

# Table of Contents

	<b>Foreword</b> . . . . .	<b>xv</b>
	<b>About This Book</b> . . . . .	<b>xvii</b>
	<b>About The Authors</b> . . . . .	<b>xxiii</b>
	<b>Guide To The Reader</b> . . . . .	<b>xxvii</b>
<b>Part I</b>	<b>Some Concepts</b> . . . . .	<b>1</b>
<b>1</b>	<b>On Patterns and Pattern Languages</b> . . . . .	<b>3</b>
1.1	Patterns Introduced . . . . .	4
1.2	Inside Patterns . . . . .	6
1.3	Between Patterns . . . . .	10
1.4	Into Pattern Languages . . . . .	13
1.5	Patterns Connected . . . . .	15
<b>2</b>	<b>On Distributed Systems</b> . . . . .	<b>17</b>
2.1	Benefits of Distribution . . . . .	18
2.2	Challenges of Distribution . . . . .	20
2.3	Technologies for Supporting Distribution . . . . .	22
2.4	Limitations of Middleware . . . . .	32
<b>3</b>	<b>On the Pattern Language</b> . . . . .	<b>33</b>
3.1	Intent, Scope, and Audience . . . . .	34
3.2	Origins and Genesis . . . . .	35

3.3	Structure and Content . . . . .	36
3.4	Presentation . . . . .	44
3.5	Practical Use . . . . .	49
<b>Part II</b>	<b>A Story . . . . .</b>	<b>53</b>
<b>4</b>	<b>Warehouse Management Process Control . .</b>	<b>57</b>
4.1	System Scope . . . . .	58
4.2	Warehouse Management Process Control . . .	60
<b>5</b>	<b>Baseline Architecture . . . . .</b>	<b>65</b>
5.1	Architecture Context . . . . .	66
5.2	Partitioning the Big Ball of Mud . . . . .	67
5.3	Decomposing the Layers . . . . .	68
5.4	Accessing Domain Object Functionality . . . .	71
5.5	Bridging the Network . . . . .	72
5.6	Separating User Interfaces . . . . .	76
5.7	Distributing Functionality . . . . .	79
5.8	Supporting Concurrent Domain Object Access	82
5.9	Achieving Scalable Concurrency . . . . .	85
5.10	Crossing the Object-Oriented/Relational Divide	87
5.11	Configuring Domain Objects at Runtime . . . .	89
5.12	Baseline Architecture Summary . . . . .	90
<b>6</b>	<b>Communication Middleware . . . . .</b>	<b>95</b>
6.1	A Middleware Architecture for Distributed Systems . . . . .	96
6.2	Structuring the Internal Design of the Middleware . . . . .	100
6.3	Encapsulating Low-level System Mechanisms .	103
6.4	Demultiplexing ORB Core Events . . . . .	105
6.5	Managing ORB Connections . . . . .	108
6.6	Enhancing ORB Scalability . . . . .	111
6.7	Implementing a Synchronized Request Queue	114
6.8	Interchangeable Internal ORB Mechanisms . .	116

6.9	Consolidating ORB Strategies . . . . .	118
6.10	Dynamic Configuration of ORBs . . . . .	121
6.11	Communication Middleware Summary . . . . .	124
<b>7</b>	<b>Warehouse Topology . . . . .</b>	<b>129</b>
7.1	Warehouse Topology Baseline . . . . .	130
7.2	Representing Hierarchical Storage . . . . .	131
7.3	Navigating the Storage Hierarchy . . . . .	133
7.4	Modeling Storage Properties . . . . .	135
7.5	Varying Storage Behavior . . . . .	137
7.6	Realizing Global Functionality . . . . .	140
7.7	Traversing the Warehouse Topology . . . . .	142
7.8	Supporting Control Flow Extensions . . . . .	144
7.9	Connecting to the Database . . . . .	146
7.10	Maintaining In-Memory Storage Data . . . . .	147
7.11	Configuring the Warehouse Topology . . . . .	149
7.12	Detailing the Explicit Interface . . . . .	151
7.13	Warehouse Topology Summary . . . . .	153
<b>8</b>	<b>The Story Behind The Pattern Story . . . . .</b>	<b>157</b>
<b>Part III</b>	<b>The Language . . . . .</b>	<b>163</b>
<b>9</b>	<b>From Mud To Structure . . . . .</b>	<b>167</b>
	Domain Model ** . . . . .	182
	Layers ** . . . . .	185
	Model-View-Controller ** . . . . .	188
	Presentation-Abstraction-Control . . . . .	191
	Microkernel ** . . . . .	194
	Reflection * . . . . .	197
	Pipes and Filters ** . . . . .	200
	Shared Repository ** . . . . .	202
	Blackboard . . . . .	205
	Domain Object ** . . . . .	208

<b>10</b>	<b>Distribution Infrastructure</b> . . . . .	211
	Messaging ** . . . . .	221
	Message Channel ** . . . . .	224
	Message Endpoint ** . . . . .	227
	Message Translator ** . . . . .	229
	Message Router ** . . . . .	231
	Publisher-Subscriber ** . . . . .	234
	Broker ** . . . . .	237
	Client Proxy ** . . . . .	240
	Requestor ** . . . . .	242
	Invoker ** . . . . .	244
	Client Request Handler ** . . . . .	246
	Server Request Handler ** . . . . .	249
<b>11</b>	<b>Event Demultiplexing and Dispatching</b> . . .	253
	Reactor ** . . . . .	259
	Proactor * . . . . .	262
	Acceptor-Connector ** . . . . .	265
	Asynchronous Completion Token ** . . . . .	268
<b>12</b>	<b>Interface Partitioning</b> . . . . .	271
	Explicit Interface ** . . . . .	281
	Extension Interface ** . . . . .	284
	Introspective Interface ** . . . . .	286
	Dynamic Invocation Interface * . . . . .	288
	Proxy ** . . . . .	290
	Business Delegate ** . . . . .	292
	Facade ** . . . . .	294
	Combined Method ** . . . . .	296
	Iterator ** . . . . .	298
	Enumeration Method ** . . . . .	300
	Batch Method ** . . . . .	302

<b>13</b>	<b>Component Partitioning</b> . . . . .	305
	Encapsulated Implementation ** . . . . .	313
	Whole-Part ** . . . . .	317
	Composite ** . . . . .	319
	Master-Slave * . . . . .	321
	Half-Object plus Protocol ** . . . . .	324
	Replicated Component Group * . . . . .	326
<b>14</b>	<b>Application Control</b> . . . . .	329
	Page Controller ** . . . . .	337
	Front Controller ** . . . . .	339
	Application Controller ** . . . . .	341
	Command Processor ** . . . . .	343
	Template View ** . . . . .	345
	Transform View ** . . . . .	347
	Firewall Proxy ** . . . . .	349
	Authorization ** . . . . .	351
<b>15</b>	<b>Concurrency</b> . . . . .	353
	Half-Sync/Half-Async ** . . . . .	359
	Leader/Followers ** . . . . .	362
	Active Object ** . . . . .	365
	Monitor Object ** . . . . .	368
<b>16</b>	<b>Synchronization</b> . . . . .	371
	Guarded Suspension ** . . . . .	380
	Future ** . . . . .	382
	Thread-Safe Interface * . . . . .	384
	Double-Checked Locking . . . . .	386
	Strategized Locking ** . . . . .	388
	Scoped Locking ** . . . . .	390
	Thread-Specific Storage . . . . .	392

	Copied Value **	394
	Immutable Value **	396
<b>17</b>	<b>Object Interaction</b>	<b>399</b>
	Observer **	405
	Double Dispatch **	408
	Mediator *	410
	Command **	412
	Memento **	414
	Context Object **	416
	Data Transfer Object **	418
	Message **	420
<b>18</b>	<b>Adaptation and Extension</b>	<b>423</b>
	Bridge **	436
	Object Adapter **	438
	Chain of Responsibility *	440
	Interpreter	442
	Interceptor **	444
	Visitor **	447
	Decorator	449
	Execute-Around Object **	451
	Template Method *	453
	Strategy **	455
	Null Object **	457
	Wrapper Facade **	459
	Declarative Component Configuration *	461
<b>19</b>	<b>Modal Behavior</b>	<b>463</b>
	Objects for States *	467
	Methods for States *	469
	Collections for States **	471

<b>20</b>	<b>Resource Management</b> . . . . .	473
	Container *	488
	Component Configurator *	490
	Object Manager **	492
	Lookup **	495
	Virtual Proxy **	497
	Lifecycle Callback **	499
	Task Coordinator *	501
	Resource Pool **	503
	Resource Cache **	505
	Lazy Acquisition **	507
	Eager Acquisition **	509
	Partial Acquisition *	511
	Activator **	513
	Evictor **	515
	Leasing **	517
	Automated Garbage Collection **	519
	Counting Handle **	522
	Abstract Factory **	525
	Builder *	527
	Factory Method **	529
	Disposal Method **	531
<b>21</b>	<b>Database Access</b> . . . . .	533
	Database Access Layer **	538
	Data Mapper **	540
	Row Data Gateway **	542
	Table Data Gateway **	544
	Active Record	546
<b>22</b>	<b>A Departing Thought</b> . . . . .	549

<b>Glossary</b>	<b>553</b>
<b>References</b>	<b>573</b>
<b>Index of Patterns</b>	<b>587</b>
<b>Index of Names</b>	<b>593</b>
<b>Subject Index</b>	<b>595</b>