3 The UML Specification and Metamodel                            | 210        |
| 10.4 What Do We Model?                                              | 213        |
| 10.4.1 Architecture                                                 | 214        |
| 10.4.2 Domains                                                      | 215        |
| 10.4.3 Products                                                     | 217        |
| 10.4.4 Solutions                                                    | 217        |
| 10.5 Abstraction and Architecture Made Simple                       | 219        |
| 10.5.1 Abstraction                                                  | 219        |
| 10.5.2 Architecture                                                 | 222        |
| 10.6 Perspectives: A Generic Modeling Framework                     | 225        |
| <b>11 Putting It All Together: Reflecting on the Work of Design</b> | <b>227</b> |
| 11.1 The Work of Design                                             | 228        |
| 11.1.1 What Is Design?                                              | 229        |
| 11.1.2 Beyond Patterns and Paradigms                                | 231        |
| 11.2 Elements of Reflective Design                                  | 232        |
| 11.2.1 Problem Setting                                              | 234        |
| 11.2.2 A Language of Design                                         | 235        |
| 11.2.3 A Language about Designing                                   | 236        |
| 11.2.4 Performance                                                  | 237        |
| 11.2.5 Closure                                                      | 239        |
| 11.2.6 Reflective Design and Systems Modeling                       | 239        |
| 11.3 To Be Continued...                                             | 240        |
| <b>References</b>                                                   | <b>241</b> |
| <b>Index</b>                                                        | <b>247</b> |