2.1.1.1 Processes and process decompositions 68
2.1.1.2 Process hierarchy diagram 68
2.1.2 Business process modeling 70
2.1.2.1 Flow objects, connecting objects, swimlanes and artifacts 70
2.1.2.2 Business process diagram 72
2.1.3 Solution envisioning 73
2.1.3.1 Solution envisioning process 74
2.1.3.2 Implementation strategies and capability architecture 76

Review quiz 2.1

2.2 Requirements elicitation 77
2.2.1 System requirements 79
2.2.1.1 Functional requirements 79
2.2.1.2 Non-functional requirements 80
2.2.2 Traditional methods of requirements elicitation 81
2.2.2.1 Interviewing customers and domain experts 81
2.2.2.2 Questionnaires 83
2.2.2.3 Observation 84
2.2.2.4 Study of documents and software systems 84
2.2.3 Modern methods of requirements elicitation 85
2.2.3.1 Prototyping 85
2.2.3.2 Brainstorming 86
2.2.3.3 Joint application development 87
2.2.3.4 Rapid application development 88

Review quiz 2.2

2.3 Requirements negotiation and validation 89
2.3.1 Out-of-scope requirements 89
2.3.2 Requirements dependency matrix 90
2.3.3 Requirements – risks and priorities 90

Review quiz 2.3

2.4 Requirements management 91
2.4.1 Requirements identification and classification 92
2.4.2 Requirements hierarchies 93
2.4.3 Change management 93
2.4.4 Requirements traceability 94

Review quiz 2.4

2.5 Requirements business model 94
2.5.1 System scope model 95
2.5.2 Business use case model 97
2.5.3 Business glossary 99
2.5.4 Business class model 101

Review quiz 2.5

2.6 Requirements document 103
2.6.1 Document templates 103
2.6.2 Project preliminaries 104
2.6.3 System services 105
3 Fundamentals of Visual Modeling

Objectives 122

3.1 The use case view 123
3.1.1 Actors 124
3.1.2 Use cases 124
3.1.3 Use case diagrams 126
3.1.4 Documenting use cases 127
Review quiz 3.1 128

3.2 The activity view 128
3.2.1 Actions 130
3.2.2 Activity diagrams 130
Review quiz 3.2 132

3.3 The structure view 133
3.3.1 Classes 133
3.3.2 Attributes 135
3.3.3 Associations 136
3.3.4 Aggregation 137
3.3.5 Generalization 138
3.3.6 Class diagrams 139
Review quiz 3.3 140

3.4 The interaction view 141
3.4.1 Sequence diagrams 141
3.4.2 Communication diagrams 143
3.4.3 Class methods 144
Review quiz 3.4 146

3.5 The state machine view 146
3.5.1 States and transitions 146
3.5.2 State machine diagrams 147
Review quiz 3.5 149
3.6  The implementation view
  3.6.1 Subsystems and packages 149
  3.6.2 Components and component diagrams 150
  3.6.3 Nodes and deployment diagrams 152
Review quiz 3.6 153
Summary 154
Key terms 154
Multiple-choice test 156
Questions 157
Exercises 158
Exercises: video store 159
Review quiz answers 161
Multiple-choice test answers 161
Answers to odd-numbered questions 162
Solutions to odd-numbered exercises 163
Solutions to odd-numbered exercises: video store 166

4  Requirements Specification
Objectives 169
4.1  Architectural prerogatives 170
  4.1.1 Model-View-Controller 171
  4.1.2 The Core J2EE architecture 172
  4.1.3 Presentation-Controller-Bean-Mediator-Entity-Resource
    4.1.3.1 The PCBMER layers 175
    4.1.3.2 The PCBMER principles 176
Review quiz 4.1 178

4.2 State specifications 178
  4.2.1 Modeling classes 178
    4.2.1.1 Discovering classes 179
    4.2.1.1.1 Noun phrase approach 179
    4.2.1.1.2 Common class pattern approach 180
    4.2.1.1.3 Use case-driven approach 181
    4.2.1.1.4 CRC approach 181
    4.2.1.1.5 Mixed approach 182
    4.2.1.1.6 Guidelines for discovering classes 182
    4.2.1.1.7 Examples of discovering classes 183
    4.2.1.2 Specifying classes 186
    4.2.1.2.1 Naming classes 186
    4.2.1.2.2 Discovering and specifying class attributes 186
    4.2.1.2.3 Examples of discovering and specifying class attributes 187
  4.2.2 Modeling associations 193
    4.2.2.1 Discovering associations 193
    4.2.2.2 Specifying associations 195
    4.2.2.2.1 Example of specifying associations 195
4.2.3  Modeling aggregation and composition relationships 196
  4.2.3.1  Discovering aggregations and composition 198
  4.2.3.2  Specifying aggregations and compositions 199
  4.2.3.3  Example of aggregation and composition specifications 199
4.2.4  Modeling generalization relationships 200
  4.2.4.1  Discovering generalizations 201
  4.2.4.2  Specifying generalization 201
  4.2.4.3  Examples of generalization specifications 202
4.2.5  Modeling interfaces 202
  4.2.5.1  Discovering interfaces 203
  4.2.5.2  Specifying interfaces 204
  4.2.5.3  Examples of interface specifications 204
4.2.6  Modeling objects 205
  4.2.6.1  Specifying objects 205
  4.2.6.2  Example of object specifications 205

Review quiz 4.2 206

4.3  Behavior specifications 206
  4.3.1  Modeling use cases 207
  4.3.1.1  Specifying use cases 208
  4.3.1.2  Examples of use case specifications 209
  4.3.2  Modeling activities 213
  4.3.2.1  Discovering actions 214
  4.3.2.2  Specifying actions 214
  4.3.2.3  Example of activity specifications 214
  4.3.3  Modeling interactions 215
  4.3.3.1  Discovering message sequences 216
  4.3.3.2  Specifying message sequences 216
  4.3.3.3  Examples of sequence specifications 216
  4.3.4  Modeling operations 219
  4.3.4.1  Discovering class operations 219
  4.3.4.2  Specifying class operations 219
  4.3.4.3  Examples of operation specifications 220

Review quiz 4.3 221

4.4  State change specifications 221
  4.4.1  Modeling object states 222
  4.4.1.1  Discovering object states 222
  4.4.1.2  Specifying object states 222
  4.4.1.3  Example of state machine specifications 223

Review quiz 4.4 223
Summary 224
Key terms 225
Multiple-choice test 226
Questions 228
Exercises: video store 229
Exercises: contact management 230
5 Moving from Analysis to Design

Objectives

5.1 Advanced class modeling

5.1.1 Extension mechanisms

5.1.1.1 Stereotypes

5.1.1.2 Comments and constraints

5.1.1.3 Tags

5.1.2 Visibility and encapsulation

5.1.2.1 Protected visibility

5.1.2.2 Accessibility of inherited class properties

5.1.2.3 Package and friendly visibility

5.1.3 Derived information

5.1.3.1 Derived attribute

5.1.3.2 Derived association

5.1.4 Qualified association

5.1.5 Association class versus reified class

5.1.5.1 Model with association class

5.1.5.2 Model with reified class

Review quiz 5.1

5.2 Advanced generalization and inheritance modeling

5.2.1 Generalization and substitutability

5.2.2 Inheritance versus encapsulation

5.2.3 Interface inheritance

5.2.4 Implementation inheritance

5.2.4.1 Proper use of implementation inheritance – extension inheritance

5.2.4.2 Problematic use of implementation inheritance – restriction inheritance

5.2.4.3 Improper use of implementation inheritance – convenience inheritance

5.2.4.4 The evils of implementation inheritance

5.2.4.4.1 Fragile base class

5.2.4.4.2 Overriding, down-calls and up-calls

5.2.4.4.3 Multiple implementation inheritance

Review quiz 5.2

5.3 Advanced aggregation and delegation modeling

5.3.1 Putting more semantics into aggregation

5.3.1.1 The ExclusiveOwns aggregation

5.3.1.2 The Owns aggregation
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1.3 The Has aggregation</td>
<td>273</td>
</tr>
<tr>
<td>5.3.1.4 The Member aggregation</td>
<td>274</td>
</tr>
<tr>
<td>5.3.2 Aggregation as an alternative to generalization</td>
<td>275</td>
</tr>
<tr>
<td>5.3.2.1 Delegation and prototypical systems</td>
<td>276</td>
</tr>
<tr>
<td>5.3.2.2 Delegation versus inheritance</td>
<td>276</td>
</tr>
<tr>
<td>5.3.3 Aggregation and holons – some cerebral ammunition</td>
<td>277</td>
</tr>
<tr>
<td>Review quiz 5.3</td>
<td>278</td>
</tr>
<tr>
<td>5.4 Advanced interaction modeling</td>
<td>278</td>
</tr>
<tr>
<td>5.4.1 Lifelines and messages</td>
<td>279</td>
</tr>
<tr>
<td>5.4.1.1 Accounting for basic technology</td>
<td>280</td>
</tr>
<tr>
<td>5.4.1.2 Visualizing technology contributions in interaction models</td>
<td>282</td>
</tr>
<tr>
<td>5.4.2 Fragments</td>
<td>284</td>
</tr>
<tr>
<td>5.4.3 Interaction uses</td>
<td>287</td>
</tr>
<tr>
<td>Review quiz 5.4</td>
<td>288</td>
</tr>
<tr>
<td>Summary</td>
<td>288</td>
</tr>
<tr>
<td>Key terms</td>
<td>289</td>
</tr>
<tr>
<td>Multiple-choice test</td>
<td>289</td>
</tr>
<tr>
<td>Questions</td>
<td>290</td>
</tr>
<tr>
<td>Exercises</td>
<td>291</td>
</tr>
<tr>
<td>Exercises: time logging</td>
<td>292</td>
</tr>
<tr>
<td>Exercises: advertising expenditure</td>
<td>293</td>
</tr>
<tr>
<td>Review quiz answers</td>
<td>295</td>
</tr>
<tr>
<td>Multiple-choice test answers</td>
<td>295</td>
</tr>
<tr>
<td>Answers to odd-numbered questions</td>
<td>295</td>
</tr>
<tr>
<td>Solutions to odd-numbered exercises</td>
<td>299</td>
</tr>
<tr>
<td>Solutions to exercises: time logging</td>
<td>300</td>
</tr>
</tbody>
</table>

6 System Architecture and Program Design 304

Objectives 304

6.1 Distributed physical architecture 305

6.1.1 Peer-to-peer architecture 306

6.1.2 Tiered architecture 307

6.1.3 Database-centered architecture 308

Review quiz 6.1 309

6.2 Multilayer logical architecture 310

6.2.1 Architectural complexity 311

6.2.1.1 Spatial cognitive complexity 312

6.2.1.2 Structural complexity 313

6.2.1.2.1 Structural complexity of networks 313

6.2.1.2.2 Structural complexity of hierarchies 314

6.2.2 Architectural patterns 316

6.2.2.1 Façade 317

6.2.2.2 Abstract Factory 318

6.2.2.3 Chain of Responsibility 322

6.2.2.4 Observer 324
8 Persistence and Database Design

Objectives

8.1 Business objects and persistence
8.1.1 Database management systems
8.1.2 Levels of data models
8.1.3 Integrating application and database modeling
8.1.4 Underpinnings of object–database mapping
Review quiz 8.1

8.2 Relational database model
8.2.1 Columns, domains and rules
8.2.2 Tables
8.2.3 Referential integrity
8.2.4 Triggers
8.2.5 Stored procedures
8.2.6 Views
8.2.7 Normal forms
Review quiz 8.2

8.3 Object-relational mapping
8.3.1 Mapping entity classes
8.3.2 Mapping associations
8.3.3 Mapping aggregations
8.3.4 Mapping generalizations
Review quiz 8.3

8.4 Patterns for managing persistent objects
8.4.1 Searching for persistent objects
8.4.2 Loading persistent objects
8.4.3 Unloading persistent objects
Review quiz 8.4

8.5 Designing database access and transactions
8.5.1 Levels of SQL programming
8.5.2 Designing business transactions
8.5.2.1 Short transactions
  8.5.2.1.1 Pessimistic concurrency control
  8.5.2.1.2 Levels of isolation
  8.5.2.1.3 Automatic recovery
  8.5.2.1.4 Programmable recovery
  8.5.2.1.5 Designing stored procedures and triggers
8.5.2.2 Long transactions
Review quiz 8.5

Summary

Key terms

Multiple-choice tests

Questions

Exercises: contact management
Exercises: telemarketing

Review quiz answers
9 Quality and Change Management

Objectives

9.1 Quality management
9.1.1 Quality assurance
  9.1.1.1 Checklists, reviews and audits
  9.1.1.2 Test-driven development
9.1.2 Quality control
  9.1.2.1 Test concepts and techniques
  9.1.2.2 Testing system services
  9.1.2.3 Testing system constraints
    9.1.2.3.1 Graphical user interface testing
    9.1.2.3.2 Database testing
    9.1.2.3.3 Authorization testing
    9.1.2.3.4 Testing of other constraints
Review quiz 9.1

9.2 Change management
9.2.1 Instrumenting and managing change requests
  9.2.1.1 Submitting a change request
  9.2.1.2 Keeping track of change requests
9.2.2 Traceability
  9.2.2.1 System features to use cases and use case requirements
  9.2.2.2 Test plans to test cases and test requirements
  9.2.2.3 UML diagrams to documents and requirements
  9.2.2.4 Use case requirements to test requirements
  9.2.2.5 Test requirements to defects
  9.2.2.6 Use case requirements to enhancements
Review quiz 9.2

Summary

Key terms

Multiple-choice test

Questions

Review quiz answers

Multiple-choice test answers

Answers to odd-numbered questions

10 Tutorial-style Review and Reinforcement

Objectives

10.1 Use case modeling
  10.1.1 Actors
  10.1.2 Use cases
  10.1.3 Use case diagram
  10.1.4 Documenting use cases
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2</td>
<td>Activity modeling</td>
<td>526</td>
</tr>
<tr>
<td>10.2.1</td>
<td>Actions</td>
<td>526</td>
</tr>
<tr>
<td>10.2.2</td>
<td>Activity diagram</td>
<td>528</td>
</tr>
<tr>
<td>10.3</td>
<td>Class modeling</td>
<td>529</td>
</tr>
<tr>
<td>10.3.1</td>
<td>Classes</td>
<td>529</td>
</tr>
<tr>
<td>10.3.2</td>
<td>Attributes</td>
<td>531</td>
</tr>
<tr>
<td>10.3.3</td>
<td>Associations</td>
<td>532</td>
</tr>
<tr>
<td>10.3.4</td>
<td>Aggregations</td>
<td>533</td>
</tr>
<tr>
<td>10.3.5</td>
<td>Generalizations</td>
<td>533</td>
</tr>
<tr>
<td>10.3.6</td>
<td>Class diagram</td>
<td>533</td>
</tr>
<tr>
<td>10.4</td>
<td>Interaction modeling</td>
<td>534</td>
</tr>
<tr>
<td>10.4.1</td>
<td>Sequence diagram</td>
<td>534</td>
</tr>
<tr>
<td>10.4.2</td>
<td>Communication diagram</td>
<td>537</td>
</tr>
<tr>
<td>10.5</td>
<td>State machine modeling</td>
<td>540</td>
</tr>
<tr>
<td>10.5.1</td>
<td>States and transitions</td>
<td>540</td>
</tr>
<tr>
<td>10.5.2</td>
<td>State machine diagram</td>
<td>541</td>
</tr>
<tr>
<td>10.6</td>
<td>Implementation models</td>
<td>542</td>
</tr>
<tr>
<td>10.6.1</td>
<td>Subsystems</td>
<td>542</td>
</tr>
<tr>
<td>10.6.2</td>
<td>Packages</td>
<td>543</td>
</tr>
<tr>
<td>10.6.3</td>
<td>Components</td>
<td>545</td>
</tr>
<tr>
<td>10.6.4</td>
<td>Notes</td>
<td>546</td>
</tr>
<tr>
<td>10.7</td>
<td>Object collaboration design</td>
<td>548</td>
</tr>
<tr>
<td>10.7.1</td>
<td>Use case design specifications</td>
<td>550</td>
</tr>
<tr>
<td>10.7.2</td>
<td>User interface prototyping</td>
<td>552</td>
</tr>
<tr>
<td>10.7.3</td>
<td>Sequence diagram</td>
<td>553</td>
</tr>
<tr>
<td>10.7.4</td>
<td>Design-level class diagram</td>
<td>555</td>
</tr>
<tr>
<td>10.8</td>
<td>Window navigation design</td>
<td>556</td>
</tr>
<tr>
<td>10.8.1</td>
<td>User experience (UX) elements</td>
<td>556</td>
</tr>
<tr>
<td>10.8.2</td>
<td>Behavioral UX collaboration</td>
<td>556</td>
</tr>
<tr>
<td>10.8.3</td>
<td>Structural UX collaboration</td>
<td>558</td>
</tr>
<tr>
<td>10.9</td>
<td>Database design</td>
<td>559</td>
</tr>
<tr>
<td>10.9.1</td>
<td>Object-relational mapping</td>
<td>559</td>
</tr>
<tr>
<td>10.9.2</td>
<td>Referential integrity design</td>
<td>559</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td>562</td>
</tr>
<tr>
<td>Exercises: online shopping</td>
<td>562</td>
<td></td>
</tr>
</tbody>
</table>

Appendix: Fundamentals of Object Technology | 565 |
<p>| A.1 Real-life analogy | 566 |
| A.2 Instance object | 566 |
| A.2.1 Object notation | 567 |
| A.2.2 How do objects collaborate? | 568 |
| A.2.3 Identity and object communication | 569 |
| A.2.3.1 Persistent link | 569 |
| A.2.3.2 Transient link | 570 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2.3.3 Message passing</td>
<td>571</td>
</tr>
<tr>
<td>A.3 Class</td>
<td>571</td>
</tr>
<tr>
<td>A.3.1 Attribute</td>
<td>572</td>
</tr>
<tr>
<td>A.3.1.1 Attribute type that designates a class</td>
<td>572</td>
</tr>
<tr>
<td>A.3.1.2 Attribute visibility</td>
<td>573</td>
</tr>
<tr>
<td>A.3.2 Operation</td>
<td>574</td>
</tr>
<tr>
<td>A.3.2.1 Operations objects collaborate</td>
<td>574</td>
</tr>
<tr>
<td>A.3.2.2 Operation visibility and scope</td>
<td>575</td>
</tr>
<tr>
<td>A.3.3 Class object</td>
<td>575</td>
</tr>
<tr>
<td>A.4 Variables, methods and constructors</td>
<td>576</td>
</tr>
<tr>
<td>A.5 Association</td>
<td>578</td>
</tr>
<tr>
<td>A.5.1 Association degree</td>
<td>579</td>
</tr>
<tr>
<td>A.5.2 Association multiplicity</td>
<td>579</td>
</tr>
<tr>
<td>A.5.3 Association link and extent</td>
<td>580</td>
</tr>
<tr>
<td>A.5.4 Association class</td>
<td>581</td>
</tr>
<tr>
<td>A.6 Aggregation and composition</td>
<td>582</td>
</tr>
<tr>
<td>A.6.1 Buried reference</td>
<td>583</td>
</tr>
<tr>
<td>A.6.2 Inner class</td>
<td>583</td>
</tr>
<tr>
<td>A.6.3 Delegation</td>
<td>586</td>
</tr>
<tr>
<td>A.7 Generalization and inheritance</td>
<td>587</td>
</tr>
<tr>
<td>A.7.1 Polymorphism</td>
<td>588</td>
</tr>
<tr>
<td>A.7.2 Overriding versus overloading</td>
<td>589</td>
</tr>
<tr>
<td>A.7.3 Multiple inheritance</td>
<td>590</td>
</tr>
<tr>
<td>A.7.4 Multiple classification</td>
<td>590</td>
</tr>
<tr>
<td>A.7.5 Dynamic classification</td>
<td>591</td>
</tr>
<tr>
<td>A.8 Abstract class</td>
<td>592</td>
</tr>
<tr>
<td>A.9 Interface</td>
<td>593</td>
</tr>
<tr>
<td>A.9.1 Interface versus abstract class</td>
<td>593</td>
</tr>
<tr>
<td>A.9.2 Implementing an interface</td>
<td>594</td>
</tr>
<tr>
<td>A.9.3 Using an interface</td>
<td>594</td>
</tr>
<tr>
<td>Summary</td>
<td>595</td>
</tr>
<tr>
<td>Questions</td>
<td>597</td>
</tr>
<tr>
<td>Answers to odd-numbered questions</td>
<td>598</td>
</tr>
<tr>
<td>Bibliography</td>
<td>600</td>
</tr>
<tr>
<td>Index</td>
<td>607</td>
</tr>
</tbody>
</table>