

SECOND EDITION

# Object-Oriented Systems Analysis And Design **Using UML**

Simon Bennett, Steve McRobb and Ray Farmer



Mc  
Graw  
Hill

# Contents

|                                                      |                  |
|------------------------------------------------------|------------------|
| <b>Preface</b>                                       | <i>page</i> xiii |
| <b>Acknowledgements</b>                              | xix              |
| <br>                                                 |                  |
| <b>1 Information Systems—What are They?</b>          | <b>1</b>         |
| 1.1 Introduction                                     | 1                |
| 1.2 Information Systems in Practice                  | 3                |
| 1.3 General Systems Theory                           | 5                |
| 1.4 Information and Information Systems              | 14               |
| 1.5 Strategies for Success                           | 18               |
| 1.6 Summary                                          | 22               |
| <br>                                                 |                  |
| <b>2 Problems in Information Systems Development</b> | <b>25</b>        |
| 2.1 Introduction                                     | 25               |
| 2.2 What are the Problems?                           | 26               |
| 2.3 Why Things Go Wrong                              | 33               |
| 2.4 The Ethical Dimension                            | 38               |
| 2.5 Costs of Failure                                 | 41               |
| 2.6 Summary                                          | 42               |
| <br>                                                 |                  |
| <b>3 Avoiding the Problems</b>                       | <b>45</b>        |
| 3.1 Introduction                                     | 45               |
| 3.2 Project Life Cycles                              | 47               |
| 3.3 Managing Information System Development          | 55               |
| 3.4 User Involvement                                 | 56               |
| 3.5 Methodological Approaches                        | 57               |
| 3.6 CASE                                             | 57               |
| 3.7 Summary                                          | 61               |
| <br>                                                 |                  |
| <b>4 What is Object-Orientation?</b>                 | <b>63</b>        |
| 4.1 Introduction                                     | 63               |
| 4.2 Basic Concepts                                   | 64               |
| 4.3 The Origins of Object-Orientation                | 77               |
| 4.4 Object-Oriented languages Today                  | 80               |
| 4.5 Summary                                          | 82               |
| <br>                                                 |                  |
| <b>A1 Agate Ltd Case Study—Introduction</b>          | <b>84</b>        |
| A1.1 Introduction to Agate                           | 84               |
| A1.2 Existing Computer Systems                       | 85               |
| A1.3 Business Activities in the Current System       | 85               |
| A1.4 Summary of requirements                         | 86               |

|           |                                                   |            |
|-----------|---------------------------------------------------|------------|
| <b>B1</b> | <b>FoodCo Ltd Case Study—Introduction</b>         | <b>88</b>  |
| B1.1      | Introduction to FoodCo                            | 88         |
| B1.2      | FoodCo Today                                      | 91         |
| B1.3      | The Proposal                                      | 94         |
| <b>5</b>  | <b>Modelling Concepts</b>                         | <b>96</b>  |
| 5.1       | Introduction                                      | 96         |
| 5.2       | Models and diagrams                               | 97         |
| 5.3       | Drawing Activity Diagrams                         | 105        |
| 5.4       | A Development Process                             | 109        |
| 5.5       | Summary                                           | 116        |
| <b>6</b>  | <b>Requirements Capture</b>                       | <b>118</b> |
| 6.1       | Introduction                                      | 118        |
| 6.2       | User Requirements                                 | 119        |
| 6.3       | Fact Finding Techniques                           | 122        |
| 6.4       | User Involvement                                  | 131        |
| 6.5       | Documenting Requirements                          | 133        |
| 6.6       | Use Cases                                         | 134        |
| 6.7       | Requirements Capture and Modelling                | 143        |
| 6.8       | Summary                                           | 144        |
| <b>A2</b> | <b>Agate Ltd Case Study—Requirements Model</b>    | <b>149</b> |
| A2.1      | Introduction                                      | 149        |
| A2.2      | Requirements List                                 | 149        |
| A2.3      | Actors and Use Cases                              | 150        |
| A2.4      | Glossary                                          | 156        |
| A2.5      | Initial Architecture                              | 157        |
| A2.6      | Activities of Requirements Modelling              | 157        |
| <b>7</b>  | <b>Requirements analysis</b>                      | <b>160</b> |
| 7.1       | Introduction                                      | 160        |
| 7.2       | What Must a Requirements Model Do?                | 161        |
| 7.3       | Use Case Realization                              | 161        |
| 7.4       | The Class Diagram                                 | 168        |
| 7.5       | Drawing a Class Diagram                           | 176        |
| 7.6       | CRC (Class Responsibility Collaboration) Cards    | 190        |
| 7.7       | Assembling the Analysis Class Diagram             | 194        |
| 7.8       | Summary                                           | 195        |
| <b>A3</b> | <b>Agate Ltd Case Study—Requirements Analysis</b> | <b>200</b> |
| A3.1      | Introduction                                      | 200        |
| A3.2      | Use Case Realizations                             | 200        |
| A3.3      | Assembling the Analysis Class Diagram             | 206        |
| A3.4      | Activities of Requirements Analysis               | 207        |
| <b>8</b>  | <b>Refining the Requirements Model</b>            | <b>209</b> |
| 8.1       | Introduction                                      | 209        |
| 8.2       | Component-based Development                       | 210        |
| 8.3       | Adding Further Structure                          | 214        |

|           |                                                 |            |
|-----------|-------------------------------------------------|------------|
| 8.4       | Software Development Patterns                   | 223        |
| 8.5       | Summary                                         | 228        |
| <b>9</b>  | <b>Object Interaction</b>                       | <b>231</b> |
| 9.1       | Introduction                                    | 231        |
| 9.2       | Object Interaction and Collaboration            | 232        |
| 9.3       | Interaction Sequence Diagrams                   | 234        |
| 9.4       | Collaboration Diagrams                          | 245        |
| 9.5       | Model Consistency                               | 248        |
| 9.6       | Summary                                         | 249        |
| <b>10</b> | <b>Specifying Operations</b>                    | <b>251</b> |
| 10.1      | Introduction                                    | 251        |
| 10.2      | The Role of Operation Specifications            | 252        |
| 10.3      | Contracts                                       | 253        |
| 10.4      | Describing Operation Logic                      | 254        |
| 10.5      | Object Constraint Language                      | 265        |
| 10.6      | Creating an Operation Specification             | 268        |
| 10.7      | Summary                                         | 270        |
| <b>11</b> | <b>Specifying Control</b>                       | <b>272</b> |
| 11.1      | Introduction                                    | 272        |
| 11.2      | States and Events                               | 273        |
| 11.3      | Basic Notation                                  | 275        |
| 11.4      | Further Notation                                | 279        |
| 11.5      | Preparing a Statechart                          | 282        |
| 11.6      | Consistency Checking                            | 286        |
| 11.7      | Quality Guidelines                              | 288        |
| 11.9      | Summary                                         | 288        |
| <b>A4</b> | <b>Agate Ltd Case Study—Further Analysis</b>    | <b>290</b> |
| A4.1      | Introduction                                    | 290        |
| A4.2      | Sequence Diagrams                               | 291        |
| A4.3      | Statecharts                                     | 292        |
| A4.4      | Operation Specifications                        | 293        |
| A4.5      | Further Refinement of the Class Diagram         | 295        |
| A4.6      | Further Activities of Requirements Analysis     | 297        |
| <b>12</b> | <b>Moving Into Design</b>                       | <b>300</b> |
| 12.1      | Introduction                                    | 300        |
| 12.2      | How is Design Different from Analysis?          | 301        |
| 12.3      | Logical and Physical Design                     | 304        |
| 12.4      | System Design and Detailed Design               | 305        |
| 12.5      | Qualities and Objectives of Analysis and Design | 310        |
| 12.6      | Measurable Objectives in Design                 | 316        |
| 12.7      | Planning for Design                             | 317        |
| 12.8      | Summary                                         | 318        |

|           |                                              |            |
|-----------|----------------------------------------------|------------|
| <b>13</b> | <b>System Design</b>                         | <b>321</b> |
| 13.1      | Introduction                                 | 321        |
| 13.2      | The Major Elements of System Design          | 322        |
| 13.3      | Software Architecture                        | 323        |
| 13.4      | Concurrency                                  | 336        |
| 13.5      | Processor Allocation                         | 338        |
| 13.6      | Data Management Issues                       | 339        |
| 13.7      | Development Standards                        | 340        |
| 13.8      | Prioritizing Design Trade-offs               | 341        |
| 13.9      | Design for Implementation                    | 342        |
| 13.10     | Summary                                      | 342        |
| <b>14</b> | <b>Object Design</b>                         | <b>344</b> |
| 14.1      | Introduction                                 | 344        |
| 14.2      | Class Specification                          | 346        |
| 14.3      | Interfaces                                   | 351        |
| 14.4      | Criteria for Good Design                     | 352        |
| 14.5      | Designing Associations                       | 356        |
| 14.6      | Integrity Constraints                        | 361        |
| 14.7      | Designing Operations                         | 364        |
| 14.8      | Normalization                                | 365        |
| 14.9      | Summary                                      | 366        |
| <b>15</b> | <b>Design Patterns</b>                       | <b>368</b> |
| 15.1      | Introduction                                 | 368        |
| 15.2      | Software Development Patterns                | 369        |
| 15.3      | Documenting Patterns—Pattern Templates       | 371        |
| 15.4      | Design Patterns                              | 372        |
| 15.5      | How to Use Design Patterns                   | 382        |
| 15.6      | Benefits and Dangers of Using Patterns       | 384        |
| 15.7      | Summary                                      | 385        |
| <b>16</b> | <b>Human–Computer Interaction</b>            | <b>387</b> |
| 16.1      | Introduction                                 | 387        |
| 16.2      | The User Interface                           | 388        |
| 16.3      | Approaches to User Interface Design          | 396        |
| 16.4      | Standards and Legal Requirements             | 405        |
| 16.5      | Summary                                      | 407        |
| <b>17</b> | <b>Designing Boundary Classes</b>            | <b>409</b> |
| 17.1      | Introduction                                 | 409        |
| 17.2      | The Architecture of the Presentation Layer   | 410        |
| 17.3      | Prototyping the User Interface               | 412        |
| 17.4      | Designing Classes                            | 415        |
| 17.5      | Designing Interaction with Sequence Diagrams | 419        |
| 17.6      | The Class Diagram Revisited                  | 427        |
| 17.7      | User Interface Design Patterns               | 427        |
| 17.8      | Modelling the Interface Using Statecharts    | 430        |
| 17.9      | Summary                                      | 438        |

|           |                                                      |            |
|-----------|------------------------------------------------------|------------|
| <b>18</b> | <b>Data Management Design</b>                        | <b>440</b> |
| 18.1      | Introduction                                         | 440        |
| 18.2      | Persistence                                          | 441        |
| 18.3      | File Systems                                         | 444        |
| 18.4      | Database Management Systems                          | 450        |
| 18.5      | Designing for Relational Database Management Systems | 454        |
| 18.6      | Designing for Object Database Management Systems     | 462        |
| 18.7      | Distributed Databases                                | 464        |
| 18.8      | Designing Data Management Classes                    | 466        |
| 18.9      | Summary                                              | 475        |
| <b>A5</b> | <b>Agate Ltd Case Study—Design</b>                   | <b>477</b> |
| A5.1      | Introduction                                         | 477        |
| A5.2      | Architecture                                         | 477        |
| A5.3      | Sample Use Case                                      | 479        |
| A5.4      | Class Diagrams                                       | 479        |
| A5.5      | Sequence Diagrams                                    | 483        |
| A5.6      | Database Design                                      | 486        |
| A5.7      | Statecharts                                          | 486        |
| A5.8      | Activities of Design                                 | 486        |
| <b>19</b> | <b>Implementation</b>                                | <b>490</b> |
| 19.1      | Introduction                                         | 490        |
| 19.2      | Software Implementation                              | 491        |
| 19.3      | Component Diagrams                                   | 495        |
| 19.4      | Deployment Diagrams                                  | 497        |
| 19.5      | Software Testing                                     | 499        |
| 19.6      | Data Conversion                                      | 502        |
| 19.7      | User Documentation and Training                      | 504        |
| 19.8      | Implementation Strategies                            | 505        |
| 19.9      | Review and Maintenance                               | 507        |
| 19.10     | Summary                                              | 511        |
| <b>20</b> | <b>Reusable Components</b>                           | <b>513</b> |
| 20.1      | Introduction                                         | 513        |
| 20.2      | Why Reuse?                                           | 514        |
| 20.3      | Planning a Strategy for Reuse                        | 520        |
| 20.4      | Commercially Available Componentware                 | 522        |
| 20.5      | Case Study Example                                   | 524        |
| 20.6      | Summary                                              | 531        |
| <b>21</b> | <b>Managing Object-Oriented Projects</b>             | <b>533</b> |
| 21.1      | Introduction                                         | 533        |
| 21.2      | Resource Allocation and Planning                     | 534        |
| 21.3      | Managing Iteration                                   | 541        |
| 21.4      | Dynamic Systems Development Method                   | 541        |
| 21.5      | Extreme Programming                                  | 545        |
| 21.6      | Software Metrics                                     | 547        |
| 21.7      | Process Patterns                                     | 549        |
| 21.8      | Legacy Systems                                       | 549        |

|           |                                                          |            |
|-----------|----------------------------------------------------------|------------|
| 21.9      | Introducing Object Technology                            | 550        |
| 21.10     | Summary                                                  | 551        |
| <b>22</b> | <b>Systems Development Methodologies</b>                 | <b>554</b> |
| 22.1      | Introduction                                             | 554        |
| 22.2      | 'Method' and 'Methodology'                               | 555        |
| 22.3      | Why Use a Methodology?                                   | 557        |
| 22.4      | A Brief Historical Review                                | 558        |
| 22.5      | The Unified Software Development Process                 | 560        |
| 22.6      | Participative Design Approaches                          | 564        |
| 22.7      | Issues in Choosing a Methodology                         | 566        |
| 22.8      | Hard <i>versus</i> Soft Methodologies                    | 567        |
| 22.9      | Summary                                                  | 569        |
|           | <b>Appendix A Notation Summary</b>                       | <b>572</b> |
|           | <b>Appendix B Selected Solutions and Answer Pointers</b> | <b>578</b> |
|           | <b>Glossary</b>                                          | <b>590</b> |
|           | <b>Bibliography</b>                                      | <b>597</b> |
|           | <b>Index</b>                                             | <b>605</b> |