

# Preface

---

## Background to the Book

The Faculty of Computing Sciences and Engineering at De Montfort University began teaching object-oriented systems analysis and design to first-year students on undergraduate programmes in Business Information Systems and Multimedia Computing in 1997, having taught structured approaches for many years. Object-oriented approaches to analysis, design and programming are now taught on all diploma, undergraduate and postgraduate programmes in Computer Science and Information Systems. At that time, we were all working at De Montfort University and wanted a book to support our teaching, one that put the analysis and design activities in the context of the whole systems life cycle (whatever kind of life cycle that is), and that included generic analysis and design issues, such as fact finding. Most books on object-oriented approaches to analysis and design concentrate on object-orientation and on the notation.

Another concern with many textbooks that we have used is that they do not employ a consistent case study as a source of examples throughout the book. We try to teach or train by means of practical case studies so that students experience something that is close to the development of a real system rather than a series of disjointed exercises. We particularly like the approach taken by Barbara Robinson and Mary Prior in their book *Systems Analysis Techniques* (1995). Throughout the book they use two case studies, one for examples and one for exercises. However, their book teaches a structured approach using the notation of Structured Systems Analysis and Design Method (SSADM).

When McGraw-Hill offered us the opportunity to write our own book, we decided to take a similar approach and have developed two case studies through the course of the text, although not to the same level of detail.

Since the publication of the first edition in 1999, many things have changed. Only one of the authors is still at De Montfort University; another left the University to become an Information Systems Consultant for Ericsson Intracom, and is now working for Ceesio AG as a Systems Architect in Corporate IT; the third now works as Associate Dean in the School of Mathematical and Information Sciences at Coventry University. As well as our careers, our ideas have developed, helped in large part by all the feedback and reviews that we have received. UML itself has also changed, most recently with the introduction of Version 2.0, which prompted this new edition, and has also become more widely accepted as the common language for systems modelling. But we still believe in the value for teaching and learning of basing the book around a consistent thread of case study material, and have therefore retained this approach in this new edition.

## Who Should Read this Book?

The three authors of this book believe that systems analysis and design are activities that should take place in the context of the organizations that will use the

information systems that are the result. The examples we use are based on business organizations, but they could be any kind of organization in the public or voluntary (not for profit) sectors, and the approach we adopt is suitable for most kinds of information system including real-time systems. The book starts with three chapters that set the development of information systems in this context.

We expect most of our readers to be students undertaking a Diploma, Bachelor's or Master's course in a computing or information systems subject. It will also be of relevance to some students of other subjects, such as business studies, who want to understand how business information systems are developed without wanting to be programmers, analysts or designers.

The book is also suitable for professionals in computing and information systems, many of whom began their professional career before the advent of object-oriented development techniques and who want to upgrade their skills by learning about object-oriented analysis and design. We have used the Unified Modelling Language (UML), which is the *de facto* standard notation for object-oriented development.

### Case Studies

In our teaching and training we use case studies as the basis of tutorials and practical work. We also use the same case studies to provide examples in taught material and in student assessments. We believe that it is important that students see analysis and design as a coherent process that goes from initial fact-finding through to implementation, and not as a series of disjointed exercises. This book uses two practical case studies. The first of these, Agate Ltd, is an advertising company. Agate is used for examples in most of the chapters of the book that explain techniques and for some exercises. The second case study, FoodCo Ltd, is a grower of fruit and vegetables and a manufacturer of packaged foods. FoodCo is used for most of the exercises that are included in chapters for the reader to complete.

The two case studies are introduced in short chapters (A1 and B1) that can be found between Chapters 4 and 5. In these first two case study chapters, we provide background material about the two companies and explain some of their requirements for computerized information systems. Chapter A2 (between Chapters 6 and 7) presents examples of the requirements model, while Chapter A3 (between Chapters 7 and 8) brings together some examples from the first version of the analysis model for Agate's new system. Chapter A4 (between Chapters 11 and 12) presents examples from the analysis model after it has been further developed. Chapter A5 (between Chapters 18 and 19) brings together some examples from the design model for the new system. We do not provide models for the FoodCo case study, but FoodCo forms the basis of most of the practical exercises at the end of chapters. A few partial solutions are provided where they are required as the basis for later exercises.

If you are using this book as a teacher, you are welcome to use these materials and to develop them further as the basis of practical exercises and assessments. Some exercises that you may want to use are provided in each chapter. Further models, solutions and case studies are provided on the book's website, and this will continue to develop over time.

## Exercises for Readers

Each chapter contains two kinds of exercises for readers. First we provide Review Questions. The aim of these is to allow you to check that you have understood the material in the chapter or section of a chapter that you have just read. Most of these Review Questions should only take a few minutes at most to complete. Solutions to some of these questions are to be found at the back of the book. The answers to many of them are to be found in the text. Some require you to apply the techniques you have learned, and the answers to these are not always provided.

At the end of each chapter are Case Study Work, Exercises and Projects. These are exercises that will take longer to complete. Some are suitable to be used as tutorial exercises or homework, some could be used as assignments, and some are longer projects that could be developed over a matter of weeks. We have provided answer pointers to some of these exercises to help you.

Solutions to more of these exercises are available via the book's website to bona fide university and college teachers who adopt the book as the set text for their courses. Please contact McGraw-Hill for details of the support materials.

## Structure of the Book

Although we have not formally divided the book into sections, there are four parts to the book, each of which has a different focus.

### *Part 1*

Chapters 1 to 4 provide the background to information systems analysis and design and to object-orientation. In the first three of these chapters we explain why analysis and design are important in the development of computerized systems and introduce fundamental concepts such as those of systems theory. Chapter 4 introduces some of the ideas of object-orientation that will be developed in the second part.

### *Part 2*

The second part of the book begins with the first two case study chapters (A1 and B1) and includes Chapters 5 to 11. The focus of this part of the book is on the activities of requirements gathering and systems analysis and the basic notation of the Unified Modelling Language (UML). In it we introduce use cases, class diagrams, sequence diagrams, communication diagrams, activity diagrams, state-chart diagrams and the Object Constraint Language (OCL). Chapter 5, which was added for the second edition, discusses models and diagrams and presents one of the UML diagramming techniques—activity diagrams—as an example. In Chapter 5 we also provide an overview of the way the UML techniques fit together in the iterative development life cycle. This part includes three case study chapters (A2, A3 and A4), which illustrate the development of the UML models as the analysis activities progress. The purpose of these case study chapters is to illustrate the development of models as the analysis and design progress. We do not have the space in the book to provide all the analysis and design documentation.

**Part 3**

The third part of the book is about system design. It includes Chapters 12 to 18 and concludes with examples from the design model for the Agate case study (Chapter A5). In this part we develop the use of most of the diagramming techniques introduced in Part 2. We do this by enhancing the analysis models that we have produced to take design decisions into account. This part covers system architecture, system design, design patterns, and the design of objects, user interfaces and data storage. The design model at the end of this part serves the same purpose as the one at the end of the analysis chapters. Chapter 12 is new to the third edition and covers various views of the architecture of businesses and the information systems that support them. Architecture has become more important in professional practice, and model-driven architecture (MDA) is one of the factors that has influenced the evolution of UML into Version 2.0. The former Chapters 12, 13 and 14 have been consolidated into Chapters 13 and 14.

**Part 4**

In the final part we cover the implementation of systems and the issues of how the systems life cycle is organised and how reusable components can be developed. The chapter on implementation introduces the last of the UML diagram types, component diagrams and deployment diagrams. The chapter on project management that was in the previous editions has been removed and made available on the book's website as part of an attempt to keep the page count, and hence the price, at a reasonable level.

**Pathways through the Book**

Whatever the formal structure of the book, you the reader are welcome to work through it in whatever order you like. However, the book has been written to be read sequentially and not used as a reference, as one critical reviewer pointed out on Amazon. One of the authors has written another book on UML, also published by McGraw-Hill, that can more easily be used as a reference book.

We have taught several undergraduate analysis and design modules. These include a first-year module on analysis and design, with the emphasis on the life-cycle and on analysis; and a second-year module with an emphasis on design and the use of methodologies. We would expect to cover the following chapters in each module.

Analysis module—Chapters 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. (Statechart diagrams are omitted from this so that students are not overloaded with different diagramming notations in the first year.)

Design module—Chapters 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 (available from the book website).

We have tried to group together techniques under the general headings of analysis and design, even though many of them are used throughout the life cycle. This does not mean that we necessarily advocate following a life cycle model that treats analysis and design as separate phases. We suggest an iterative life cycle in which the models of the system are progressively elaborated and in which analysis and design are interwoven. However, we do believe that analysis and design are separate activities, even if they are not separate stages in a project's life cycle. We also think that it is easier to learn analysis and design as separate activities rather

than merged together. (This is like a chef being taught to cook desserts and main courses separately. Later this chef can either follow a structured approach to cooking, in which a dessert is prepared in advance and then chilled and stored until required, and the main course is then cooked separately. Or, when she is experienced, she can take an iterative approach and progressively build the meal, switching from main course to dessert and back again.)

If you plan to use this book for a course that concentrates on UML, then you may want to use the following path through the book—Chapters 5, 6, 7, 8, 9, 10, 11, 17, 18, 19, and including the case study chapters. The other chapters can be read as background.

If you are familiar with the general aspects of information systems development and of object-orientation, and are reading this book in order to gain an understanding of how you can use UML in analysis and design, then you can start at Chapter A1, the first of the case study chapters, and work your way through from there. If you are not familiar with object-oriented approaches, then you should also include Chapter 4.

### **Transferable Skills**

Some of the skills of the systems analyst and designer can be classified as transferable or professional skills. Most employers place a high value on these skills. Many colleges and universities provide special modules on these skills, embed them in other modules or provide self-study packages for students. We have included material on fact-finding skills, particularly interviewing and questionnaire design, within the text of the book. We have not included other skills explicitly, but there are opportunities for teachers to use the exercises to develop skills in problem solving, group work, research, report writing and oral presentation.

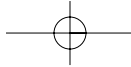
### **Web Site and Support Materials**

This book is supported by materials that we have placed on the book's website. The website is accessible at [www.mcgraw-hill.co.uk/textbooks/bennett/](http://www.mcgraw-hill.co.uk/textbooks/bennett/). The website material has been revised for the third edition. It includes self-test exercises for students. Materials for lecturers also include Microsoft PowerPoint slides to accompany each chapter, solutions to some of the exercises and copies of most of the figures from the book that can be used for teaching. If you use this material in your teaching materials, we ask that you acknowledge our copyright on the material.

We welcome feedback about the book. Some of the changes that we have made in writing the third edition have been based on feedback from lecturers and students around the world who have used our book. You can email us at [authors@OOADtext.info](mailto:authors@OOADtext.info) or write to us care of McGraw-Hill at the address at the beginning of the book.

### **Latest Version of UML**

This edition is based on UML Version 2.0. This version was due to be released in 2001, but has taken longer to produce than was anticipated. The specification is by no means perfect, and a Version 2.1 is likely to be produced in the near future, though this should only be an editorial release that will tidy up anomalies in the 2.0 specification.



### Changes in the Third Edition

The third edition has some significant changes from the second. The most significant change is in the introduction of UML 2.0. This new version has had the most significant effect on interaction diagrams: sequence diagrams, communication diagrams (formerly collaboration diagrams); and the two new diagram types, interaction overview diagrams and timing diagrams. The underlying semantics of activity diagrams has changed, although this does not have such significant effects on the notation. Artefacts have been introduced into deployment diagrams. All chapters have been revised and updated to reflect developments in the world of object-oriented analysis and design, the evolution of UML and changes in our own thinking. Chapter 12 on architecture has been added, and the material in the former Chapters 12, 13 and 14 consolidated into Chapters 13 and 14. The chapter on project management has been moved to the book's website. A summary of changes is available on the website. This is provided for lecturers who need to revise their teaching materials.

*Simon Bennett, Steve McRobb, Ray Farmer  
Leicester, UK  
February 2005*

