

Contents

Preface	xiii
Acknowledgements	xvii
About the Author	xix
Chapter 1 Introduction to UML	1
1.1 UML Basic Modeling Concepts	1
1.2 Structural Elements and Diagrams	4
1.2.1 Small Things: Objects, Classes, and Interfaces.....	4
1.2.2 Relations.....	9
1.2.3 Big Things: Subsystems, Components, and Packages	15
1.3 Behavioral Elements and Diagrams.....	18
1.3.1 Actions and Activities.....	18
1.3.2 Operations and Methods.....	19
1.3.3 Activity Diagrams.....	20
1.3.4 State Diagrams	21
1.3.5 Interactions	26
1.4 Use Case and Requirements Models	28
1.5 Summary.....	32
Chapter 2 The Harmony Process	33
2.1 Introduction	33
2.2 The Harmony Development Process	34
2.2.1 Why a Process?	34
2.2.2 Harmony Process Overview	39
2.3 The Systems Engineering Harmony Workflows in Detail	44
2.3.1 System Functional Analysis	44
2.3.2 Build Use Case Model Workflow	46
2.3.3 Design Syntheses: System Architectural Design	46
2.4 The Hand-off from Systems Engineering.....	46
2.5 The Software Workflows in Detail.....	50
2.5.1 Analysis with the Harmony Process.....	50
2.5.2 Design with the Harmony Process	57
2.5.3 Verification and Validation (V&V).....	62
2.5.4 Increment Review (Party!) Workflow.....	63
2.6 Summary.....	65

Chapter 3 Meeting Industry Standards.....	67
3.1 Overview.....	67
3.2 On the Importance of Being Standard	67
3.3 Architectural Framework Standards (I'm looking at you UPDM)	68
3.3.1 UPDM	72
3.4 IEC 61508.....	72
3.4.1 Process Requirements	74
3.5 DO-178B/C.....	76
3.5.1 Verification	76
3.5.2 DO-178C	79
3.5.3 Tool Qualification.....	82
3.5.4 Complying with DO-178B/C Using the Harmony Process.....	83
3.6 IEC 62304.....	84
3.7 CMMI-DEV.....	84
3.8 Summary.....	87
References.....	88
Chapter 4 Specifying Requirements.....	89
4.1 Overview.....	89
4.2 Representing Requirements in UML and SysML	90
4.3 Specification View: State Machines for Requirements Capture	101
4.3.1 Developing the State Machine in the Harmony Nanocycle	102
References.....	109
Chapter 5 Systems Architecture: Deployment and Subsystems Architecture.....	111
5.1 Overview.....	111
5.2 The Hand-off from Systems to Downstream Engineering	136
5.3 Looking Ahead	146
Chapter 6 Dependability Architecture.....	149
6.1 Overview.....	149
6.2 A (Not-So) Quick Note about Design Patterns.....	152
6.2.1 Construct the Initial Model	153
6.2.2 Identify Important Design Criteria.....	153
6.2.3 Rank Design Criteria.....	154
6.2.3 Select Design Patterns and Technologies.....	154
6.2.4 Apply Design Patterns and Technologies.....	155
6.2.5 Validate Design Solution	155
6.3 What is a Design Pattern?.....	155
6.3.1 Basic Structure of Design Patterns.....	156
6.3.2 Using Design Patterns in Development.....	157
6.4 Summary.....	175
Chapter 7 High-Fidelity Modeling.....	179
7.1 Overview.....	179
7.2 A Quick Note about Structured Design with UML	180

7.3 High-Fidelity Modeling Workflow	181
7.4 Key Strategies for Object Identification	182
7.4.1 Underline the Nouns	182
7.4.2 Identify the Causal Agents.....	184
7.4.3 Identify Services (Passive Contributors or Server Objects)	184
7.4.4 Identify Messages and Information Flows	184
7.4.5 Identify Real-World Items	184
7.4.6 Identify Physical Devices	185
7.4.7 Identify Key Concepts	185
7.4.8 Identify Transactions	185
7.4.9 Identify Persistent Information.....	185
7.4.10 Identify Visual Elements.....	186
7.4.11 Identify Control Elements	186
7.4.12 Apply Scenarios.....	186
7.5 Summary.....	216
Chapter 8 Distribution Architecture.....	219
8.1 Overview.....	219
8.2 Summary.....	223
Chapter 9 Concurrency and Resource Architecture	225
9.1 What is the Concurrency and Resource Architecture?	225
9.2 Harmony Concurrency and Resource Architecture Workflow.....	234
9.3 Summary.....	241
Chapter 10 Collaboration and Detailed Design	243
10.1 Overview.....	243
10.2 Collaboration Design	244
10.2.1 Delegation Pattern Strategy	247
10.2.2 Interface Abstraction Pattern Strategy	247
10.3 Detailed Design	251
10.4 Summary.....	274
Chapter 11 Specifying Requirements: Answers.....	277
11.1 Answer 4.1: Identifying Kinds of Requirements.....	277
11.2 Answer 4.2: Identifying Use Cases for the Roadrunner Traffic Light Control System.....	277
11.2.1 Additional Questions.....	280
11.3 Answer 4.3: Mapping Requirements to Use Cases	280
11.4 Answer 4.4: Identifying Use Cases for the Coyote UAV System.....	281
11.4.1 Additional Questions.....	282
11.5 Answer 4.5: Create a Requirements Table.....	283
11.6 Answer 4.6: Capturing Quality of Service Requirements	283
11.7 Answer 4.7: Operational View: Identifying Traffic Light Scenarios	284
11.8 Answer 4.8: Operational View: Coyote UAV Optical Surveillance Scenarios.....	291

11.9	Answer 4.9: Specification View: Use Case Descriptions	294
11.10	Answer 4.10: Simple State Machine Specification.....	295
11.11	Answer 4.11: Specification View: Capturing Complex Requirements	296
11.11.1	Step 1: No Cars or Pedestrians Are Considered (Just Get Through Traffic Working)	299
11.11.2	Step 2: Add Turn Lanes.....	302
11.11.3	Step 3: Add Pedestrians	305
11.11.4	Additional Questions.....	305
11.12	Answer 4.12: Operational to Specification View: Capturing Operational Contracts	305
11.12.1	Step 1: Draw the Block Diagram	306
11.12.2	Step 2: Define the Interfaces	308
11.12.3	Step 3: Defining Pre- and Postconditions for the Interfaces	308
11.12.4	Step 4: Construct a Use-Case Activity Diagram Representing All of The Scenarios Previously Specified for the Use Case	309
11.12.5	Step 5: Define a State Machine for the Use Case Block that is Consistent with the Set of Scenarios	309
	References.....	312
	Chapter 12 Deployment and Subsystems Architecture: Answers.....	313
12.1	Answer 5.1: Organizing the Systems Model	313
12.1.1	The Roadrunner Model.....	313
12.1.2	The CUAV Suite of Models.....	313
12.2	Answer 5.2: Subsystem Identification	316
12.3	Answer 5.3: Mapping Operational Contracts into the Subsystem Architecture.....	323
12.4	Answer 5.4: Identifying Subsystem Use Cases	331
12.5	Answer 5.5: Creating the Shared Model.....	337
12.6	Answer 5.6: Initiating the Subsystem Model.....	340
	Chapter 13 Dependability Architecture: Answers	349
13.1	Answer 6.1: Safety Architecture	349
13.2	Answer 6.2: Reliability Architecture	360
13.3	Answer 6.3: Security Architecture	361
	Chapter 14 High-Fidelity Modeling: Answers.....	369
14.1	Answer 7.1: Apply Nouns and Causal Agents Strategies	369
14.2	Answer 7.2: Apply Services and Messages Strategies.....	382
14.3	Answer 7.3: Apply the Strategies with a Test-Driven Development Approach.....	385
14.3.1	Step 1: Get the Light Working.....	387
14.3.2	Step 2: Add a Controller and Step a Single Light through its States.....	387
14.3.3	Step 3: Add Through Lights for Both Primary and Secondary Roads and Their Control	388

14.3.4 Step 3.1: Convert TrafficController State Machine to And-States	390
14.3.5 Step 4: Add Car Sensor for Turn Lanes.....	392
14.3.6 Step 5: Add the Second Turn Lane	393
14.3.7 Step 6: Add Pedestrian Buttons and Identify Where Processing of Pedestrian Requests Will Be Handled.....	394
14.3.8 Step 7: Handle Pedestrian Requests	396
14.3.9 Test-Driven Development for Coyote UAV.....	399
Chapter 15 Distribution Architecture: Answers.....	411
15.1 Answer 8.1: Roadrunner Distribution Architecture	411
15.2 Answer 8.2: Coyote UAV Distribution Architecture	411
Chapter 16 Concurrency and Resource Architecture: Answers.....	419
16.1 Answer 9.1: Roadrunner Concurrency and Resource Architecture	419
16.2 Answer 9.2: Reconnaissance Concurrency and Resource Architecture	419
Chapter 17 Collaboration and Detailed Design: Answers	423
17.1 Answer 10.1: Applying Collaboration Design Patterns: Part 1	423
17.2 Answer 10.2: Applying Collaboration Design Patterns: Part 2	424
17.3 Answer 10.3: Applying Detailed Design State Behavioral Patterns.....	429
17.3.1 Additional Work.....	432
17.4 Answer 10.4: Applying Detailed Design Idioms	434
Appendix A: The Roadrunner™ Intersection Controller System Specification.....	443
Appendix B: The Coyote Unmanned Aerial Vehicle System (CUAVS)	455
Appendix C: UML Notational Summary	467
Index.....	491