

Chapter 1 – Introduction	13
1.1 – OBJECT-ORIENTED DESIGN	13
1.2 – LOGICAL VS PHYSICAL DESIGN	14
1.3 – MODULARITY	15
<i>1.3.1 – Unit of Modularity – The JAR File</i>	<i>16</i>
<i>1.3.2 – OSGi</i>	<i>16</i>
1.4 – Who This Book is For	16
1.5 – CHAPTER SYNOPSIS	17
Chapter 2 – Module Defined	18
Chapter 3 – Two Facets of Modularity	18
Chapter 4 – Architecture and Modularity	18
Chapter 5 – Taming the Beast	18
Chapter 6 – Realizing Reuse	18
Chapter 7 – Modularity and SOA	18
Chapter 8 – Reference Implementation	18
<i>1.5.2 – Part 2 - The Pattern Catalog</i>	<i>18</i>
Chapter 9 – Module Base Patterns	18
Chapter 10 – Module Dependency Patterns	19
Chapter 11 – Module Usability Patterns	19
Chapter 12 – Module Extensibility Patterns	19
Chapter 13 – Module Utility Patterns	19
<i>1.5.3 - OSGi and The Patterns</i>	<i>19</i>
Chapter 14 – Introducing OSGi	19
Chapter 15 – The Patterns Using OSGi	20

Chapter 16 – The Loan Sample and OSGi	20
Chapter 17 – OSGi and Scala	20
Chapter 18 – OSGi and Groovy	20
Chapter 19 – The Future of Modularity and OSGi	20
1.6 – PATTERN FORM	20
Pattern Name	20
Pattern Statement	21
Sketch	21
Description	21
Implementation Variations	21
Consequences	21
Samples	21
Wrapping Up	21
1.7 – PATTERN CATALOG	22
<i>1.7.1 – Base Patterns</i>	<i>22</i>
<i>1.7.2 – Dependency Patterns</i>	<i>22</i>
<i>1.7.3 – Usability Patterns</i>	<i>22</i>
<i>1.7.4 – Extensibility Patterns</i>	<i>22</i>
<i>1.7.5 – Utility Patterns</i>	<i>22</i>
1.8 - THE CODE	23
Chapter 2 – Module Defined	24
2.1 – DEFINING A MODULE	24
<i>2.1.1 – Deployable</i>	<i>25</i>
<i>2.1.2 – Manageable</i>	<i>25</i>

2.1.3 – Testable	26
2.1.4 – Natively Reusable	26
2.1.5 – Composable	26
2.1.6 – Stateless	26
2.2 – A BIT MORE SUCCINCTLY	26
Chapter 3 – The Two Facets of Modularity	28
3.1 – THE RUNTIME MODEL	28
3.2 – THE DEVELOPMENT MODEL	28
3.2.1 – The Programming Model	28
3.2.2 – The Design Paradigm	29
3.2.2.1 – Object-Orientation	29
3.2.2.2 – Enterprise JavaBeans	30
3.2.2.3 – Lessons Learned	30
3.3 – MODULARITY TODAY	31
3.4 - CONCLUSION	31
Chapter 4 – Architecture and Modularity	33
4.1 – DEFINING ARCHITECTURE	33
4.2 – A SOFTWARE ARCHITECTURE STORY	34
4.2.1 – The Ivory Tower	34
4.2.2 – Turtles and the Tower	35
4.3 – THE GOAL OF ARCHITECTURE	37
4.3.1 – The Paradox	38
4.3.2 – Eliminating Architecture	38

4.4 – MODULARITY – THE MISSING INGREDIENT	39
4.5 – ANSWERING OUR QUESTIONS	40
4.6 - CONCLUSION	41
Chapter 5 – Taming the Beast	42
5.1 – ENTERPRISE COMPLEXITY	42
5.2 – TECHNICAL DEBT	43
5.3 – DESIGN ROT	44
<i>5.3.1 – Hinder Maintenance</i>	<i>44</i>
<i>5.3.2 – Prevent Extensibility</i>	<i>45</i>
<i>5.3.3 – Inhibit Reusability</i>	<i>45</i>
<i>5.3.4 – Restrict Testability</i>	<i>45</i>
<i>5.3.5 – Hamper Integration</i>	<i>45</i>
<i>5.3.6 – Limit Understanding</i>	<i>46</i>
5.4 – CYCLIC DEPENDENCIES – THE DEATH KNELL	46
<i>5.4.1 – Types of Cycles</i>	<i>46</i>
<i>5.4.2 – Creeping Cycles</i>	<i>49</i>
<i>5.4.3 – Managing Cycles</i>	<i>50</i>
<i>5.4.4 – But Are Cycles Always Bad?</i>	<i>50</i>
5.5 – IN THE REAL WORLD	51
5.6 – JOINTS, MODULES, & SOLID	51
5.7 – MANAGING COMPLEXITY	52
<i>5.7.1 – Illustrating the Benefit</i>	<i>52</i>
5.8 – BENEFITS OF MODULARITY	54

Chapter 6 – Realizing Reuse	56
6.1 – THE USE/REUSE PARADOX	56
6.2 – THE REUSE DISCLAIMER	57
6.2.1 – <i>Granularity</i>	57
6.2.2 – <i>Weight</i>	58
6.3 – REUSE OR USE	59
6.4 -MODULAR TENSION	59
6.5 – MODULAR DESIGN	60
6.6 – CONCLUSION	61
Chapter 7 – Modularity and SOA	63
7.1 – ALL THE WAY DOWN, REVISITED	63
7.2.1 – <i>Structural Flexibility – Different Entities, Different Purpose</i>	64
7.2 – GRANULARITY, ARCHITECTURE’S NEMESIS	65
7.2.1 – <i>A Really Simple Example</i>	65
7.2.2 – <i>Bring It Up A Level</i>	67
7.2.3 – <i>Another Dimension</i>	67
7.2.4 – <i>The Complete Picture</i>	68
7.2.5 – <i>A Service Example</i>	69
7.3 – AN ALTERNATIVE VIEW	72
7.6 – CONCLUSION	73
Chapter 8 – Reference Implementation	75
8.1 – WHY NO OSGI?	75
8.2 – LET’S GET STARTED – THE SYSTEM	75

8.3 – VERSION ONE	76
8.4 – FIRST REFACTORING	78
<i>8.4.1 – Wrapping Up and Getting Ready for the Next Refactoring</i>	<i>80</i>
8.5 – SECOND REFACTORING	81
8.6 – THIRD REFACTORING	83
<i>8.6.1 – Wrapping Up and Getting Ready for the Fourth Refactoring</i>	<i>85</i>
8.7 – FOURTH REFACTORING	86
<i>8.7.1 – A Note on the Benefit of OSGi</i>	<i>87</i>
<i>8.7.2 – Wrapping Up and Getting Ready for the Next Refactoring</i>	<i>88</i>
8.8 – FIFTH REFACTORING	88
8.9 – SIXTH REFACTORING	89
8.10 – SEVENTH REFACTORING	92
8.11 – THE POSTMORTEM	93
<i>8.11.1 – A Note On Module Testing</i>	<i>94</i>
<i>8.11.2. – A Note On Managing Modules</i>	<i>95</i>
<i>8.11.3 – A Note on Module Reuse</i>	<i>97</i>
<i>8.11.4 – A Note on The Build</i>	<i>98</i>
<i>8.11.5 – A Note on Object Orientation</i>	<i>98</i>
8.12 – CONCLUSION	99
Base Patterns	101
Manage Relationships	102
Module Reuse	110
Cohesive Modules	121

Dependency Patterns	126
Acyclic Relationships	127
Levelize Modules	137
Physical Layers	142
Container Independence	150
Independent Deployment	156
Usability Patterns	163
Published Interface	164
External Configuration	174
Default Implementation	179
Module Facade	184
Extensibility Patterns	191
Abstract Modules	192
Implementation Factory	199
Separate Abstractions	206
Utility Patterns	214
Colocate Exceptions	215
Levelized Build	220
Test Module	229
Chapter 14 – Introducing OSGi	235
14.1 – INTRODUCING OSGi	235
14.1.2 - A BIT OF HISTORY	235

14.1.2 - <i>BENEFITS OF OSGi</i>	235
14.3 - THE OSGi BUNDLE	238
14.3.1 - <i>BUNDLE STATE</i>	238
14.3.2 - <i>OSGi MICRO-SERVICES</i>	238
14.3.3 - <i>BLUEPRINT SERVICES</i>	239
14.4 - OSGi RUNTIME MANAGEMENT	239
14.5 – THE TWO FACETS OF MODULARITY REVISITED	240
14.6 - OSGi AND THE PATTERNS	240
14.6.1 - <i>Exporting & Importing Packages</i>	240
14.6.2 - <i>Dynamism</i>	241
14.6.3 - <i>Blueprint Specification</i>	241
Chapter 15 - THE LOAN SAMPLE AND OSGi	243
15.1 - GETTING STARTED	243
15.2 - THE MANIFESTS	244
15.3 - THE SERVICES	245
15.3.1 - <i>Blueprint Services</i>	246
15.3.2 - <i>The Loan Bean Configuration</i>	246
15.3.3 - <i>OSGi Service Declarations</i>	249
15.4 - INSTALLATION	250
15.5 - EXECUTION AND BEYOND	250
Chapter 16. - OSGi and Scala	252
16.1. - GETTING STARTED	252
16.2 - THE SCALA CODE	252

16.2.1 - The Manifest	254
16.3 - SCALA BEAN CONFIGURATION	254
16.4 - SCALA SERVICE CONFIGURATION	255
16.5 - BUILDING THE GROOVY MODULE	255
16.6 - INSTALLATION AND EXECUTION	255
Chapter 17. - OSGi and Groovy	257
17.1. - GETTING STARTED	257
17.2 - THE GROOVY CODE	257
17.2.1 - The Manifest	259
17.3 - GROOVY BEAN CONFIGURATION	259
17.4 - GROOVY SERVICE CONFIGURATION	259
17.5 - BUILDING THE GROOVY MODULE	260
17.6 - INSTALLATION AND EXECUTION	260
Chapter 18 - Future of OSGi	262
1818..1 - OSGI AS ENABLER	262
18.2 - THE DISRUPTION	263
18.2.1 - A BIT OF [RECENT] PLATFORM HISTORY	263
18.2.1.1 - Apple	263
18.2.1.2 - Eclipse	263
18.2.1.3 - Hudson	264
18.2.1.4 - Social Media	264
18.2.1.5 - And Others	264
18.3 - THE POWER OF ECOSYSTEMS	264

<i>18.3.1. - ECOSYSTEM AND THE TWO FACETS OF MODULARITY</i>	<i>265</i>
18.4 - THE ECOSYSTEM	265
Additional Notes	266
OPEN CLOSED PRINCIPLE (OCP)	268
LISKOV SUBSTITUTION PRINCIPLE (LSP)	270
DEPENDENCY INVERSION PRINCIPLE (DIP)	272
INTERFACE SEGREGATION PRINCIPLE	273
COMPOSITE REUSE PRINCIPLE (CRP)	275