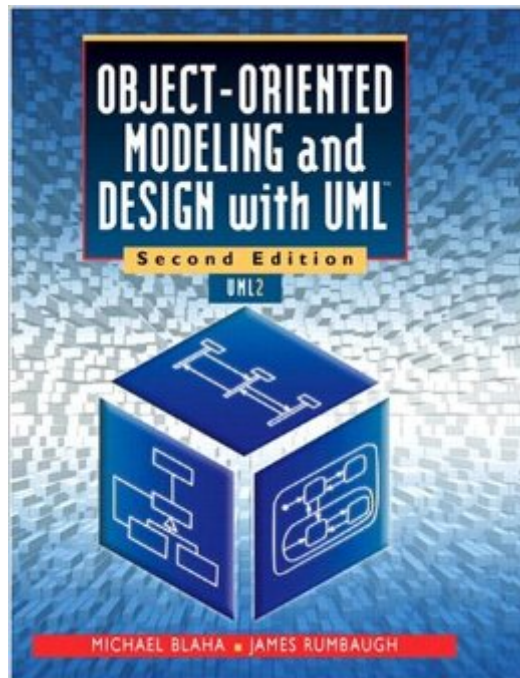# Object-Oriented Modeling And Design With UML (2nd Edition)

Synopsis

This revision offers a crisp, clear explanation of the basics of object-oriented thinking via UML models, then presents a process for applying these principles to software development, including C++, Java, and relational databases. An integrated case study threads throughout the book, illustrating key ideas as well as their application.Â

Book Information

Paperback: 496 pages

Publisher: Pearson; 2 edition (December 9, 2004)

Language: English

ISBN-10: 0130159204

ISBN-13: 978-0130159205

Product Dimensions: 7 x 1.3 x 9.1 inches

Shipping Weight: 2 pounds (View shipping rates and policies)

Average Customer Review: 3.9 out of 5 starsÂ Â See all reviewsÂ (7 customer reviews)

Best Sellers Rank: #495,507 in Books (See Top 100 in Books)   #41 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > UML   #186 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design   #670 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

One striking feature about this book is the sheer mass of exercises given for each chapter. OO modelling and designing using UML can be treated as a very high level process. Indeed it is certainly possible to do this, after reading the book. But the authors have striven to make as concrete as possible to anyone new to UML.Just as one example, consider Chapter 3, on class modelling. It has some of the ambiance of the designing of relational tables, if perhaps you are more familiar with the latter. Books on relational design often have many examples and problems, to drive home the concepts to students. Likewise here. Granted, if you do hail from a relational background, the style of object oriented modelling, and the usage of UML with which to express this, will be different from what you are used to.Later in the book, another useful feature is how you are challenged to convert pseudocode UML examples into actual code, C++ or Java. A vital skill. Since often a top down approach with UML is where UML encodes a large picture. But you then have to manually instantiate this as code. So the book gives you lots of practice.By the way - the book also has useful tips on good object oriented practices for both C++ and Java. These arose out of the

discussion about UML. But you should consider adopting these practices irrespective of whether you use UML or not.

I bought this book in hopes of learning better molding techniques from the solutions to the problems considering the authors are well know experts in the field, but there are about 2 solutions for every 10 problems and the rest you have to get it from the instructor, sorry some of us are working professionals. The book is written in typical academic style: definitions, technique and example. Best practices or real word encounters are almost none. The techniques are standard nothing spectacular or advanced. I also found that the UML notation is poorly explained, so expect to use a supplement if you don't understand the notation. If you would like to go through a solid walkthrough while learning UML notation, Analysis and modeling techniques through 3 iteration of the unified process, more like the real world, I recommend Craig Larman's Applying UML and Patterns isbn: 0131489062. if you like to learn UML notation with well explained standard modeling techniques with practice problems and solutions consider Schaum's outline UML Isbn: 0077107411 who's authors' are reputed professionals in the U.K.

UML is still a pretty new notation language which is developing as we speak. An experienced programmer may run into some small ambiguities, where there is no concrete way to do something.There are various names and notations for several things in UML. The capitalizations/small case for state diagrams may not always match up with other books, and some things can be drawn in more than one way. For instance, a pentagon to describe the entire diagram may exist or not be used at all in some cases.The discrepencies not only show the language's slow progression because of many differing opinions of language developers, but it is also a good indication of how difficult it may be to notice and understand how functions and drawings are to be interpretted between books and software usage. What's also interesting is that Rumbaugh has written various UML books, so he is aware of many of these differences. The question is whether authors as a team could only agree on a certain way to write the book, or if Rumbaugh was more in a hurry with the others to just get books done and printed to make money. If Rumbaugh comes out with a book on his own, it may be worth considering strongly! The same could probably be true for Blaha.Like many UML books probably, the book doesn't help you become aware of something like if an interface cannot be pointing to another interface. That is a more simpler example, but there are much more complex case whereas if you do not have a good grip on computer languages, it probably won't make much sense. Even though I cannot expect a book to cover most of these type

of things, I think a book can attempt to cover some of these things so that a beginner could learn more easily. Part of the problem is because the language is still developing, and people just want to get books printed. If you know your UML well, here's your chance!You can still pass a course not knowing any of the idiosyncracies, but at a graduate level, it will not be a fulfilling way to learn.The way the book goes over the content it does cover, it does it very well. There are some things missing such as differing ways to show the same things, or it does not elaborate on topics enough that one would probably need to know for a graduate level course.

Had some good stuff, but didn't help me as much as I'd hoped with my problems at work.

A tough read but then again computer science is always a tough read.

Good Product. Received in a timely manner.

Good book

Object Success : A Manager's Guide to Object-Oriented Technology And Its Impact On the Corporation (Object-Oriented Series) Object-Oriented Modeling and Design with UML (2nd Edition) Object-Oriented Analysis and Design for Information Systems: Modeling with UML, OCL, and IFML The Object-Oriented Approach: Concepts, Systems Development, and Modeling with UML, Second Edition Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition) Reusable Software : The Base Object-Oriented Component Libraries (Prentice Hall Object-Oriented Series) Object-Oriented Software Engineering: Using UML, Patterns and Java (2nd Edition) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) UML and the Unified Process: practical object-oriented analysis and design Systems Analysis and Design: An Object-Oriented Approach with UML Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach UML Distilled: A Brief Guide to the Standard Object Modeling Language (2nd Edition) Fundamentals of Object-Oriented Design in UML Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition) [Economy Edition] Object-Oriented Software Engineering: Practical Software Development Using UML and Java Object-Oriented Technology: From Diagram to Code with Visual Paradigm for UML UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition) Visual Object-Oriented Programming Using Delphi With CD-ROM (SIGS: Advances in

Object Technology) Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach Structured Finance Modeling with Object-Oriented VBA

Dmca