

# Contents

Preface .....	xiii
Acknowledgments .....	xix
About the Author .....	xxi
Introduction .....	xxiii

## Part I: The Roots of Object-Oriented Development . . . . . 1

<b>Chapter 1: Historical Perspective.</b> .....	<b>3</b>
History .....	3
Structured Development .....	5
Functional Decomposition .....	7
Lessons Hard Won .....	12
Global Data .....	12
Elephantine Program Units .....	12
Software Structure .....	13
Lack of Cohesion .....	14
Coupling .....	14
Technical Innovation .....	17
The Turing Machine .....	17
Languages and Methodologies .....	18
Sets and Graphs .....	21
Normal Form (NF) .....	22
Data Flows .....	25
State Machines .....	27
<b>Chapter 2: Object Technology</b> .....	<b>29</b>
Basic Philosophy .....	30
Maintainability .....	30
Problem Space Abstraction .....	31
OOA, OOD, and OOP .....	32

Subject Matter . . . . .	33
Separation of Concerns . . . . .	34
Levels of Abstraction . . . . .	35
Problem Space Abstraction . . . . .	37
Encapsulation . . . . .	39
Cohesion . . . . .	40
Logical Indivisibility . . . . .	40
Communication Models . . . . .	42
Breadth-First Processing (aka Peer-to-Peer Collaboration) . . . . .	44
Elaboration versus Translation . . . . .	45
The Message Paradigm . . . . .	47
Object Characteristics . . . . .	49
<b>Chapter 3: Generalization, Inheritance, Genericity, and Polymorphism . . . . .</b>	<b>53</b>
Generalization . . . . .	54
Inheritance . . . . .	56
Polymorphism . . . . .	57
Inclusion (or Inherent) Polymorphism . . . . .	60
Genericity . . . . .	61
<b>Chapter 4: MBD Road Map . . . . .</b>	<b>63</b>
Problem Space versus Computing Space . . . . .	63
Problem Space . . . . .	67
Computing Space . . . . .	68
Transformation . . . . .	69
Maintainability . . . . .	69
Domain Analysis . . . . .	69
Modeling Invariants . . . . .	71
Partitioning the Application . . . . .	72
A Static View . . . . .	72
A Dynamic View . . . . .	73
<b>Chapter 5: Modeling Invariants . . . . .</b>	<b>77</b>
So Just What Is Modeling Invariants? . . . . .	78
The Invariant Side . . . . .	78
The Data Side . . . . .	80
The Rewards . . . . .	81
Examples . . . . .	84
Bank ATM Software . . . . .	84
Hardware Interface . . . . .	89

Depreciation .....	93
Remote POS Entry Example .....	99
<b>Chapter 6: Application Partitioning .....</b>	<b>105</b>
Why Do We Care? .....	105
Basic Concepts of Application Partitioning .....	107
Subject Matter .....	109
Client/Service Relationships .....	114
Levels of Abstraction .....	116
Interfaces .....	118
Identifying Subsystems .....	119
When abstracting the same entities as other subsystems, does the subsystem in hand have a different view of them? .....	119
Is there a client/service relationship? .....	119
Is the service more detailed than the client? .....	120
Is knowledge shared with other subsystems? .....	120
Is behavior shared with other subsystems? .....	120
Is the subject matter cohesive? .....	121
Is the boundary clearly defined? .....	121
Could it be reused as-is? .....	121
Is it bounded by a network in a distributed environment? .....	121
Is it a different problem space? .....	122
Bridges .....	122
The Message Paradigm, Yet Again .....	123
The Bridge Model .....	125
Describing Subsystems .....	127
Subsystem Descriptions .....	128
Relationship Descriptions .....	129
Requirements Allocation .....	130
An Example: Pet Care Center .....	130
Processes .....	145
Legacy Replacement: A Practical Example .....	147
<b>Part II: The Static Model .....</b>	<b>151</b>
<b>Chapter 7: Road Map to Part II .....</b>	<b>153</b>
What Is the Static Model? .....	154

Knowledge versus Behavior . . . . .	156
Practical Note . . . . .	158
<b>Chapter 8: Classes . . . . .</b>	<b>161</b>
Abstract Representation . . . . .	161
Model of Something Real . . . . .	162
Local to a Subsystem or Component. . . . .	164
Logical Indivisibility . . . . .	164
Delegation. . . . .	165
Class Notation . . . . .	167
Identifying Classes and Their Responsibilities . . . . .	169
Object Blitz . . . . .	169
Use Case Variant. . . . .	172
Examples. . . . .	172
The Legendary Bank ATM Controller. . . . .	173
Pet Care Center: Disposition . . . . .	181
Using Sequence and Collaboration Diagrams . . . . .	186
<b>Chapter 9: Class Responsibilities . . . . .</b>	<b>191</b>
Attributes: What the Objects of a Class Should Know. . . . .	191
Definitions and Notation . . . . .	191
Not the Same as Data Storing . . . . .	193
State . . . . .	194
Abstract Data Type. . . . .	195
Operations and Methods: What an Object Must Do . . . . .	197
Definitions and Notation . . . . .	199
Identifying Behaviors . . . . .	201
Anthropomorphizing . . . . .	206
Process . . . . .	207
Examples. . . . .	209
ATM. . . . .	209
Pet Care Center: Disposition. . . . .	222
<b>Chapter 10: Associations . . . . .</b>	<b>233</b>
Definitions and Basics . . . . .	234
Notation . . . . .	239
The Nature of Logical Connections. . . . .	242
Navigating to Knowledge versus Navigating to Behavior . . . . .	244

Association Roles . . . . .	244
Association Paths . . . . .	246
Conditionality . . . . .	250
Multiplicity . . . . .	255
Replacing a Problem Space “One-or-More” with a Model “One” . . . . .	256
Supplementing a Problem Space Association with a Second Association . . . . .	257
Selection versus Participation . . . . .	260
Constraints . . . . .	261
Association Classes . . . . .	264
Reification . . . . .	267
Identifying Associations . . . . .	269
Examples . . . . .	273
The ATM Controller . . . . .	273
Pet Care Center: Disposition . . . . .	277
<b>Chapter 11: Referential and Knowledge Integrity . . . . .</b>	<b>279</b>
Knowledge Integrity . . . . .	280
Timeliness . . . . .	280
Consistency . . . . .	281
Snapshots versus Immediate Access . . . . .	282
Dependent Attributes . . . . .	283
Normalization of Attributes . . . . .	285
Referential Integrity . . . . .	289
Identity and Referential Attributes . . . . .	290
Association Loops . . . . .	292
Relational versus OO Paradigms: Worlds Apart . . . . .	295
<b>Chapter 12: Generalization Redux . . . . .</b>	<b>299</b>
Subclassing . . . . .	300
Notation and Rules . . . . .	300
Generalization and Specialization . . . . .	304
Categorization . . . . .	307
Inclusion Polymorphism . . . . .	312
Why {disjoint, complete} Subsets? . . . . .	315
Multi-directional Subclassing, Multiple Inheritance, and Composition . . . . .	317
Liskov Substitution Principle . . . . .	323



- Alternatives to Generalization . . . . . 328
  - Delegation. . . . . 329
  - Parametric Polymorphism . . . . . 331
  - Basic Abstraction . . . . . 332
- Chapter 13: Identifying Knowledge . . . . . 333**
  - What Is the Nature of OO Knowledge? . . . . . 334
  - Abstracting Aggregates . . . . . 335
  - Picking the Right Abstraction . . . . . 341
    - Abstracting What People Do . . . . . 343
    - What Does the Underlying Entity Know? . . . . . 344
    - What Subset of Entity Knowledge Should the Object Abstraction Know? . . . . . 345
    - What Is the Subject Matter? . . . . . 345
    - What Is the Level of Abstraction? . . . . . 350
  - Does the Abstraction Need to Coalesce Entity Knowledge? . . . . . 351
  
- Part III: The Dynamic Model . . . . . 355**
- Chapter 14: Road Map to Part III . . . . . 357**
  - Part III Road Map . . . . . 357
    - It's All about Behavior . . . . . 358
    - Object Life Cycles . . . . . 362
    - Asynchronous Solution . . . . . 363
    - Synchronous Services . . . . . 369
  - Action Languages . . . . . 373
  - Mealy versus Moore versus Harel . . . . . 374
  - The Learning Curve. . . . . 376
- Chapter 15: The Finite State Machine . . . . . 377**
  - Basic Finite State Automata . . . . . 378
    - Notation . . . . . 383
    - The Role of Design by Contract (DbC) . . . . . 387
  - Looking for State Machines. . . . . 390
    - Knowledge versus Behavior. . . . . 391
    - Managing Sequences . . . . . 395
  - Some Examples . . . . . 407
    - Garage Door Opener . . . . . 407

Automated Home Kitchen . . . . .	409
Air Traffic Control System . . . . .	410
Rock . . . . .	411
<b>Chapter 16: States, Transitions, Events, and Actions. . . . .</b>	<b>415</b>
States. . . . .	415
Transitions . . . . .	421
Events . . . . .	423
Actions . . . . .	427
The Execution Model . . . . .	430
Naming Conventions . . . . .	433
<b>Chapter 17: Developing State Models . . . . .</b>	<b>437</b>
Designing State Machines . . . . .	437
Examples. . . . .	450
ATM Controller: Character Display . . . . .	451
ATM Controller: Dispatcher. . . . .	457
ATM Controller: Deposit . . . . .	468
<b>Chapter 18: Abstract Action Languages . . . . .</b>	<b>475</b>
AALs and ADFDs . . . . .	476
AAL Syntax. . . . .	478
Examples. . . . .	480
Garage Door Opener . . . . .	480
ATM: Character Display. . . . .	484
<b>Glossary . . . . .</b>	<b>489</b>
<b>Index . . . . .</b>	<b>501</b>