

# Contents

---

<b>Preface</b>	<b>xv</b>
<b>PART I Background and fundamentals</b>	<b>1</b>
<b>1. Introducing and motivating domain architectures</b>	<b>3</b>
1.1 What is this book?	3
1.2 Why have we written this book?	4
1.3 For whom is this book intended?	5
1.4 Why should I read this book?	5
1.5 What is a domain architecture, really?	5
1.6 The Datasim Development Process (DDP)	8
1.7 The structure of this book	9
1.8 What this book does <i>not</i> cover	10
<b>2. Domain architecture catalogue</b>	<b>11</b>
2.1 Introduction and objectives	11
2.2 Management Information Systems (MIS) (Chapter 5)	13
2.3 Process Control Systems (PCS) (Chapter 6)	16
2.4 Resource Allocation and Tracking (RAT) systems (Chapter 7)	18
2.5 Manufacturing (MAN) systems (Chapter 8)	19
2.6 Access Control Systems (ACS) (Chapter 9)	20
2.7 Lifecycle and composite models (Chapter 10)	21
<b>3. Software lifecycle and Datasim Development Process (DDP)</b>	<b>23</b>
3.1 Introduction and objectives	23
3.2 The Software Lifecycle	24
3.3 Reducing the scope	25

3.4	The requirements/architecture phase in detail	29
3.5	The object-oriented analysis process	30
3.6	Project cultures and DDP	33
3.6.1	Calendar-driven projects	34
3.6.2	Requirements-driven projects	34
3.6.3	Documentation-driven style	35
3.6.4	Quality-driven style	36
3.6.5	Architecture-driven style	36
3.6.6	Process-driven style and the DDP	37
3.7	Summary and conclusions	38
<b>4.</b>	<b>Fundamental concepts and documentation issues</b>	<b>41</b>
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
<b>PART II</b>	<b>Domain architectures (meta models)</b>	<b>57</b>
<b>5.</b>	<b>Management Information Systems (MIS)</b>	<b>59</b>
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
<b>6.</b>	<b>Process Control Systems (PCS)</b>	<b>77</b>
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
<b>7.</b>	<b>Resource Allocation and Tracking (RAT) systems</b>	<b>111</b>
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
<b>8.</b>	<b>Manufacturing (MAN) systems</b>	<b>127</b>
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
<b>9.</b>	<b>Access Control Systems (ACS)</b>	<b>147</b>
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	157
9.10.1	Security models for Web-based applications	157
9.10.2	Access control during design: the Proxy pattern	159
9.11	Using ACS with other systems	162
<b>10.</b>	<b>Lifecycle and composite models</b>	<b>163</b>
10.1	Introduction and objectives	163
10.2	Background and history	164
10.3	Motivational example: the Rent-a-machine system	164
10.4	General applicability	168
10.5	Goals, processes and activities	170
10.6	Context diagram and system decomposition	171
10.7	Stakeholders, viewpoints and requirements	171
10.8	UML classes	172
10.9	Use cases	173
10.10	Specializations of LCM	174
10.11	Using LCM systems with other systems	174
10.12	Summary and conclusions	175
<b>PART III</b>	<b>Applications (models)</b>	<b>177</b>
<b>11.</b>	<b>Project resource management system: Manpower Control (MPC) system</b>	<b>179</b>
11.1	Introduction and objectives	179
11.2	Description and scope of problem	180
11.3	Core processing and context diagram	181
11.4	Requirements and use case analysis	183
11.4.1	Functional requirements and use cases	183
11.4.2	Non-functional requirements	186
11.5	Validating use cases	187
11.6	Class architecture	189
11.7	Generalizations	192
11.8	Summary and conclusions	192
<b>12.</b>	<b>Home Heating System (HHS)</b>	<b>193</b>
12.1	Introduction and objectives	193
12.2	Background and history	194
12.2.1	Hatley–Pirbhai	194
12.2.2	The Booch approach	197

12.3	Description of problem	197
12.4	Goals, processes and context	197
12.5	System decomposition and PAC model	200
12.6	Viewpoints and requirements analysis	201
12.7	Use cases	202
12.8	Validation efforts	207
12.9	Creating statecharts	209
12.10	Generalization efforts	212
12.11	Summary and conclusions	213
<b>13.</b>	<b>Elevator Control System (ELS)</b>	<b>215</b>
13.1	Introduction and objectives	215
13.2	Domain categories and ELS	216
13.3	A traditional object-oriented requirement specification	217
13.4	Re-engineering ELS: goals and processes	220
13.5	Stakeholders and their requirements	223
13.6	Requirements	225
13.7	System decomposition of ELS	227
13.8	PAC decomposition of ELS	230
13.9	Major use cases	232
13.9.1	Normal use cases	232
13.9.2	Exceptional use cases	233
13.10	Summary and conclusions	235
	Appendix 13.1: Definitions	235
<b>14.</b>	<b>Order Processing Systems (OPS)</b>	<b>237</b>
14.1	Introduction and objectives	237
14.2	Customer Requirements Specification (CRS): the product management vision of OPS	239
14.2.1	Business concerns and stakeholders' viewpoints	239
14.3	OPS as a lifecycle model	240
14.3.1	Order Creation System (OCS)	242
14.3.2	Order Realization System (ORS)	243
14.3.3	Order Management System (OMS)	244
14.4	Behavioural aspects	245
14.4.1	Front Office	246
14.4.2	Back Office	246
14.4.3	Middle Office	247
14.4.4	External groups	247
14.5	Collecting requirements from multiple stakeholder viewpoints	248
14.5.1	Critical use cases	249

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
<b>15.</b>	<b>Drink Vending Machine (DVM)</b>	<b>263</b>
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
<b>16.</b>	<b>Multi-tasking lifecycle applications</b>	<b>281</b>
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

<b>PART IV Domain architecture summary and ‘how to use’ documentation</b>	<b>307</b>
<b>17. Summary of domain architectures</b>	<b>309</b>
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)	312
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
<b>18. Using domain architectures and analogical reasoning</b>	<b>321</b>
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	325
18.4.1 Advantages and disadvantages	326
18.5 The prototype view	327
18.5.1 Advantages and disadvantages	328
18.6 The exemplar-based view	329
18.6.1 Advantages and disadvantages	330
18.7 Summary and conclusions	331
Appendix 18.1: Analogical reasoning and learning by analogy	331
<b>Appendix 1. The Inquiry Cycle and related cognitive techniques</b>	<b>335</b>
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
<b>Appendix 2. The Presentation–Abstraction–Control (PAC) pattern</b>		<b>345</b>
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
<b>Appendix 3. Relationships with other models and methodologies</b>		<b>359</b>
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
<b>Appendix 4. The ‘Hello World’ example: the Simple Digital Watch (SDW)</b>		<b>371</b>
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

A4.7	UML classes	375
A4.8	Statecharts	375
<b>Appendix 5. Using domain architectures: seven good habits</b>		<b>379</b>
<b>References</b>		<b>383</b>
<b>Index</b>		<b>387</b>