Contents

1 The Big Idea  
1.1 The Power of Analogy: The Architecture of Buildings, 2  
1.1.1 Limitations of the Analogy, 5  
1.1.2 So, What's the Big Idea?, 6  
1.2 The Power and Necessity of Big Ideas: The Architecture of the Web, 7  
1.3 The Power of Architecture in the Small: Architecture on the Desktop, 12  
1.4 The Power of Architecture in Business: Productivity and Product Lines, 15  
1.5 End Matter, 20  
1.6 Review Questions, 21  
1.7 Exercises, 21  
1.8 Further Reading, 21

2 Architectures in Context: The Reorientation of Software Engineering  
2.1 Fundamental Understandings, 24  
2.2 Requirements, 25  
2.3 Design, 29  
2.3.1 Design Techniques, 31  
2.4 Implementation, 33  
2.4.1 Implementation Strategies, 35  
2.5 Analysis and Testing, 38  
2.6 Evolution and Maintenance, 40  
2.7 Processes, 42  
2.7.1 The Turbine Visualization, 43  
2.7.2 Example Process Depictions, 49  
2.8 End Matter, 52  
2.9 Review Questions, 53  
2.10 Exercises, 54  
2.11 Further Reading, 55

3 Basic Concepts  
3.1 Terminology, 58  
3.1.1 Architecture, 58  
3.1.2 Component, 68  
3.1.3 Connector, 70  
3.1.4 Configuration, 72  
3.1.5 Architectural Style, 72  
3.1.6 Architectural Pattern, 73
3.2 Models, 75
3.3 Processes, 75
3.4 Stakeholders, 79
3.5 End Matter, 79
3.6 Review Questions, 80
3.7 Exercises, 80
3.8 Further Reading, 81

4 Designing Architectures
4.1 The Design Process, 85
4.2 Architectural Conception, 87
  4.2.1 Fundamental Conceptual Tools, 87
  4.2.2 The Grand Tool: Refined Experience, 90
4.3 Refined Experience in Action: Styles and Architectural Patterns, 91
  4.3.1 Domain Specific Software Architectures, 93
  4.3.2 Architectural Patterns, 94
  4.3.3 Introduction to Styles, 99
  4.3.4 Simple Styles, 102
  4.3.5 More Complex Styles, 124
  4.3.6 Discussion: Patterns and Styles, 137
  4.3.7 Design Recovery, 139
4.4 Architectural Conception in Absence of Experience:
  Unprecedented Design, 144
4.5 Putting it All Together: Design Processes Revisited, 149
  4.5.1 Insights from Requirements, 150
  4.5.2 Insights from Implementation, 151
4.6 End Matter, 152
4.7 Review Questions, 153
4.8 Exercises, 154
4.9 Further Reading, 154

5 Connectors
5.1 Connectors in Action: A Motivating Example, 159
5.2 Connector Foundations, 161
5.3 Connector Roles, 163
5.4 Connector Types and Their Variation Dimensions, 164
  5.4.1 Procedure Call Connectors, 165
  5.4.2 Event Connectors, 166
  5.4.3 Data Access Connectors, 167
  5.4.4 Linkage Connectors, 168
  5.4.5 Stream Connectors, 169
  5.4.6 Arbitrator Connectors, 170
  5.4.7 Adaptor Connectors, 170
  5.4.8 Distributor Connectors, 171
5.5 Example Connectors, 172
  5.5.1 Event-Based Data Distribution Connectors, 173
  5.5.2 Grid-Based Data Distribution Connectors, 174
  5.5.3 Client-Server–Based Data Distribution Connectors, 175
  5.5.4 P2P-Based Data Distribution Connectors, 176
5.6 Using the Connector Framework, 176
  5.6.1 Selecting Appropriate Connectors, 177
  5.6.2 Detecting Mismatches, 180
5.7 End Matter, 181
5.8 Review Questions, 182
5.9 Exercises, 183
5.10 Further Reading, 183

6 Modeling
  6.1 Modeling Concepts, 186
    6.1.1 Stakeholder-Driven Modeling, 186
    6.1.2 Basic Architectural Concepts, 188
    6.1.3 Elements of the Architectural Style, 189
    6.1.4 Static and Dynamic Aspects, 190
    6.1.5 Functional and Non-Functional Aspects, 191
  6.2 Ambiguity, Accuracy, and Precision, 191
    6.2.1 Ambiguity, 192
    6.2.2 Accuracy and Precision, 192
  6.3 Complex Modeling: Mixed Content and Multiple Views, 194
    6.3.1 Views and Viewpoints, 194
    6.3.2 Consistency among Views, 196
  6.4 Evaluating Modeling Techniques, 198
  6.5 Specific Modeling Techniques, 199
    6.5.1 Generic Techniques, 199
    6.5.2 Early Architecture Description Languages, 209
    6.5.3 Domain- and Style-Specific ADLs, 221
    6.5.4 Extensible ADLs, 229
  6.6 When Systems Become Too Complex to Model, 242
  6.7 End Matter, 243
  6.8 Review Questions, 246
  6.9 Exercises, 246
  6.10 Further Reading, 246

7 Visualization
  7.1 Visualization Concepts, 250
    7.1.1 Canonical Visualizations, 250
    7.1.2 Textual Visualizations, 251
    7.1.3 Graphical Visualizations, 253
    7.1.4 Hybrid Visualizations, 255
    7.1.5 The Relationship between Visualizations and Views, 255
7.2 Evaluating Visualizations, 259
  7.2.2 Constructing a Visualization, 261
  7.2.3 Coordinating Visualizations, 264
  7.2.4 Beyond Design: Using Visualization Dynamically, 266
7.3 Common Issues in Visualization, 268
  7.3.1 Same Symbol, Different Meaning, 268
  7.3.2 Differences without Meaning, 269
  7.3.3 Decorations without Meaning, 269
  7.3.4 Borrowed Symbol, Different Meaning, 270
7.4 Evaluating Visualization Techniques, 272
7.5 Techniques, 272
  7.5.1 Textual Visualizations, 272
  7.5.2 Informal Graphical Editors, 274
  7.5.3 UML: The Unified Modeling Language, 276
  7.5.4 Rapide, 280
  7.5.5 The Labeled Transition State Analyzer (LTSA), 282
  7.5.6 xADL 2.0, 284
7.6 End Matter, 288
7.7 Review Questions, 289
7.8 Exercises, 290
7.9 Further Reading, 290

8 Analysis
8.1 Analysis Goals, 295
  8.1.1 Completeness, 295
  8.1.2 Consistency, 296
  8.1.3 Compatibility, 302
  8.1.4 Correctness, 303
8.2 Scope of Analysis, 303
  8.2.1 Component- and Connector-Level Analysis, 304
  8.2.2 Subsystem- and System-Level Analysis, 305
  8.2.3 Data Exchanged in the System or Subsystem, 307
  8.2.4 Architectures at Different Abstraction Levels, 308
  8.2.5 Comparison of Two or More Architectures, 310
8.3 Architectural Concern being Analyzed, 310
8.4 Level of Formality of Architectural Models, 312
8.5 Type of Analysis, 313
8.6 Level of Automation, 314
8.7 System Stakeholders, 315
8.8 Analysis Techniques, 317
  8.8.1 Inspections and Reviews, 317
  8.8.2 Model-Based Analysis, 322
  8.8.3 Simulation-Based Analysis, 328
8.9 End Matter, 333
8.10 Review Questions, 334
9 Implementation

9.1 Concepts, 338
  9.1.1 The Mapping Problem, 338
  9.1.2 Architecture Implementation Frameworks, 340
  9.1.3 Evaluating Frameworks, 343
  9.1.4 Middleware, Component Models, and Application Frameworks, 343
  9.1.5 Building a New Framework, 346
  9.1.6 Concurrency, 347
  9.1.7 Generative Technologies, 348
  9.1.8 Ensuring Architecture-to-Implementation Consistency, 349

9.2 Existing Frameworks, 350
  9.2.1 Frameworks for the Pipe-and-Filter Architectural Style, 350
  9.2.2 Frameworks for the C2 Architectural Style, 352

9.3 Examples, 360
  9.3.1 Implementing Lunar Lander in the Pipe-and-Filter Style Using the java.io Framework, 360
  9.3.2 Implementing Lunar Lander in the C2-Style Using the Lightweight C2 Framework, 366

9.4 End Matter, 377
9.5 Review Questions, 378
9.6 Exercises, 378
9.7 Further Reading, 379

10 Deployment and Mobility

10.1 Overview of Deployment and Mobility Challenges, 385
10.2 Software Architecture and Deployment, 387
  10.2.1 Basic Concepts, 388
  10.2.2 Deployment Activities, 388
  10.2.3 Tool Support, 402

10.3 Software Architecture and Mobility, 405
  10.3.1 Basic Concepts, 405
  10.3.2 Mobility Paradigms, 405
  10.3.3 Challenges in Migrating Code, 406

10.4 End Matter, 408
10.5 Review Questions, 409
10.6 Exercises, 409
10.7 Further Reading, 410

11 Applied Architectures and Styles

11.1 Distributed and Networked Architectures, 414
  11.1.1 Limitations of the Distributed Systems Viewpoint, 415
11.2 Architectures for Network-Based Applications, 416
11.2.1 The REpresentational State Transfer Style (REST), 416
11.2.2 Commercial Internet-Scale Applications, 422
11.3 Decentralized Architectures, 424
   11.3.1 Shared Resource Computation: The Grid World, 425
   11.3.2 Peer-to-Peer Styles, 426
   11.3.3 Summary Notes on Latency and Agency, 432
11.4 Service-Oriented Architectures and Web Services, 433
11.5 Architectures from Specific Domains, 437
   11.5.1 Robotics, 437
   11.5.2 Wireless Sensor Networks, 443
11.6 End Matter, 443
11.7 Review Questions, 445
11.8 Exercises, 445
11.9 Further Reading, 446

12 Designing for Non-Functional Properties 447
12.1 Efficiency, 450
   12.1.1 Software Components and Efficiency, 451
   12.1.2 Software Connectors and Efficiency, 453
   12.1.3 Architectural Configurations and Efficiency, 455
12.2 Complexity, 459
   12.2.1 Software Components and Complexity, 460
   12.2.2 Software Connectors and Complexity, 463
   12.2.3 Architectural Configurations and Complexity, 464
12.3 Scalability and Heterogeneity, 467
   12.3.1 Software Components and Scalability, 468
   12.3.2 Software Connectors and Scalability, 471
   12.3.3 Architectural Configurations and Scalability, 473
12.4 Adaptability, 475
   12.4.1 Software Components and Adaptability, 475
   12.4.2 Software Connectors and Adaptability, 476
   12.4.3 Architectural Configurations and Adaptability, 477
12.5 Dependability, 478
   12.5.1 Software Components and Dependability, 480
   12.5.2 Software Connectors and Dependability, 480
   12.5.3 Architectural Configurations and Dependability, 481
12.6 End Matter, 483
12.7 Review Questions, 484
12.8 Exercises, 484
12.9 Further Reading, 485

13 Security and Trust 487
13.1 Security, 489
13.2 Design Principles, 492
13.3 Architectural Access Control, 497
CONTENTS

13.3.1 Access Control Models, 498
13.3.2 Connector-Centric Architectural Access Control, 499

13.4 Trust Management, 508
13.4.1 Trust, 509
13.4.2 Trust Model, 510
13.4.3 Reputation-Based Systems, 510
13.4.4 Architectural Approach to Decentralized Trust Management, 514

13.5 End Matter, 522
13.6 Review Questions, 523
13.7 Exercises, 523
13.8 Further Reading, 524

14 Architectural Adaptation 525
14.1 Concepts of Architecture-Centric Adaptation, 526
14.1.1 Sources and Motivations for Change, 526
14.1.2 Shearing Layers, 529
14.1.3 Structural Elements Subject to Change, 532
14.1.4 Change Agents and Context, 535
14.1.5 Architecture: The Central Abstraction, 539

14.2 A Conceptual Framework for Architectural Adaptation, 539
14.3 Techniques for Supporting Architecture-Centric Change, 542
14.3.1 Basic Techniques Corresponding to Activities of the Conceptual Framework, 542
14.3.2 Architectures/Styles that Support Adaptation, 551
14.3.3 The Special Problems of On-the-Fly and Autonomous Adaptation, 557

14.4 End Matter, 561
14.5 Review Questions, 562
14.6 Exercises, 563
14.7 Further Reading, 563

15 Domain-Specific Software Engineering 565
15.1 Domain-Specific Software Engineering in a Nutshell, 570
15.1.1 Similar Problems, Similar Solutions, 570
15.1.2 Viewing DSSE through the Prism of Domain, Business, and Technology, 573

15.2 Domain-Specific Software Architecture, 574
15.2.1 Domain Knowledge, 574
15.2.2 Canonical Requirements, 585
15.2.3 Canonical Solution Strategies—Reference Architectures, 588
15.2.4 Product Lines and Architecture, 591
15.2.5 Product-Line Concepts, 592
15.2.6 Specifying the Architecture of a Product Line, 594
15.2.7 Capturing Variations Over Time, 559
15.2.8 Using Product Lines as Tools for What-If Analysis, 600
15.2.9 Implementing Product Lines, 602
15.2.10 Unifying Product Architectures with Different Intellectual Heritage, 604
15.2.11 Organizational Issues in Creating and Managing Product Lines, 606
15.3 DSSAs, Product Lines, and Architectural Styles, 606
15.4 DSSE Examples, 608
15.4.1 Koala and Consumer Electronics, 608
15.4.2 Software-Defined Radios, 611
15.5 End Matter, 616
15.6 Review Questions, 618
15.7 Exercises, 618
15.8 Further Reading, 619

16 Standards 621
16.1 What Are Standards?, 622
16.1.1 Why Use Standards?, 623
16.1.2 Drawbacks of Standards, 624
16.1.3 When to Adopt, 625
16.2 Specific Standards, 627
16.2.1 Conceptual Standards, 627
16.2.2 Notational Standards, 639
16.2.3 SysML, 641
16.2.4 Standard Tools, 645
16.2.5 Telelogic System Architect, 647
16.3 Process Standards, 648
16.3.1 Rational Unified Process, 648
16.3.2 Model-Driven Architecture, 650
16.4 End Matter, 651
16.5 Review Questions, 653
16.6 Exercises, 653
16.7 Further Reading, 654

17 People, Roles, and Teams 655
17.1 Who Are Software Architects?, 657
17.1.1 Architect as a Software Designer, 658
17.1.2 Architect as a Domain Expert, 658
17.1.3 Architect as a Software Technologist, 659
17.1.4 Architect as a Standards Compliance Expert, 660
17.1.5 Architect as a Software Engineering Economist, 661
17.1.6 Some Bad Habits, 661
17.2 What Do Software Architects Do?, 663
17.2.1 Develop Project Strategy, 663
17.2.2 Design Systems, 664
17.2.3 Communicate with Stakeholders, 664
17.2.4 Lead, 664
17.3 How Do Software Architects Work?, 665