# Table of Contents

Mastering UML with Rational Rose 2002

## Chapter 1: Introduction to UML
- Introduction to the Object−Oriented Paradigm ............................................. 4
  - Encapsulation .................................................................................................. 5
  - Inheritance ..................................................................................................... 6
  - Polymorphism ................................................................................................ 8
- What Is Visual Modeling? .............................................................................. 8
- Systems of Graphical Notation .................................................................... 9
  - Booch Notation ............................................................................................ 10
  - Object Management Technology (OMT) .................................................. 10
  - Unified Modeling Language (UML) ............................................................. 11
- Understanding UML Diagrams ................................................................. 12
  - Business Use Case Diagrams .................................................................... 13
  - Use Case Diagrams ..................................................................................... 14
  - Activity Diagrams ....................................................................................... 14
  - Sequence Diagrams ................................................................................... 16
  - Collaboration Diagrams ............................................................................ 17
  - Class Diagrams ........................................................................................... 17
  - Statechart Diagrams .................................................................................. 18
  - Component Diagrams ............................................................................... 19
- Deployment Diagrams ................................................................................ 21
- Visual Modeling and the Software Development Process ......................... 22
  - Inception ...................................................................................................... 24
  - Elaboration ................................................................................................. 24
  - Construction ............................................................................................... 25
  - Transition .................................................................................................... 26
- Summary ........................................................................................................ 26

## Chapter 2: A Tour of Rose
- What Is Rose? ............................................................................................... 27
- Getting Around in Rose ............................................................................... 30
  - Parts of the Screen ...................................................................................... 31
- Exploring Four Views in a Rose Model ....................................................... 38
  - Use Case View ............................................................................................ 38
  - Logical View ............................................................................................... 40
  - Component View ....................................................................................... 42
  - Deployment View ....................................................................................... 43
- Working with Rose ....................................................................................... 45
  - Creating Models ......................................................................................... 45
  - Saving Models ............................................................................................. 46
  - Exporting and Importing Models .............................................................. 47
  - Publishing Models to the Web ................................................................... 48
  - Working with Controlled Units .................................................................. 50
  - Using the Model Integrator ....................................................................... 52
  - Working with Notes ................................................................................... 53
  - Working with Packages ............................................................................. 54
  - Adding Files and URLs to Rose Model Elements ..................................... 56
  - Adding and Deleting Diagrams ................................................................. 56
- Setting Global Options ................................................................................ 57
  - Working with Fonts .................................................................................... 58
  - Working with Colors .................................................................................. 58
Table of Contents

Chapter 2: A Tour of Rose

Summary.........................................................59

Chapter 3: Business Modeling.........................................................61

Introduction to Business Modeling.....................................................61
  Why Model the Business?.........................................................61
  Do I Need to Do Business Modeling?..........................................62
  Business Modeling in an Iterative Process....................................63

Business–Modeling Concepts.........................................................66
  Business Actors............................................................................66
  Business Workers..........................................................................67
  Business Use Cases.........................................................................68
  Business Use Case Diagrams.......................................................69
  Activity Diagrams.........................................................................70
  Business Entities.............................................................................72
  Organization Unit...........................................................................73

Where Do I Start?...............................................................................74
  Identifying the Business Actors....................................................74
  Identifying the Business Workers................................................75
  Identifying the Business Use Cases..............................................76
  Showing the Interactions...............................................................76
  Documenting the Details...............................................................77

Creating Business Use Case Diagrams............................................78
  Deleting Business Use Case Diagrams...........................................79
  The Use Case Diagram Toolbar...................................................80
  Adding Business Use Cases........................................................80
  Business Use Case Specifications................................................81
  Assigning a Priority to a Business Use Case...............................83
  Viewing Diagrams for a Business Use Case.................................83
  Viewing Relationships for a Business Use Case...........................86

Working with Business Actors.......................................................87
  Adding Business Actors...............................................................87
  Adding Actor Specifications.........................................................88
  Assigning an Actor Stereotype.....................................................89
  Setting Business Actor Multiplicity..............................................89
  Viewing Relationships for a Business Actor..................................90

Working with Relationships.........................................................91
  Association Relationship.............................................................91
  Generalization Relationship.......................................................92

Working with Organization Units................................................94
  Adding Organization Units........................................................94
  Deleting Organization Units.......................................................95

Activity Diagrams..........................................................................95
  Adding an Activity Diagram.......................................................96
  Adding Details to an Activity Diagram........................................97

Summary.........................................................................................102

Chapter 4: Use Cases and Actors.................................................104

Use Case Modeling Concepts.......................................................104
  Actors.........................................................................................104
  Use Cases....................................................................................105
  Traceability..................................................................................107
## Table of Contents

### Chapter 4: Use Cases and Actors
- Flow of Events.................................................................108
- Relationships.................................................................114
- Use Case Diagrams..........................................................118
- Activity Diagrams............................................................119
- Activity..............................................................................120
- Start and End States.........................................................121
- Objects and Object Flows..................................................121
- Transitions.......................................................................122
- Synchronization..............................................................122

#### Working with Use Cases in Rational Rose
- The Use Case Diagram Toolbar........................................123
- Creating Use Case Diagrams............................................124
- Deleting Use Case Diagrams.............................................126
- Adding Use Cases............................................................127
- Deleting Use Cases..........................................................129
- Use Case Specifications....................................................130
- Naming a Use Case..........................................................131
- Viewing Participants of a Use Case.................................132
- Assigning a Use Case Stereotype......................................132
- Assigning a Priority to a Use Case.................................133
- Creating an Abstract Use Case.........................................133
- Viewing Diagrams for a Use Case....................................134
- Viewing Relationships for a Use Case.............................136

#### Working with Actors
- Adding Actors.................................................................137
- Deleting Actors...............................................................139
- Actor Specifications.........................................................139
- Naming Actors.................................................................141
- Assigning an Actor Stereotype.........................................142
- Setting Actor Multiplicity...............................................142
- Creating an Abstract Actor..............................................143
- Viewing Relationships for an Actor...............................144
- Viewing an Actor's Instances.........................................145

#### Working with Relationships
- Association Relationship................................................145
- Includes Relationship.....................................................146
- Extends Relationship.....................................................148
- Generalization Relationship............................................148

#### Working with Activity Diagrams
- The Activity Diagram Toolbar.........................................149
- Creating Activity Diagrams..............................................150
- Deleting Activity Diagrams..............................................154

#### Exercise
- Problem Statement..........................................................154
- Create a Use Case Diagram............................................154

### Chapter 5: Object Interaction
- Interaction Diagrams......................................................159
- What Is an Object............................................................161
- What Is a Class...............................................................161
# Table of Contents

## Chapter 5: Object Interaction

- Where Do I Start? ................................................................. 162
- Finding Objects .................................................................. 162
- Finding the Actor ............................................................ 163
- Using Interaction Diagrams ........................................... 164
- Sequence Diagrams ........................................................ 164
  - The Sequence Diagram Toolbar .................................. 165
- Collaboration Diagrams ................................................... 166
  - The Collaboration Diagram Toolbar ............................ 167
- Working with Actors on an Interaction Diagram .......... 167
- Working with Objects ...................................................... 168
  - Adding Objects to an Interaction Diagram .................. 168
  - Deleting Objects from an Interaction Diagram .......... 169
  - Setting Object Specifications ................................... 169
  - Naming an Object ..................................................... 170
  - Mapping an Object to a Class .................................... 171
  - Setting Object Persistence ......................................... 173
  - Using Multiple Instances of an Object ....................... 174
- Working with Messages ................................................... 175
  - Adding Messages to an Interaction Diagram .............. 175
  - Adding Messages to a Sequence Diagram .................. 175
  - Deleting Messages from a Sequence Diagram .......... 176
  - Reordering Messages in a Sequence Diagram .......... 176
  - Message Numbering in a Sequence Diagram ............ 177
  - Viewing the Focus of Control in a Sequence Diagram ... 178
  - Adding Messages to a Collaboration Diagram ............ 179
  - Deleting Messages from a Collaboration Diagram ....... 180
  - Message Numbering in a Collaboration Diagram ...... 181
  - Adding Data Flows to a Collaboration Diagram .......... 181
  - Setting Message Specifications ................................. 182
  - Naming a Message ..................................................... 182
  - Mapping a Message to an Operation ......................... 183
  - Setting Message Synchronization Options ................ 185
  - Setting Message Frequency ....................................... 188
- End of a Lifeline ............................................................... 189
- Working with Scripts ....................................................... 189
- Switching Between Sequence and Collaboration Diagrams 191
- Two–Pass Approach to Interaction Diagrams ............... 192
- Exercise ...................................................................... 195
  - Problem Statement ................................................... 195
  - Create Interaction Diagrams .................................... 195
- Summary ................................................................. 200

## Chapter 6: Classes and Packages .............................................. 201

- Logical View of a Rose Model .......................................... 201
- Class Diagrams ............................................................. 201
  - What Is a Class? ......................................................... 202
  - Finding Classes ........................................................ 203
  - Creating Class Diagrams ......................................... 205
  - Deleting Class Diagrams .......................................... 207
  - Organizing Items on a Class Diagram ....................... 207
  - Using the Class Diagram Toolbar .............................. 208
# Table of Contents

## Chapter 6: Classes and Packages
- Working with Classes ................................................................. 209
- Adding Classes ........................................................................ 209
- Class Stereotypes ................................................................. 212
  - Analysis Stereotypes ...................................................... 212
  - Class Types ................................................................. 217
  - Interfaces .......................................................................... 224
  - Web Modeling Stereotypes ............................................. 225
  - Other Language Stereotypes ....................................... 228
- Class Specifications ................................................................. 230
  - Naming a Class .................................................................. 231
  - Setting Class Visibility .................................................... 232
  - Setting Class Multiplicity .............................................. 233
  - Setting Storage Requirements for a Class ..................... 234
  - Setting Class Persistence .............................................. 234
  - Setting Class Concurrency ............................................ 235
  - Creating an Abstract Class ............................................ 235
  - Viewing Class Attributes .............................................. 236
  - Viewing Class Operations ........................................... 236
  - Viewing Class Relationships ....................................... 237
  - Using Nested Classes ................................................... 237
  - Viewing the Interaction Diagrams That Contain a Class ... 238
  - Setting Java Class Specifications .................................... 239
  - Setting CORBA Class Specifications ................................ 241
- Working with Packages .............................................................. 242
  - Adding Packages .......................................................... 242
  - Deleting Packages .......................................................... 243
- Exercise ...................................................................................... 244
  - Problem Statement ......................................................... 244
  - Creating a Class Diagram .............................................. 244
- Summary .................................................................................. 250

## Chapter 7: Attributes and Operations ................................................................. 251
- Working with Attributes .......................................................... 251
  - Finding Attributes .......................................................... 251
  - Adding Attributes .......................................................... 252
  - Deleting Attributes .......................................................... 255
  - Setting Attribute Specifications ..................................... 256
  - Setting the Attribute Containment .................................. 264
  - Making an Attribute Static ............................................ 265
  - Specifying a Derived Attribute ..................................... 265
- Working with Operations .......................................................... 266
  - Finding Operations ......................................................... 267
  - Adding Operations .......................................................... 268
  - Deleting Operations .......................................................... 271
  - Setting Operation Specifications .................................... 272
  - Adding Arguments to an Operation ............................... 278
  - Specifying the Operation Protocol ............................... 279
  - Specifying the Operation Qualifications ....................... 280
  - Specifying the Operation Exceptions ............................ 281
  - Specifying the Operation Size ....................................... 281
  - Specifying the Operation Time ....................................... 281
# Table of Contents

## Chapter 7: Attributes and Operations
- Specifying the Operation Concurrency ............................................................ 282
- Specifying the Operation Preconditions .......................................................... 282
- Specifying the Operation Postconditions ......................................................... 283
- Specifying the Operation Semantics ............................................................... 284
- Displaying Attributes and Operations on Class Diagrams ............................... 285
- Showing Attributes ......................................................................................... 286
- Showing Operations ....................................................................................... 288
- Showing Visibility ......................................................................................... 290
- Showing Stereotypes ..................................................................................... 291
- Mapping Operations to Messages ................................................................... 292
- Mapping an Operation to a Message on an Interaction Diagram .................. 294
- Exercise ......................................................................................................... 295
- Problem Statement ....................................................................................... 295
- Add Attributes and Operations ...................................................................... 296
- Summary ....................................................................................................... 300

## Chapter 8: Relationships .................................................................................. 301
- Relationships ................................................................................................. 301
  - Types of Relationships .................................................................................. 301
  - Finding Relationships ................................................................................... 303
- Associations ................................................................................................... 304
  - Using Web Association Stereotypes .............................................................. 306
  - Creating Associations ................................................................................... 307
  - Deleting Associations .................................................................................. 310
- Dependencies .................................................................................................. 311
  - Creating Dependencies ................................................................................ 313
  - Deleting Dependencies ................................................................................ 314
- Package Dependencies .................................................................................... 315
  - Creating Package Dependencies .................................................................. 316
  - Deleting Package Dependencies .................................................................. 317
- Aggregations .................................................................................................... 317
  - Creating Aggregations ................................................................................ 318
  - Deleting Aggregations ................................................................................. 320
- Generalizations ............................................................................................... 321
  - Creating Generalizations .......................................................................... 322
  - Deleting Generalizations .......................................................................... 323
- Working with Relationships ............................................................................ 324
  - Setting Multiplicity ..................................................................................... 324
  - Using Relationship Names ......................................................................... 326
  - Using Stereotypes ....................................................................................... 327
  - Using Roles .................................................................................................. 328
  - Setting Export Control ............................................................................... 330
  - Using Static Relationships ......................................................................... 331
  - Using Friend Relationships ......................................................................... 332
  - Setting Containment ................................................................................... 333
  - Using Qualifiers ......................................................................................... 334
  - Using Link Elements .................................................................................. 335
  - Using Constraints ....................................................................................... 336
- Exercise ......................................................................................................... 338
- Problem Statement ....................................................................................... 338
- Adding Relationships ..................................................................................... 338
# Table of Contents

## Chapter 8: Relationships
- Summary..............................................................................................................................................340

## Chapter 9: Object Behavior......................................................................................................................341
- Statechart Diagrams.........................................................................................................................341
  - Creating a Statechart Diagram........................................................................................................342
  - Adding States...................................................................................................................................343
  - Adding State Details........................................................................................................................344
  - Adding Transitions........................................................................................................................349
  - Adding Transition Details............................................................................................................350
  - Adding Special States...............................................................................................................352
  - Using Nested States and State History..................................................................................353
- Exercise..................................................................................................................................................355
  - Problem Statement....................................................................................................................355
  - Create a Statechart Diagram..................................................................................................356
- Summary...............................................................................................................................................359

## Chapter 10: Component View.....................................................................................................................360
- What Is a Component?........................................................................................................................360
  - Types of Components..................................................................................................................360
- Component Diagrams........................................................................................................................362
  - Creating Component Diagrams....................................................................................................363
  - Adding Components..................................................................................................................364
  - Adding Component Details.........................................................................................................367
  - Adding Component Dependencies..........................................................................................372
- Exercise.................................................................................................................................................373
  - Problem Statement....................................................................................................................373
- Summary.............................................................................................................................................380

## Chapter 11: Deployment View.....................................................................................................................381
- Deployment Diagrams........................................................................................................................381
  - Opening the Deployment Diagram...............................................................................................381
  - Adding Processors.....................................................................................................................382
  - Adding Processor Details............................................................................................................384
  - Adding Devices........................................................................................................................387
  - Adding Device Details................................................................................................................389
  - Adding Connections...................................................................................................................391
  - Adding Connection Details.........................................................................................................392
  - Adding Processes.......................................................................................................................394
- Exercise..................................................................................................................................................397
  - Problem Statement....................................................................................................................397
  - Create Deployment Diagram..................................................................................................397
- Summary.............................................................................................................................................400

## Chapter 12: Introduction to Code Generation and Reverse Engineering Using Rational Rose........401
- Preparing for Code Generation...........................................................................................................401
  - Step One: Check the Model.........................................................................................................402
  - Step Two: Create Components.....................................................................................................404
  - Step Three: Map Classes to Components......................................................................................405
  - Step Four: Set the Code—Generation Properties..........................................................................406
  - Step Five: Select a Class, Component, or Package......................................................................409
  - Step Six: Generate Code..............................................................................................................409
Table of Contents

Chapter 12: Introduction to Code Generation and Reverse Engineering Using Rational Rose
What Gets Generated?.........................................................................................................................410
Introduction to Reverse Engineering Using Rational Rose...............................................................411
Model Elements Created During Reverse Engineering.................................................................412
Round-Trip Engineering....................................................................................................................415
Summary........................................................................................................................................415

Generating Code in ANSI C++ and Visual C++. ............................................................................417
Converting a C++ Model to an ANSI C++ Model............................................................................418
ANSI C++ Code−Generation Properties..........................................................................................419
  Class Properties............................................................................................................................420
  Attribute Properties.......................................................................................................................421
  Operation Properties......................................................................................................................422
  Package (Class Category) Properties............................................................................................424
  Component (Module Specification) Properties..............................................................................424
  Role Properties.............................................................................................................................427
  Generalization Properties..............................................................................................................428
Visual C++ Code−Generation Properties.......................................................................................428
  Class Model Assistant..................................................................................................................428
  Component Properties................................................................................................................431
  Project Properties.........................................................................................................................433
  Visual C++ and ATL Objects.........................................................................................................434
Generated Code...............................................................................................................................435
  Code Generated for Classes..........................................................................................................435
  Code Generated for Attributes......................................................................................................439
  Code Generated for Operations....................................................................................................441
Visual C++ Code Generation............................................................................................................443
Reverse Engineering ANSI C++. ....................................................................................................443
Reverse Engineering Visual C++. ....................................................................................................445
Summary........................................................................................................................................445

Chapter 14: Java Code Generation and Reverse Engineering....................................................447
Overview.........................................................................................................................................447
Introduction to Rose J.......................................................................................................................448
Beginning a Java Project....................................................................................................................449
  Selecting a Java Framework..........................................................................................................449
  Linking to IBM VisualAge for Java..............................................................................................450
  Linking to Microsoft Visual J++....................................................................................................451
Java Code−Generation Properties..................................................................................................451
  Project Properties.........................................................................................................................452
  Class Properties..........................................................................................................................456
  Attribute Properties......................................................................................................................458
  Operation Properties....................................................................................................................459
  Module Properties.........................................................................................................................460
  Role Properties.............................................................................................................................461
Generating Code...............................................................................................................................462
Generated Code...............................................................................................................................462
  Classes......................................................................................................................................463
  Attributes..................................................................................................................................465
  Operations.................................................................................................................................466
  Bidirectional Associations.............................................................................................................468
# Table of Contents

Chapter 14: Java Code Generation and Reverse Engineering

- Unidirectional Associations ................................................................. 471
- Associations with a Multiplicity of One to Many ...................................... 472
- Associations with a Multiplicity of Many to Many .................................. 474
- Reflexive Associations ........................................................................ 476
- Aggregations ....................................................................................... 476
- Dependency Relationships ..................................................................... 478
- Generalization Relationships .............................................................. 479
- Interfaces ............................................................................................ 480
- Java Beans .......................................................................................... 481
- Support for J2EE .................................................................................. 484
  - EJBs ............................................................................................... 484
  - Servlets ......................................................................................... 487
  - JAR and WAR Files ....................................................................... 488
  - Automated J2EE Deployment ......................................................... 489
- Reverse Engineering .............................................................................. 490
- Summary ............................................................................................. 491

Chapter 15: Visual Basic Code Generation and Reverse Engineering ......................................................... 493

- Starting a Visual Basic Project ............................................................. 494
- Visual Basic Code−Generation Properties ............................................ 494
  - Class Properties ............................................................................. 495
  - Attribute Properties ....................................................................... 498
  - Operation Properties ....................................................................... 499
- Module Specification Properties .......................................................... 502
- Role Properties .................................................................................... 503
- Generalization Properties ..................................................................... 504
- Using the Code−Generation Wizard ..................................................... 505
- Generated Code .................................................................................. 509
  - Classes ............................................................................................ 509
  - Attributes ....................................................................................... 530
  - Operations ....................................................................................... 531
  - Bidirectional Associations .............................................................. 531
  - Unidirectional Associations ............................................................ 533
  - Associations with a Multiplicity of One to Many .................................. 534
  - Associations with a Multiplicity of Many to Many ................................ 534
  - Reflexive Associations ..................................................................... 535
  - Aggregations .................................................................................... 535
  - Dependency Relationships .............................................................. 536
  - Generalization Relationships .......................................................... 536
- Reverse Engineering .............................................................................. 537
- Summary ............................................................................................. 539

Chapter 16: XML DTD Code Generation and Reverse Engineering ............................................................ 541

- Overview ............................................................................................. 541
- Introduction to XML DTD ..................................................................... 542
  - Elements ........................................................................................ 542
  - Attributes ....................................................................................... 543
  - Entities and Notations ..................................................................... 543
- DTD−to−UML Mapping ....................................................................... 545
- DTD Code−Generation Properties ....................................................... 546
  - Project Properties ........................................................................... 546
# Table of Contents

## Chapter 16: XML DTD Code Generation and Reverse Engineering
- Class Properties..........................................................................................................................547
- Attribute Properties......................................................................................................................551
- Role Properties.............................................................................................................................552
- Component Properties.....................................................................................................................553
- Generated Code...............................................................................................................................554
  - Classes.......................................................................................................................................555
  - Attributes.................................................................................................................................562
- Reverse Engineering DTD..............................................................................................................564
- Summary.......................................................................................................................................565

## Chapter 17: CORBA/IDL Code Generation and Reverse Engineering..........................................................567
- CORBA/IDL Code−Generation Properties.....................................................................................567
  - Project Properties.......................................................................................................................568
  - Class Properties..........................................................................................................................570
  - Attribute Properties.....................................................................................................................575
  - Operation Properties.....................................................................................................................576
  - Module Properties........................................................................................................................578
  - Association (Role) Properties.......................................................................................................579
  - Dependency Properties.................................................................................................................580
- Generated Code...............................................................................................................................581
  - Classes.......................................................................................................................................581
  - Attributes.................................................................................................................................588
  - Operations.................................................................................................................................591
  - Bidirectional Associations.........................................................................................................591
  - Unidirectional Associations........................................................................................................595
  - Associations with a Multiplicity of One to Many......................................................................595
  - Associations with a Multiplicity of Many to Many....................................................................599
  - Associations with Bounded Multiplicity..................................................................................600
  - Reflexive Associations.................................................................................................................602
  - Aggregations..............................................................................................................................604
  - Dependency Relationships............................................................................................................604
  - Generalization Relationships......................................................................................................605
- Reverse Engineering Relationships..................................................................................................608
- Reverse Engineering CORBA Source Code....................................................................................609
- Summary.......................................................................................................................................609

## Chapter 18: Rose Data Modeler..............................................................................................................610
- Object Models and Data Models......................................................................................................610
- Creating a Data Model.....................................................................................................................612
- Logic in a Data Model.......................................................................................................................613
- Adding a Database..........................................................................................................................614
- Adding Tablespaces........................................................................................................................615
- Adding a Schema............................................................................................................................621
  - Creating a Data Model Diagram.................................................................................................622
- Creating Domain Packages and Domains.......................................................................................623
- Adding Tables.................................................................................................................................626
  - Adding Columns..........................................................................................................................628
  - Setting a Primary Key..................................................................................................................631
  - Adding Constraints......................................................................................................................631
  - Adding Triggers...........................................................................................................................633
  - Adding Indexes............................................................................................................................635
# Table of Contents

## Chapter 18: Rose Data Modeler

- Adding Stored Procedures ...................................................................................................................636
- Adding Relationships ...........................................................................................................................639
  - Adding Referential Integrity Rules ..............................................................................................642
- Working with Views .............................................................................................................................644
- Generating an Object Model from a Data Model ...............................................................................648
- Generating a Data Model from an Object Model ...............................................................................649
- Generating a Database from a Data Model ..........................................................................................651
- Updating an Existing Database ..........................................................................................................653
- Reverse Engineering a Database ........................................................................................................655
- Summary ..............................................................................................................................................656

## Chapter 19: Web Modeling ............................................................................................................................657

- Modeling a Web Application ...............................................................................................................657
  - Web Class Stereotypes ...................................................................................................................659
  - Relationships ................................................................................................................................666
- Reverse Engineering a Web Application .............................................................................................668
- Generating Code for a Web Application .............................................................................................670
- Summary ..............................................................................................................................................671

## Appendix: Getting Started with UML ..........................................................................................................672

- Building a Business Use Case Diagram .............................................................................................672
- Building a Workflow (Activity) Diagram .............................................................................................675
- Building a Use Case Diagram ..............................................................................................................679
- Building an Interaction Diagram .........................................................................................................684
- Building a Class Diagram ....................................................................................................................688
- Web Modeling .....................................................................................................................................691
- Adding Class Relationships ...............................................................................................................694
- Building a Statechart Diagram ..........................................................................................................696
- Building a Component Diagram .........................................................................................................699
- Building a Deployment Diagram .........................................................................................................701
any kind with regard to the completeness or accuracy of the contents herein and accept no liability of any kind including but not limited to performance, merchantability, fitness for any particular purpose, or any losses or damages of any kind caused or alleged to be caused directly or indirectly from this book.

Software License Agreement: Terms and Conditions

The media and/or any online materials accompanying this book that are available now or in the future contain programs and/or text files (the "Software") to be used in connection with the book. SYBEX hereby grants to you a license to use the Software, subject to the terms that follow. Your purchase, acceptance, or use of the Software will constitute your acceptance of such terms.

The Software compilation is the property of SYBEX unless otherwise indicated and is protected by copyright to SYBEX or other copyright owner(s) as indicated in the media files (the "Owner(s)"). You are hereby granted a single–user license to use the Software for your personal, noncommercial use only. You may not reproduce, sell, distribute, publish, circulate, or commercially exploit the Software, or any portion thereof, without the written consent of SYBEX and the specific copyright owner(s) of any component software included on this media.

In the event that the Software or components include specific license requirements or end–user agreements, statements of condition, disclaimers, limitations or warranties ("End–User License"), those End–User Licenses supersede the terms and conditions herein as to that particular Software component. Your purchase, acceptance, or use of the Software will constitute your acceptance of such End–User Licenses.

By purchase, use or acceptance of the Software you further agree to comply with all export laws and regulations of the United States as such laws and regulations may exist from time to time.

Software Support

Components of the supplemental Software and any offers associated with them may be supported by the specific Owner(s) of that material, but they are not supported by SYBEX. Information regarding any available support may be obtained from the Owner(s) using the information provided in the appropriate read.me files or listed elsewhere on the media.

Should the manufacturer(s) or other Owner(s) cease to offer support or decline to honor any offer, SYBEX bears no responsibility. This notice concerning support for the Software is provided for your information only. SYBEX is not the agent or principal of the Owner(s), and SYBEX is in no way responsible for providing any support for the Software, nor is it liable or responsible for any support provided, or not provided, by the Owner(s).

Warranty

SYBEX warrants the enclosed media to be free of physical defects for a period of ninety (90) days after purchase. The Software is not available from SYBEX in any other form or media than that enclosed herein or posted to http://www.sybex.com/. If you discover a defect in the media during this warranty period, you may obtain a replacement of identical format at no charge by sending the defective media, postage prepaid, with proof of purchase to:

SYBEX Inc.
Product Support Department
1151 Marina Village Parkway
Alameda, CA 94501
After the 90–day period, you can obtain replacement media of identical format by sending us the defective disk, proof of purchase, and a check or money order for $10, payable to SYBEX.

Disclaimer

SYBEX makes no warranty or representation, either expressed or implied, with respect to the Software or its contents, quality, performance, merchantability, or fitness for a particular purpose. In no event will SYBEX, its distributors, or dealers be liable to you or any other party for direct, indirect, special, incidental, consequential, or other damages arising out of the use of or inability to use the Software or its contents even if advised of the possibility of such damage. In the event that the Software includes an online update feature, SYBEX further disclaims any obligation to provide this feature for any specific duration other than the initial posting.

The exclusion of implied warranties is not permitted by some states. Therefore, the above exclusion may not apply to you. This warranty provides you with specific legal rights; there may be other rights that you may have that vary from state to state. The pricing of the book with the Software by SYBEX reflects the allocation of risk and limitations on liability contained in this agreement of Terms and Conditions.

Shareware Distribution

This Software may contain various programs that are distributed as shareware. Copyright laws apply to both shareware and ordinary commercial software, and the copyright Owner(s) retains all rights. If you try a shareware program and continue using it, you are expected to register it. Individual programs differ on details of trial periods, registration, and payment. Please observe the requirements stated in appropriate files.

Copy Protection

The Software in whole or in part may or may not be copy–protected or encrypted. However, in all cases, reselling or redistributing these files without authorization is expressly forbidden except as specifically provided for by the Owner(s) therein.

Acknowledgments

A great deal of effort goes into writing a book. While some of this work is done by the authors, a lot of it is done by a whole team of people. We would like to thank everyone involved in this book. Thanks to Richard Mills and Jordan Gold at Sybex for making it possible, and to Tom Cirtin, who was instrumental in getting the book ready for publication. Thanks to Eric Aker for performing the technical review. Thanks to the editorial and production team at Sybex: Mae Lum, Donna Crossman, Jill Niles, Christine Detlefs, Kevin Ly, and Tony Jonick. Thanks to indexer Nancy Guenther and thanks to the proofreaders: Emily Hsuan, Nelson Kim, Yariv Rabinovitch, and Nancy Riddiough. We couldn't have done it without all of you.
Chapter 1: Introduction to UML

The pace of business is getting faster and faster, with a greater need to compete and sustain a market. In this age of e-commerce, e-business, e-tailing, and other e's, "traditional" system development just doesn't cut it anymore. Systems now must be developed in "Internet time." Also, this faster pace has increased the need for flexible systems. Before, a user could send a request to the data-processing center and wait two years for a change. Now a user sends a request for change to the IT department and demands it in two weeks! Six-week development cycles, demanding managers, demanding users, and even the concept of XP (extreme programming) drive this point: System changes must happen fast!

This is where the Unified Modeling Language (UML) enters the picture. UML is the industry-standard modeling notation for object-oriented systems, and is the premiere platform for rapid application development. In this chapter, we describe how UML came into being, introduce the concepts of object-oriented programming, and show you how to use UML to structure your applications.

- Learning about the object-oriented paradigm and visual modeling
- Exploring types of graphical notation
- Looking at types of UML diagrams
- Developing software using visual modeling

Introduction to the Object-Oriented Paradigm

Structured programming was the mainstream in the earlier days of software engineering. Programmers began developing standard blocks of code to perform operations like printing, and then copied and pasted that code into every application they wrote. While this reduced the development time for new applications, it was difficult if a change was needed in that block of code, because the developer had to make the change everywhere that code had been copied. Structured programming presented some challenges for which object-oriented programming was designed to solve.

With object-oriented programming, developers create blocks of code, called objects. These objects are then used by the various applications. Should one of the objects require modification, a developer needs to make the change only once. Companies are rushing out to adopt this technology and integrate it into their existing applications. In fact, most applications being developed today are object-oriented. Some languages, such as Java, require an object-oriented structure. But what does it mean?

The object-oriented paradigm is a different way of viewing applications. With the object-oriented approach, you divide an application into many small chunks, or objects, that are fairly independent of one another. You can then build the application by piecing all of these objects together. Think of it as building a castle out of blocks. The first step is to make or buy some basic objects, the different types of blocks. Once you have these building blocks, you can put them together to make your castle. Once you build or buy some basic objects in the computer world, you can simply put them together to create new applications.
In the world of structured programming, to create a form with a list box, for example, you would need to write voluminous code: the code to create the form itself, the code to create and populate the list box, and the code to create an OK button that will accept the value in the list box. With object-oriented programming, on the other hand, you simply need to use three (typically prebuilt) objects: a form, a list box, and an OK button. The exercise of coding used to be along the lines of "create from scratch, but copy whatever you can find from old programs to save some time." The newer paradigm is "put together a bunch of objects, and then just focus on what's unique to this particular application."

One of the primary advantages of the object-oriented paradigm is the ability to build components once and then use them over and over again. Just as you can reuse a toy building block in a castle or a house, you can reuse a basic piece of object-oriented design and code in an accounting system, an inventory system, or an order-processing system.

So, how is this object-oriented paradigm different from the traditional approach to development? Traditionally, the approach to development has been to concern ourselves with the information that the system will maintain. With this approach, we ask the users what information they will need, design databases to hold the information, provide screens to input the information, and print reports to display the information. In other words, we focus on the information and pay less attention to what is done with the information or the behavior of the system. This approach is called data-centric and has been used to create thousands of systems over the years.

Data-centric modeling is great for database design and capturing information, but taking this approach when designing business applications presents some problems. One major challenge is that the requirements for the system will change over time. A system that is data-centric can handle a change to the database very easily, but a change to the business rules or to the behavior of the system is not so easy to implement.

The object-oriented paradigm has been developed in response to this problem. With the object-oriented approach, we focus on both information and behavior. Accordingly, we now can develop systems that are resilient and flexible to changes in information and/or behavior.

The benefit of flexibility can be realized only by designing an object-oriented system well. This requires knowledge of some principles of object orientation: encapsulation, inheritance, and polymorphism.

**Encapsulation**

In object-oriented systems, we combine a piece of information with the specific behavior that acts upon that information. Then we package these into an object. This is referred to as encapsulation. Another way to look at encapsulation is that we divide the application into small parts of related functionality. For example, we have information relating to a bank account, such as the account number, balance, customer name, address, account type, interest rate, and opening date. We also have behavior for a bank account: open, close, deposit, withdraw, change type, change customer, and change address. We encapsulate this information and behavior together into an account object. As a result, any changes to the banking system regarding accounts can simply be implemented in the account object. It works like a one-stop shop for all account information and behavior.

Another benefit of encapsulation is that it limits the effects of changes to the system. Think of a system as a body of water and the requirement change as a big rock. You drop the rock into the water and—SPLASH!—big waves are created in all directions. They travel throughout the lake, bounce off the shore, reverberate, and collide with other waves. In fact, some of the water may even splash over the shore and out of the lake. In other words, the rock hitting the water caused a huge ripple effect. But if we encapsulate our lake by dividing it into smaller bodies of water with barriers between them, then the requirement change hits the system—SPLASH! As before, waves are created in all directions. But the waves
can only go as far as one of the barriers, and then they stop. So, by encapsulating the lake, we have limited the ripple effect of dropping the rock in, as shown in Figure 1.1.

Figure 1.1: Encapsulation: Lake model

Let's apply this idea of encapsulation to the banking system. Recently, the bank management decided that if the customer has a credit account at the bank, the credit account could be used as an overdraft for their checking account. In a nonencapsulated system, we begin with a shotgun approach to impact analysis. Basically, we do not know where all of the uses of withdraw functionality are in the system, so we have to look everywhere. When we find it, we have to make some changes to incorporate this new requirement. If we're really good, we probably found about 80 percent of the uses of withdraw within the system. With an encapsulated system, we do not need to use the shotgun approach to analysis. We look at a model of our system and simply find where the withdrawal behavior was encapsulated. After locating the functionality in the account, we make our requirement change once, only in that object, and our task is complete! As you can see in Figure 1.2, only the Account class needs to change.

A concept similar to encapsulation is information hiding. Information hiding is the ability to hide the murky details of an object from the outside world. To an object, the outside world means anything outside of itself, even though that outside world includes the rest of the system. Information hiding provides the same benefit as encapsulation: flexibility. We will explore this concept more in Chapter 6, “Classes and Packages.”

Figure 1.2: Encapsulation: Banking model

**Inheritance**

*Inheritance* is the second of the fundamental object-oriented concepts. No, it has nothing to do with the million dollars you're leaving for little Johnny. It has more to do with the nose you got from your father or mother. In object-oriented systems, inheritance is a mechanism that lets you create new objects based on old ones: The *child* object inherits the qualities of a *parent* object.
You can see examples of inheritance in the natural world. There are hundreds of different types of mammals: dogs, cats, humans, whales, and so on. Each of these has certain characteristics that are unique and certain characteristics that are common to the whole group, such as having hair, being warm-blooded, and nurturing their young. In object-oriented terms, there is a mammal object that holds the common characteristics. This object is the parent of the child objects cat, dog, human, whale, etc. The dog object inherits the characteristics of the mammal object, and has some additional dog characteristics of its own, such as running in circles and slobbering. The object-oriented paradigm has borrowed this idea of inheritance from the natural world, as shown in Figure 1.3, so we can apply the same concept to our systems.

Figure 1.3: Inheritance: Natural model

One of the major benefits of inheritance is ease of maintenance. When something changes that affects all mammals, only the parent object needs to change—the child objects will automatically inherit the changes. If mammals were suddenly to become cold-blooded, only the mammal object would need to change. The cat, dog, human, whale, and other child objects would automatically inherit the new, cold-blooded characteristic of mammals.

In an object-oriented system, an example of inheritance might be in the windows. Say we have a large system with 125 windows. One day, a customer requests a disclaimer message on all of the windows. In a system without inheritance, we now have the rather tedious task of going into each one of the 125 windows and making the change. If our system were object-oriented, we would have inherited all of the windows from a common parent. Now, all we need to do is go into the parent and make the change once. All of the windows will automatically inherit the change, as shown in Figure 1.4.

Figure 1.4: Inheritance: Window model

In a banking system, we might use inheritance for the different types of accounts we have. Our hypothetical bank has four different types of accounts: checking, savings, credit card, and certificates of deposit. These different types of accounts have some similarities. Each one has an account number, interest rate, and owner. So, we can create a parent object called account to hold the common characteristics of all the accounts. The child objects can have their own unique characteristics in addition to the inherited ones. The credit account, for example, will also have a credit limit and minimum payment amount. The certificate of deposit will also have a maturity date. Changes to the parent will affect all children, but the children are free to adapt without
disturbing each other or their parents.

**Polymorphism**

The third principle of object orientation is *polymorphism*. The dictionary defines it as the occurrence of different forms, stages, or types. Polymorphism means having many forms or implementations of a particular functionality. As with inheritance, polymorphism can be seen in the natural world. Given the command, or function, of "Speak!" a human may reply, "How do you do?" The dog may reply "Woof!" The cat may reply "Meow!" but will probably just ignore you.

In terms of an object–oriented system, this means that we can have many implementations of a particular functionality. For example, we might be building a graphic drawing system. When the user wants to draw something, be it a line, circle, or rectangle, the system issues a draw command. The system is comprised of many types of shapes, each of which contains the behavior to draw itself. So, when the user wants to draw a circle, the circle object's draw command is invoked. By using polymorphism, the system figures out as it is running which type of shape is being drawn. Without polymorphism, the code for the draw function might look like this:

```java
Function Shape.drawMe()
{
    SWITCH Shape.Type
    Case "Circle"
        Shape.drawCircle();
    Case "Rectangle"
        Shape.drawRectangle();
    Case "Line"
        Shape.drawLine();
    END SWITCH
}
```

With polymorphism, the code for draw would just call a drawMe() function for the object being drawn, as in this example:

```java
Function draw()
{
    Shape.drawMe();
}
```

Each shape (circle, line, rectangle, etc.) would then have a drawMe() function to draw the particular shape.

One of the benefits of polymorphism, as with the other principles of object orientation, is ease of maintenance. What happens, for example, when the application now needs to draw a triangle? In the nonpolymorphic case, a new drawTriangle() function has to be added to the shape object. Also, the drawMe() function of the shape object has to be changed to accommodate the new type of shape. With polymorphism, we create a new triangle object with a drawMe() function to draw itself. The draw() function that initiates the drawing operation does not have to change at all.

**What Is Visual Modeling?**

If you were building a new addition to your house, you probably wouldn't start by just buying a bunch of wood and nailing it together until it looks about right. Similarly, you'd be more than a little concerned if the