Domain Architectures
Models and Architectures for UML Applications

Daniel J. Duffy
Datasim Education BV, Amsterdam, Netherlands

John Wiley & Sons, Ltd
Domain Architectures
Domain Architectures
Models and Architectures for UML Applications

Daniel J. Duffy
Datasim Education BV, Amsterdam, Netherlands

John Wiley & Sons, Ltd
Contents

Preface

PART I  Background and fundamentals

1.  Introducing and motivating domain architectures
2.  Domain architecture catalogue
3.  Software lifecycle and Datasim Development Process (DDP)
<table>
<thead>
<tr>
<th>3.4</th>
<th>The requirements/architecture phase in detail</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>The object-oriented analysis process</td>
<td>30</td>
</tr>
<tr>
<td>3.6</td>
<td>Project cultures and DDP</td>
<td>33</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Calendar-driven projects</td>
<td>34</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Requirements-driven projects</td>
<td>34</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Documentation-driven style</td>
<td>35</td>
</tr>
<tr>
<td>3.6.4</td>
<td>Quality-driven style</td>
<td>36</td>
</tr>
<tr>
<td>3.6.5</td>
<td>Architecture-driven style</td>
<td>36</td>
</tr>
<tr>
<td>3.6.6</td>
<td>Process-driven style and the DDP</td>
<td>37</td>
</tr>
<tr>
<td>3.7</td>
<td>Summary and conclusions</td>
<td>38</td>
</tr>
</tbody>
</table>

4. **Fundamental concepts and documentation issues** | 41
| 4.1 | Introduction and objectives                | 41 |
| 4.2 | How we document domain architectures       | 43 |
| 4.3 | Characteristics of ISO 9126 and its relationship with domain architectures | 44 |
| 4.4 | Documenting high-level artefacts           | 48 |
| 4.5 | Goals and core processes                   | 48 |
| 4.6 | System context                             | 50 |
| 4.7 | Stakeholders and viewpoints                | 50 |
| 4.7.1 | Documenting viewpoints                     | 52 |
| 4.8 | Documenting requirements                   | 54 |
| 4.9 | Defining and documenting use cases         | 54 |
| 4.10 | Summary and conclusions                    | 55 |

Appendix 4.1: A critical look at use cases | 55

PART II  **Domain architectures (meta models)** | 57

5. **Management Information Systems (MIS)** | 59
| 5.1 | Introduction and objectives                | 59 |
| 5.2 | Background and history                     | 59 |
| 5.3 | Motivational examples                      | 61 |
| 5.3.1 | Simple Digital Watch (SDW)                 | 61 |
| 5.3.2 | Instrumentation and control systems        | 62 |
| 5.4 | General applicability                      | 63 |
| 5.5 | Goals, processes and activities            | 64 |
| 5.6 | Context diagram and system decomposition   | 65 |
| 5.7 | Stakeholders, viewpoints and requirements  | 67 |
| 5.8 | UML classes                                | 69 |
| 5.9 | Use cases                                  | 70 |
5.10 Specializations of MIS systems 71
      5.10.1 Example: Noise control engineering 72
5.11 Using MIS systems with other systems 74
5.12 Summary and conclusions 76

6. Process Control Systems (PCS) 77
   6.1 Introduction and objectives 77
   6.2 Background and history 78
   6.3 Motivational examples 78
      6.3.1 Simple water level control 79
      6.3.2 Bioreactor 80
      6.3.3 Barrier options 81
   6.4 Reference models for Process Control Systems 83
      6.4.1 Basic components and variables 83
      6.4.2 Control engineering fundamentals 86
   6.5 General applicability 88
   6.6 Goals, processes and activities 89
   6.7 Context diagram and system decomposition 90
      6.7.1 Decomposition strategies 91
   6.8 Stakeholders, viewpoints and requirements 96
      6.8.1 Input and output variable completeness 97
      6.8.2 Robustness criteria 97
      6.8.3 Timing 98
      6.8.4 Human–Computer Interface (HCI) criteria 100
      6.8.5 State completeness 100
      6.8.6 Data age requirement 101
   6.9 UML classes 101
   6.10 Use cases 102
   6.11 Specializations of PCS systems 105
      6.11.1 Multi-level architectures 105
   6.12 Using PCS systems with other systems 106
   6.13 Summary and conclusions 107
Appendix 6.1: Message patterns in Process Control Systems 108

7. Resource Allocation and Tracking (RAT) systems 111
   7.1 Introduction and objectives 111
   7.2 Background and history 112
   7.3 Motivational examples 112
      7.3.1 Help Desk System (HDS) 113
      7.3.2 Discrete manufacturing 115
   7.4 General applicability 117
7.5 Goals, processes and activities 118  
7.6 Context diagram and system decomposition 118  
7.7 Stakeholders, viewpoints and requirements 120  
7.8 UML classes 121  
7.9 Use cases 123  
7.10 Specializations of RAT systems 124  
7.11 Using RAT systems with other systems 125  
7.12 Summary and conclusions 126  

8. **Manufacturing (MAN) systems** 127  
8.1 Introduction and objectives 127  
8.2 Background and history 128  
8.3 Motivational examples 130  
8.3.1 Compiler theory 130  
8.3.2 Graphics applications 132  
8.3.3 Human memory models 134  
8.4 General applicability 137  
8.5 Goals, processes and activities 138  
8.6 Context diagram and system decomposition 138  
8.7 Stakeholders, viewpoints and requirements 139  
8.7.1 Stakeholders and viewpoints 139  
8.7.2 Requirements 140  
8.8 UML classes 141  
8.9 Use cases 142  
8.10 Specializations of MAN systems 143  
8.11 Using MAN systems with other systems 144  
8.12 Summary and conclusions 144  

9. **Access Control Systems (ACS)** 147  
9.1 Introduction and objectives 147  
9.2 Background and history 148  
9.3 Motivational examples 148  
9.3.1 The Reference Monitor model 148  
9.4 General applicability 152  
9.5 Goals, processes and activities 152  
9.6 Context diagram and system decomposition 153  
9.7 Stakeholders, viewpoints and requirements 154  
9.8 UML classes 155  
9.9 Use cases 155
9.10 Specializations of ACS
  9.10.1 Security models for Web-based applications
  9.10.2 Access control during design: the Proxy pattern

9.11 Using ACS with other systems

10. Lifecycle and composite models
  10.1 Introduction and objectives
  10.2 Background and history
  10.3 Motivational example: the Rent-a-machine system
  10.4 General applicability
  10.5 Goals, processes and activities
  10.6 Context diagram and system decomposition
  10.7 Stakeholders, viewpoints and requirements
  10.8 UML classes
  10.9 Use cases
  10.10 Specializations of LCM
  10.11 Using LCM systems with other systems
  10.12 Summary and conclusions

PART III Applications (models)

11. Project resource management system: Manpower Control (MPC) system
  11.1 Introduction and objectives
  11.2 Description and scope of problem
  11.3 Core processing and context diagram
  11.4 Requirements and use case analysis
     11.4.1 Functional requirements and use cases
     11.4.2 Non-functional requirements
  11.5 Validating use cases
  11.6 Class architecture
  11.7 Generalizations
  11.8 Summary and conclusions

12. Home Heating System (HHS)
  12.1 Introduction and objectives
  12.2 Background and history
     12.2.1 Hatley–Pirbhai
     12.2.2 The Booch approach
12.3 Description of problem
12.4 Goals, processes and context
12.5 System decomposition and PAC model
12.6 Viewpoints and requirements analysis
12.7 Use cases
12.8 Validation efforts
12.9 Creating statecharts
12.10 Generalization efforts
12.11 Summary and conclusions

13. Elevator Control System (ELS)

13.1 Introduction and objectives
13.2 Domain categories and ELS
13.3 A traditional object-oriented requirement specification
13.4 Re-engineering ELS: goals and processes
13.5 Stakeholders and their requirements
13.6 Requirements
13.7 System decomposition of ELS
13.8 PAC decomposition of ELS
13.9 Major use cases
  13.9.1 Normal use cases
  13.9.2 Exceptional use cases
13.10 Summary and conclusions
Appendix 13.1: Definitions


14.1 Introduction and objectives
14.2 Customer Requirements Specification (CRS): the product management vision of OPS
  14.2.1 Business concerns and stakeholders’ viewpoints
14.3 OPS as a lifecycle model
  14.3.1 Order Creation System (OCS)
  14.3.2 Order Realization System (ORS)
  14.3.3 Order Management System (OMS)
14.4 Behavioural aspects
  14.4.1 Front Office
  14.4.2 Back Office
  14.4.3 Middle Office
  14.4.4 External groups
14.5 Collecting requirements from multiple stakeholder viewpoints
  14.5.1 Critical use cases
14.6 Class architecture 250
   14.6.1 Class models and diagrams 250
14.7 Design guidelines for OPS 252
   14.7.1 Data patterns 253
14.8 Functional and non-functional requirements and their realization 256
   14.8.1 ISO 9126 revisited 257
14.9 Database repository: an architectural style for data-driven systems 258
14.10 Summary and conclusions 259
Appendix 14.1: Documenting use cases 259
Appendix 14.2: Some UML class diagrams 261

15. **Drink Vending Machine (DVM)** 263
   15.1 Introduction and objectives 263
   15.2 Description of problem 264
      15.2.1 Scope and span of problem 265
   15.3 Goals, processes and context 266
   15.4 Use cases 268
   15.5 Creating an initial PAC model 269
   15.6 Class structure 270
   15.7 Interaction diagrams and interface discovery 271
      15.7.1 Sequence diagrams 271
   15.8 Summary and conclusions 278
Appendix 15.1: Collaboration diagrams in a nutshell 278

16. **Multi-tasking lifecycle applications** 281
   16.1 Introduction and objectives 281
   16.2 The problem domain 282
      16.2.1 General description of problem 282
      16.2.2 System stakeholders 285
   16.3 System features 285
   16.4 System architecture 286
      16.4.1 The PAC models 289
   16.5 Design issues: overview 291
   16.6 The proof of the pudding: enter the ACE library 291
   16.7 The challenge: applying the ACE library in the extrusion application 293
   16.8 Summary and conclusions 298
Appendix 16.1: an introduction to multi-threading 298
PART IV  Domain architecture summary and ‘how to use’ documentation

17. Summary of domain architectures 309

17.1 Introduction and objectives 309
17.2 Object Creational Systems (OCS) 310
17.3 Object Alignment Systems (OAS) 311
17.4 Object Behavioural Systems (OBS)
  17.4.1 MIS 312
  17.4.2 PCS 313
  17.4.3 ACS 314
17.5 Keeping the domain architectures distinct and orthogonal 315
  17.5.1 MAN versus RAT 316
  17.5.2 MAN versus MIS 317
  17.5.3 MAN versus PCS 317
  17.5.4 MAN versus ACS 317
  17.5.5 RAT versus MIS 317
  17.5.6 RAT versus PCS 318
  17.5.7 RAT versus ACS 318
  17.5.8 MIS versus PCS 318
  17.5.9 MIS and PCS versus ACS 318
17.6 Summary and conclusions 319

18. Using domain architectures and analogical reasoning 321

18.1 Introduction and objectives 321
18.2 In which domain architecture does my application belong?
  The bird-watching method 322
18.3 Focusing on essential system features: the framework method 324
18.4 The defining-attribute view
  18.4.1 Advantages and disadvantages 326
18.5 The prototype view
  18.5.1 Advantages and disadvantages 328
18.6 The exemplar-based view
  18.6.1 Advantages and disadvantages 330
18.7 Summary and conclusions 331
Appendix 18.1: Analogical reasoning and learning by analogy 331

Appendix 1. The Inquiry Cycle and related cognitive techniques 335

A1.1 Introduction and objectives 335
A1.2 Background and history 336
| A1.3 An introduction to the Inquiry Cycle model | 336 |
| A1.3.1 Requirements documentation | 336 |
| A1.3.2 Requirements discussion | 337 |
| A1.3.3 Requirements evolution | 337 |
| A1.4 Using the right questions | 338 |
| A1.4.1 General applicability | 340 |
| A1.5 The learning loop | 341 |
| A1.6 Summary and conclusions | 342 |

**Appendix 2. The Presentation–Abstraction–Control (PAC) pattern**

| A2.1 Introduction and objectives | 345 |
| A2.2 Motivation and background | 346 |
| A2.2.1 A short history of objects | 347 |
| A2.2.2 Subsuming object orientation in a broader context | 348 |
| A2.3 Decomposition strategies | 348 |
| A2.3.1 System decomposition and activity diagrams | 350 |
| A2.3.2 System decomposition and context diagrams | 350 |
| A2.4 PAC and object-oriented analysis | 352 |
| A2.4.1 Entity classes | 355 |
| A2.5 The relationship between PAC and UML | 355 |
| A2.6 Summary and conclusions | 357 |

**Appendix 3. Relationships with other models and methodologies**

| A3.1 Introduction | 359 |
| A3.2 Information hiding and the work of David Parnas | 360 |
| A3.3 The Rummler–Brache approach | 361 |
| A3.4 Michael Jackson’s problem frames | 363 |
| A3.5 The Hatley–Pirbhai method | 364 |
| A3.6 The Garlan and Shaw architectural styles | 365 |
| A3.7 System and design patterns | 366 |
| A3.8 The Unified Modelling Language (UML) | 367 |
| A3.9 Viewpoint-based requirements engineering | 367 |

**Appendix 4. The ‘Hello World’ example: the Simple Digital Watch (SDW)**

| A4.1 Introduction | 371 |
| A4.2 Features and description of problem | 371 |
| A4.3 Goals and processes | 372 |
| A4.4 Stakeholders, viewpoints and requirements | 373 |
| A4.5 Context diagram and system decomposition | 373 |
| A4.6 Use cases | 375 |