

Preface

Outline of the Book

The development of an *information system (IS)* – from its inception to the first release to stakeholders – comprises three iterative and incremental phases: analysis, design and implementation. This book describes the methods and techniques used in the first two phases. The implementation issues are only addressed to the extent to which they need to be considered in the design. Testing and change management are addressed in the final chapter.

The text concentrates on *object-oriented software development*. The *Unified Modeling Language (UML)* is used to capture modeling artifacts. Emphasis is placed on the *development by elaboration* where the same modeling language (UML) is used throughout the development lifecycle. Analysts, designers and programmers "speak" the same language, although perhaps use the dialects (profiles) of the language fitting their individual needs.

The early applications of object technology targeted graphical user interfaces (GUI) and focused on the speed of developing new systems and the speed of program's execution. In this book, I emphasize the application of *object technology in IS development*. The challenge is the large volume of data, complex data structures, shared access to information by many concurrent users, transaction processing, changing requirements, etc. The main advantage of object technology in IS environments is in facilitating software maintenance and scalability.

Developing information systems is synonymous with doing analysis and design "*in-the-large*". No IS project can succeed without following strict development processes and without understanding the underlying software architectures. The development is *large-scale, object-oriented, iterative and incremental*. The software architecture is based on a *client/server* solution, where the client is a workstation with GUI and the server stores a database. Client and server run in separate processes and communicate via object messaging. A server database can be relational, object-relational or purely object-oriented.

The book proposes a detailed approach to analysis and design of information systems with UML. The book identifies ways to: (1) harness the complexity of large system models, (2) improve software architectures, (3) facilitate software readability, maintainability and scalability, (4) promote layered structuring of objects, (5) handle component integration, (6) improve modeling of collaboration between GUI and persistent database objects, etc.

Distinguishing Features

This book has a number of features which – when combined – create a unique offering. The "*teach-by-example*" approach is the cornerstone of the text. The main discussion is based on examples and a guided tutorial drawn from five application domains *University Enrolment, Video Store, Contact Management, Telemarketing, and OnLine Shopping*. The examples are independent. They

build up into case studies that can be extended or diversified through questions formulated at the end of chapters ("*Exercise Questions*"). Some exercises refer to the sixth application domain – *Advertising Expenditure Measurement*.

To facilitate *self-education*, the guided tutorial (*OnLine Shopping*) and the case studies are formulated using the question-and-answer principle. A separate Section at the beginning of the book offers ***Book Activity Diagrams*** that link the question-and-answer steps used in the tutorial and in the case studies. The Book Activity Diagrams can serve as an alternative table of contents for examples scattered in the text.

The book discusses principles, methods and techniques of *good analysis* and *good design*. A special attention is paid to the design. The design is *not* treated as a straightforward transformation from analysis. The book acknowledges difficulties and intricacies of large-scale object-oriented client/server system development. In many ways, the book takes a fresh look at the "*design-in-the-large*", at iterative and incremental development of large systems, and at capabilities and limitations of tools and methods in large software production.

A unique character of the book comes from a balanced *blend of practical explanation and theoretical insight*. A major premise is to avoid unnecessary over-complication, but without the loss of rigor. The book "speaks" from experience. Topics that are not relevant to industry or that are only of research interest have been excluded.

The book is on the "*cutting-edge*" of *information technology*. It uses the latest standard in system modeling – UML. It addresses latest developments in database technology, including object-relational databases. In this context, the Internet-driven shift from "thick-clients" (i.e. large desktop computers) back to server-based computing is acknowledged. The analysis and design principles discussed in the text apply equally well to conventional client/server solutions and to modern component-based distributed applications.

Software development is not amenable to "black-white", "true-false", "zero-one" solutions. Good software solutions come from good business analysts and system designers/programmers, not from blindly applied algorithms. A policy of the book is *to warn* the reader *about potential difficulties* that the advocated approach cannot entirely resolve. The consequence is that readers will apply the learned knowledge with care and will not follow unrealistic expectations of the ease with which the approach can be applied (and may subsequently fail more dramatically than it need to).

In summary, the distinguishing features of the book are:

1. The book relates the theories to reality – in the form of practical problems and limitations, which will have to be addressed when applying the approach "in the field".
2. The book gives special attention to the design. The design is *not* treated as a straightforward transformation from analysis and it acknowledges the difficulties and intricacies of large-scale client/server system development.
3. A wealth of non-trivial examples and exercises with all solutions included in the textbook or in the supplementary materials. The Instructor's Manual is not an afterthought – it has been written concurrently with the text and is meticulous.

Intended Readership

In tune with the growing demand for the university education to be more relevant to the industry practice, the textbook is **aimed at students and practitioners** alike. This has been a difficult task but hopefully achieved. To ensure a *lasting educational benefit*, the implementation aspects of software development are discussed in non-vendor-specific terms (although commercial CASE tools have been used in illustrations and solutions).

The book is aimed at **computer science and information systems curricula**. As it contains both “high-level” system modeling topics and “low-level” user interface and database design issues, the book should be attractive to courses in *systems analysis, systems design, software engineering, databases, object technology*, and to *software project* courses that require students to develop a system following the development lifecycle: from requirements determination to GUI and database implementation. The book is designed for a one-semester course, but it can potentially be used over two one-semester courses - one on requirements analysis and the other on system design.

For the **practitioners'** audience, the presented theories are related to realities. Most problem statements, examples and exercises are drawn from the consulting practice of the author. We have adopted a policy of warning the reader of potential difficulties or limitations with advocated approaches. The following categories of practitioners are likely to benefit most from the book: *business and system analysts, designers, programmers, system architects, project leaders and managers, reviewers, testers, technical writers, and industry trainers*.

Organization of the Book




The book provides comprehensive coverage of object-oriented analysis and design of information systems. The material is presented in the order consistent with modern development processes. The book consists of **ten chapters**. The coverage is balanced between analysis and design. The first five chapters address the analysis issues, the last five – the design and related considerations.

The text should be able to accommodate readers with varying background knowledge. Two chapters in the book are dedicated to explanation of **underpinnings of analysis and design**. The remaining chapters assume that the reader understands these underpinnings. The reader has a choice of studying the "underpinnings" chapters in detail or of using them for the knowledge review only.

The book has several features to improve its clarity and to break the monotony of the text, in particular:

- *Sections* are small.
- Two highlighted frames, to mark the beginning and the end of an example, bracket the *in-text examples*. A graphical icon on the page margin also marks the beginning of each example.
- *Examples' statements and solutions* are clearly distinguished by placing the statement text in frames with colored background.

- *Page margins* are used to:
 - number and label figures and tables,
 - show graphical icons to identify examples in the text and end-of-chapter questions, as per the icon key below:

I C O N K E Y	
	Examples
	Review questions
	Exercise questions

Supplementary Materials

A comprehensive package of supplementary material is provided for the companion Web-sites. Most of the Web documents are freely available to the readers, but some material is password-protected for the benefit of instructors who have adopted the textbook in their teaching. The home page for the book is simultaneously maintained at:

<http://www.booksites.net/rasd>

<http://www.comp.mq.edu.au/rasd>

The Web package includes:

1. ***Instructor's Manual*** with:
 - "*Lecture Slides*" in Microsoft PowerPoint and Acrobat Read (.pdf) formats.
 - "*Answers & Solutions*" manual containing annotated answers and solutions to all review and exercise questions from the end of the book chapters. The organization of the manual corresponds to the textbook structure. The questions from the textbook are repeated in the manual. Answers and solutions follow the questions.
2. ***Student's Resources*** with printable lecture slides in Acrobat Read format.
3. ***Self-Education Resources*** with Rational Rose (.mdl) and PowerDesigner (.pdm) model files containing solutions to the guided tutorial, the case studies and all other modeling examples in the textbook.
4. ***Errata*** page dedicated to corrections of errors and omissions in the book.
5. ***For More Information*** page to point the reader to the latest ideas and trends in the textbook's subject area. Also, to identify courses that adopted the book and which are prepared to share their Web facilities with other readers.

Your comments, corrections, suggestions for improvements, contributions, etc. are very much appreciated. Please, direct any correspondence to:

Leszek A. Maciaszek
Department of Computing
Macquarie University ~ Sydney
NSW 2109, Australia
leszek@ics.mq.edu.au
<http://www.comp.mq.edu.au/~leszek/>
phone: +61 2 9850-9519
facsimile: +61 2 9850-9551
courier: North Ryde, Herring Road, Bld. E6A, Room 319

Acknowledgements

This book would be impossible without my interactions with friends, colleagues, students, industry gurus and all other people who, consciously or not, have shaped my knowledge in the subject area. I am truly indebted to all of them. An attempt to list all the names would be injudicious and impractical – a blanket “thank you” is everything I can do. However, even in the little space available for acknowledgements, a few “mates” must be mentioned:

- Keith Mansfield and the staff of Addison Wesley London – for recognizing the potential of this book and bringing it to the international market.
- Stephen Bills and his staff at ACNielsen AdEx, Sydney Australia, in particular: Steven Grotte, Kevin Mathie, Cameron Murray, James Rees, Jovan Spoa, and Eric Zurcher – for providing me with a production IS development test-bed for the writing of this book.
- My friends and colleagues in the Department of Computing at Macquarie University in Sydney Australia – for their support in making of this book.
- The formal (unknown to me) and informal reviewers – for invaluable feedback that has greatly benefited the book.
- The staff at Rational, Oracle and Sybase – for providing me with the CASE and database software indispensable in the development of tutorials, case studies and examples.