

Unified Language User Guide Booch

UML is the industry standard notational language. Updated to include coverage of UML 2.0, this text helps the reader master the vocabulary, rules and idioms of the UML, as well as understand what the UML is and what it is not.

Overviews the process of building and compiling executable UML models for software development. The book focuses on the BridgePoint tool suite and object action language developed by Project Technology. The authors discuss identifying system requirements, diagramming classes and attributes, constraints on the class diagram, ways of building sets of communicating statechart diagrams, and model verification. Annotation copyrighted by Book News, Inc., Portland, OR.

Use case analysis is a methodology for defining the outward features of a software system from the user's point of view. Applying Use Cases, Second Edition, offers a clear and practical introduction to this cutting-edge software development technique. Using numerous realistic examples and a detailed case study, you are guided through the application of use case analysis in the development of software systems. This new edition has been updated and expanded to reflect the Unified Modeling Language (UML) version 1.3. It also includes more complex and precise examples, descriptions of the pros and cons of various use case documentation techniques, and discussions on how other modeling approaches relate to use cases. Applying Use Cases, Second Edition, walks you through the software development process, demonstrating how use cases apply to project inception, requirements and risk analysis, system architecture, scheduling, review and testing, and documentation. Key topics include: Identifying use cases and describing actors Writing the flow of events, including basic and alternative paths Reviewing use cases for completeness and correctness Diagramming use cases with activity diagrams and sequence diagrams Incorporating user interface description and data description documents Testing architectural patterns and designs with use cases Applying use cases to project planning, prototyping, and estimating Identifying and diagramming analysis classes from use cases Applying use cases to user guides, test cases, and training material An entire section of the book is devoted to identifying common mistakes and describing their solutions. Also featured is a handy collection of documentation templates and an abbreviated guide to UML notation. You will come away from this book with a solid understanding of use cases, along with the skills you need to put use case analysis to work.

Thoroughly updated and fully compliant with Rational Rose 2002, the latest release of the industry's most popular software modeling tool, this edition contains simplified, useful case studies and helps the reader understand the core concepts of modeling and how to use UML effectively.

Schaum's Outline of UML

The Rationale Behind the Object Constraint Language

UML @ Classroom

APPLYING UML & PATTERNS 3RD EDITION

A Brief Guide to the Systems Modeling Language

The Rational Unified Process

Build server-side applications more efficiently—and improve your PHP programming skills in the process—by learning how to use design patterns in your code. This book shows you how to apply several object-oriented patterns through simple examples, and demonstrates many of them in full-fledged working applications. Learn how these reusable patterns help you solve complex problems, organize object-oriented code, and revise a big project by only changing small parts. With *Learning PHP Design Patterns*, you'll learn how to adopt a more sophisticated programming style and dramatically reduce development time. Learn design pattern concepts, including how to select patterns to handle specific problems. Get an overview of object-oriented programming concepts such as composition, encapsulation, polymorphism, and inheritance. Apply creational design patterns to create pages dynamically, using a factory method instead of direct instantiation. Make changes to existing objects or structure without having to change the original code, using structural design patterns. Use behavioral patterns to help objects work together to perform tasks. Interact with MySQL, using behavioral patterns such as Proxy and Chain of Responsibility. Explore ways to use PHP's built-in design pattern interfaces. Larman covers how to investigate requirements, create solutions and then translate designs into code, showing developers how to make practical use of the most significant recent developments. A summary of UML notation is included.

Software architecture—the conceptual glue that holds every phase of a project together for its many stakeholders—is widely recognized as a critical element in modern software development. Practitioners have increasingly discovered that close attention to a software system's architecture pays valuable dividends. Without an architecture that is appropriate for the problem being solved, a project will stumble along or, most likely, fail. Even with a superb architecture, if that architecture is not well understood or well communicated the project is unlikely to succeed. *Documenting Software Architectures, Second Edition*, provides the most complete and current guidance, independent of language or notation, on how to capture an architecture in a commonly understandable form. Drawing on their extensive experience, the authors first help you decide what information to document, and then, with guidelines and examples (in various notations,

including UML), show you how to express an architecture so that others can successfully build, use, and maintain a system from it. The book features rules for sound documentation, the goals and strategies of documentation, architectural views and styles, documentation for software interfaces and software behavior, and templates for capturing and organizing information to generate a coherent package. New and improved in this second edition: Coverage of architectural styles such as service-oriented architectures, multi-tier architectures, and data models Guidance for documentation in an Agile development environment Deeper treatment of documentation of rationale, reflecting best industrial practices Improved templates, reflecting years of use and feedback, and more documentation layout options A new, comprehensive example (available online), featuring documentation of a Web-based service-oriented system Reference guides for three important architecture documentation languages: UML, AADL, and SySML

Discover BIM: A better way to build better buildings Building Information Modeling (BIM) offers a novel approach to design, construction, and facility management in which a digital representation of the building product and process is used to facilitate the exchange and interoperability of information in digital format. BIM is beginning to change the way buildings look, the way they function, and the ways in which they are designed and built. The BIM Handbook, Third Edition provides an in-depth understanding of BIM technologies, the business and organizational issues associated with its implementation, and the profound advantages that effective use of BIM can provide to all members of a project team. Updates to this edition include: Information on the ways in which professionals should use BIM to gain maximum value New topics such as collaborative working, national and major construction clients, BIM standards and guides A discussion on how various professional roles have expanded through the widespread use and the new avenues of BIM practices and services A wealth of new case studies that clearly illustrate exactly how BIM is applied in a wide variety of conditions Painting a colorful and thorough picture of the state of the art in building information modeling, the BIM Handbook, Third Edition guides readers to successful implementations, helping them to avoid needless frustration and costs and take full advantage of this paradigm-shifting approach to construct better buildings that consume fewer materials and require less time, labor, and capital resources.

Technical Aspects and Applications

Third International Conference York, UK, October 2-6, 2000 Proceedings
Guide to Applying the UML

Object-oriented Modeling and Design

The Unified Modeling Language Reference Manual

UML for Java Programmers

This new book is the definitive primer for UML, and starts with the foundational concepts of object-orientation in order to provide the proper context for explaining UML.

Most of the articles in this volume are revised versions of papers presented during the

1st GROOM-Workshop on the Unified Modeling Language (UML). GROOM (Grundlagen objektorientierter Modellierung) is a working group of the Gesellschaft für Informatik (GI), the German Society of Computer Science. The workshop took place at the University of Mannheim (Germany) in October 1997; the local organizers were Martin Schader and Axel Korthaus, Department of Information Systems. The scientific program of the workshop included 21 talks, presented in German language on Friday, Oct. 10th, and Saturday, Oct. 11th, 1997. Researchers and practitioners interested in object-oriented software development, analysis and design of software systems, standardization efforts in the field of object technology, and particularly in the main topic of the workshop: "Applications, State of the Art, and Evaluation of the Unified Modeling Language" had the opportunity to discuss recent developments and to establish cooperation in these fields. The workshop owed much to its sponsors and supporters - University of Mannheim - Faculty of Business Administration, University of Mannheim - Sun Microsystems GmbH - Apcon Professional Concepts GmbH. Their generous support is gratefully acknowledged. In the present proceedings volume, papers are presented in three chapters as follows.

The Systems Modeling Language (SysML) extends UML with powerful systems engineering capabilities for modeling a wider spectrum of systems and capturing all aspects of a system's design. SysML Distilled is the first clear, concise guide for everyone who wants to start creating effective SysML models. (Drawing on his pioneering experience at Lockheed Martin and NASA, Lenny Delligatti illuminates SysML's core components and provides practical advice to help you create good models and good designs. Delligatti begins with an easy-to-understand overview of Model-Based Systems Engineering (MBSE) and an explanation of how SysML enables effective system specification, analysis, design, optimization, verification, and validation. Next, he shows how to use all nine types of SysML diagrams, even if you have no previous experience with modeling languages. A case study running through the text demonstrates the use of SysML in modeling a complex, real-world sociotechnical system. Modeled after Martin Fowler's classic UML Distilled, Delligatti's indispensable guide quickly teaches you what you need to know to get started and helps you deepen your knowledge incrementally as the need arises. Like SysML itself, the book is method independent and is designed to support whatever processes, procedures, and tools you already use. Coverage Includes Why SysML was created and the business case for using it Quickly putting SysML to practical use What to know before you start a SysML modeling project Essential concepts that apply to all SysML diagrams SysML diagram elements and relationships Diagramming block definitions, internal structures, use cases, activities, interactions, state machines, constraints, requirements, and packages Using allocations to define mappings among elements across a model SysML notation tables, version changes, and sources for more information

**Watch, listen, and learn as Grady Booch carefully describes key UML concepts with over 200 dynamic animated figures.*Cyber Classroom includes a fully-searchable electronic version of the classic The Unified Modeling Language User Guide, the full*

*text of the UML specification documents, PLUS a UML dictionary with over 600 hyperlinked terms *Also includes a Video Introduction to the UML by Grady Booch, over 300 practice questions to test your knowledge, hyperlinking, full-text searching, and more BONUS: Second CD-ROM includes fully searchable electronic version of The Unified Modeling Language Reference Manual. The worlds most authoritative UML training CD-ROM Now you can learn UML from the original designers: Grady Booch, James Rumbaugh, and Ivar Jacobson This training course includes the UML Multimedia Cyber Classroom CD-ROM, plus Rumbaugh/Jacobson/Boochs masterful The Unified Modeling Language Reference Manual. UML Multimedia Cyber Classroom CD-ROM*Over 300 practice questions to test your knowledge *200+ multimedia UML diagrams animate every key UML concept.*Expert insight straight from the original designers of UML applications *Find it fast CD-ROM includes fully-searchable copy of The Unified Modeling Language User Guide 100% COMPREHENSIVE, 100% AUTHORITATIVE an expert UML modeler, including concepts, syntax, modeling techniques, and more: *Modeling: Fundamental principles and rationale*UML: Overview, conceptual model, architecture & development lifecycle*Classes: Basic & Advanced*Relationships: Basic & Advanced*Common Mechanisms*Diagrams, Class Diagrams, and Object Diagrams*Interfaces, Types, & Roles*Packages & Instances*Interactions & Interaction Diagrams*Use Cases & Use Case Diagrams*Activity Diagrams*Events & Signals*State Machines*Processes & Threads*Time & Space*Statechart Diagrams*Architectural Modeling: Components, Deployment & Collaborations*Patterns & Frameworks*Systems & Models*Hundreds of terms and concepts defined in detail-by the object-oriented modeling experts who created them*Large collection of 2-color UML diagrams, extensively annotated*Expert insight into UML views designed to help you integrate UMLs key constructs into a unified whole*Detailed reference guides to the UML metamodel, notation, and standard extensions Learn modeling hands on-then apply it to a series of increasingly complex, real-world problems Rational Software Corporation and one of the original designers of the UML. Technical requirements: Windows 95/98, Windows NT 4.x, Windows 2000 Internet Explorer (Included) 20 MB disk space 32 MB RAM CD-ROM drive Sound card support*

Managing the Object-oriented Project

Object Modeling with the OCL

Executable UML

BIM Handbook

Object Solutions

Object-Oriented Modeling and Design with UML

Introduces the Unified Modeling Language, explains the fundamentals of modeling elements, structures, and the behaviors of object-oriented software systems, and offers real-world examples. The eagerly awaited Pattern-Oriented Software Architecture (POSA) Volume 4 is about a pattern language for distributed computing. The authors will guide you through the best practices and introduce

you to key areas of building distributed software systems. POSA 4 connects many stand-alone patterns, pattern collections and pattern languages from the existing body of literature found in the POSA series. Such patterns relate to and are useful for distributed computing to a single language. The panel of experts provides you with a consistent and coherent holistic view on the craft of building distributed systems. Includes a foreword by Martin Fowler A must read for practitioners who want practical advice to develop a comprehensive language integrating patterns from key literature. This book constitutes the refereed proceedings of the Third International Conference on the Unified Modeling Language, 2000, held in York, UK in October 2000. The 36 revised full papers presented together with two invited papers and three panel outlines were carefully reviewed and selected from 102 abstracts and 82 papers submitted. The book offers topical sections on use cases, enterprise applications, applications, roles, OCL tools, meta-modeling, behavioral modeling, methodology, actions and constraints, patterns, architecture, and state charts. This book shows us how to use UML and apply it in object-oriented software development. Part 1 of the book guides the reader step-by-step through the development process while part 2 explains the basics of UML in detail.

The Rails Way

A UML Pattern Language

Best of Booch

A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers

The Rational Unified Process Made Easy

An Introduction

The second edition of this text brings the content up to date and in compliance with Rational unified Process 2000. It defines the process, putting it into a proper software development context, reviewing the RUPS history and providing detailed coverage of its structure.

UML has established itself as the industry standard for modeling software systems. Schaum's Outline of UML, Second Edition, provides you with a step-by-step guide to the notation and use of UML, with a focus on the new UML 2.0 software. The book features: Complete explanations of UML modeling technique An exploration of the new UML 2.0 infrastructure Examples and exercises Two extended cases studies New review questions And more

bull; Reflects all of the changes that were integrated into RUP v2003—the latest version of the very popular product bull; Learn the key concepts, fundamentals of structure, integral content, and motivation behind the RUP bull; Covers all phases of the software development lifecycle -from concept, to delivery, to revision

This textbook mainly addresses beginners and readers with a basic knowledge of object-oriented programming languages like Java or C#, but with little or no modeling or software engineering experience -

thus reflecting the majority of students in introductory courses at universities. Using UML, it introduces basic modeling concepts in a highly precise manner, while refraining from the interpretation of rare special cases. After a brief explanation of why modeling is an indispensable part of software development, the authors introduce the individual diagram types of UML (the class and object diagram, the sequence diagram, the state machine diagram, the activity diagram, and the use case diagram), as well as their interrelationships, in a step-by-step manner. The topics covered include not only the syntax and the semantics of the individual language elements, but also pragmatic aspects, i.e., how to use them wisely at various stages in the software development process. To this end, the work is complemented with examples that were carefully selected for their educational and illustrative value. Overall, the book provides a solid foundation and deeper understanding of the most important object-oriented modeling concepts and their application in software development. An additional website offers a complete set of slides to aid in teaching the contents of the book, exercises and further e-learning material.

Applying Use Cases

A Practical Guide

Pattern-Oriented Software Architecture, A Pattern Language for Distributed Computing

Documenting Software Architectures

UML 2.0 in a Nutshell

Guide to the Unified Process featuring UML, Java and Design Patterns

This volume, dedicated to Bernd Silbermann on his sixtieth birthday, collects research articles on Toeplitz matrices and singular integral equations written by leading area experts. The subjects of the contributions include Banach algebraic methods, Toeplitz determinants and random matrix theory, Fredholm theory and numerical analysis for singular integral equations, and efficient algorithms for linear systems with structured matrices, and reflect Bernd Silbermann's broad spectrum of research interests. The volume also contains a biographical essay and a list of publications. The book is addressed to a wide audience in the mathematical and engineering sciences. The articles are carefully written and are accessible to motivated readers with basic knowledge in functional analysis and operator theory.

With its clear introduction to the Unified Modeling Language (UML) 2.0, this tutorial offers a solid understanding of each topic, covering foundational concepts of object-orientation and an introduction to each of the UML diagram types.

Social scientists, whether earnest graduate students or tenured faculty members, clearly know the rules that govern good writing. But for some reason they choose to ignore those guidelines and churn out turgid, pompous, and obscure prose.

Distinguished sociologist Howard S. Becker, true to his calling, looks for an explanation for this bizarre behavior not in the psyches of his colleagues but in the structure of his profession. In this highly personal and inspirational volume he considers academic writing as a social activity. Both the means and the reasons for writing a thesis or article or book are socially structured by the organization of graduate study, the requirements for publication, and the conditions for promotion, and the pressures arising from these situations create the writing style so often lampooned and lamented. Drawing on his thirty-five years' experience as a

researcher, writer, and teacher, Becker exposes the foibles of the academic profession to the light of sociological analysis and gentle humor. He also offers eminently useful suggestions for ways to make social scientists better and more productive writers. Among the topics discussed are how to overcome the paralyzing fears of chaos and ridicule that lead to writer's block; how to rewrite and revise, again and again; how to adopt a persona compatible with lucid prose; how to deal with that academic bugaboo, "the literature." There is also a chapter by Pamela Richards on the personal and professional risks involved in scholarly writing. In recounting his own trials and errors Becker offers his readers not a model to be slavishly imitated but an example to inspire. Throughout, his focus is on the elusive work habits that contribute to good writing, not the more easily learned rules of grammar and punctuation. Although his examples are drawn from sociological literature, his conclusions apply to all fields of social science, and indeed to all areas of scholarly endeavor. The message is clear: you don't have to write like a social scientist to be one.

The authors explain the underlying software development principles behind theRUP, and guide readers in its application in their organization.

The Unified Modeling Language

Software Engineering with Ada

The Unified Modeling Language User Guide

UML Distilled

Object-oriented Analysis and Design in Practice

The Complete UML Training Course

Object Solutions is a direct outgrowth of Grady Booch's experience with object-oriented project in development around the world. This book focuses on the development process and is the perfect resource for developers and managers who want to implement object technologies for the first time or refine their existing object-oriented development practice. The book is divided into two major sections. The first four chapters describe in detail the process of object-oriented development in terms of inputs, outputs, products, activities, and milestones. The remaining ten chapters provide practical advice on key issues including management, planning, reuse, and quality assurance. Drawing upon his knowledge of strategies used in both successful and unsuccessful projects, Grady Booch offers pragmatic advice for applying object-technologies and controlling projects effectively.

John Hunt's book guides you through the use of the UML and the Unified Process and their application to Java systems. Key topics focus explicitly on applying the notation and the method to Java. The book is clearly structured and written, making it ideal for practitioners. This second edition is considerably revised and extended and includes examples taken from the latest version of Rational Rose and Together. Considers how Agile Modelling fits with the Unified Process, and presents Design Patterns Self contained - covers both the Unified Process and UML in one book Includes real-world case studies Written by an

experienced author and industry expert Ideal for students on Software Engineering courses

The expert guide to building Ruby on Rails applications Ruby on Rails strips complexity from the development process, enabling professional developers to focus on what matters most: delivering business value. Now, for the first time, there's a comprehensive, authoritative guide to building production-quality software with Rails. Pioneering Rails developer Obie Fernandez and a team of experts illuminate the entire Rails API, along with the Ruby idioms, design approaches, libraries, and plug-ins that make Rails so valuable. Drawing on their unsurpassed experience, they address the real challenges development teams face, showing how to use Rails' tools and best practices to maximize productivity and build polished applications users will enjoy. Using detailed code examples, Obie systematically covers Rails' key capabilities and subsystems. He presents advanced programming techniques, introduces open source libraries that facilitate easy Rails adoption, and offers important insights into testing and production deployment. Dive deep into the Rails codebase together, discovering why Rails behaves as it does— and how to make it behave the way you want it to. This book will help you Increase your productivity as a web developer Realize the overall joy of programming with Ruby on Rails Learn what's new in Rails 2.0 Drive design and protect long-term maintainability with TestUnit and RSpec Understand and manage complex program flow in Rails controllers Leverage Rails' support for designing REST-compliant APIs Master sophisticated Rails routing concepts and techniques Examine and troubleshoot Rails routing Make the most of ActiveRecord object-relational mapping Utilize Ajax within your Rails applications Incorporate logins and authentication into your application Extend Rails with the best third-party plug-ins and write your own Integrate email services into your applications with ActionMailer Choose the right Rails production configurations Streamline deployment with Capistrano Designed for software professionals who are concerned about the success of their object-oriented projects, this volume covers all aspects of the Booch method and how a complete method must address a model's notation and semantics as well as a process for creating that model

Designing Strategies for Object Technology

An Introduction to Object-Oriented Modeling

The Unified Modelling Language Reference Manual

Visual Modeling with Rational Rose 2002 and UML

Learning PHP Design Patterns

A Foundation for Model-driven Architecture

This text applies object-oriented techniques to the entire software development cycle.

A tool-independent and process-independent roadmap for successfully applying the Unified Modeling Language (UML). UML is a modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system-intensive process. It was originally conceived by Rational Software Corporation and three of the most prominent methodologists in the information systems and technology industry: Grady Booch, James Rumbaugh, and Ivar Jacobson. This text contains numerous practical real-world examples to help novice and expert users understand the whole language (holistically and cohesively), including rules of usage and principles of composition, style guidelines, and a roadmap for successfully applying the UML.

The first of two UML works written by the creators of UML, this book introduces the core 80 percent of UML, approaching it in a layered fashion and providing numerous examples of its application.

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. 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.

A Brief Guide to the Standard Object Modeling Language

Views and Beyond

Covers UML 2.0

The Unified Software Development Process

Learning UML 2.0

A Practitioner's Guide to the RUP

This comprehensive guide has been fully revised to cover UML 2.0, today's standard method for modelling software systems. Filled with concise information, it's been crafted to help IT professionals read, create, and understand system artefacts expressed using UML. Includes an example-rich tutorial for those who need familiarizing with the system.

More than 300,000 developers have benefited from past editions of UML Distilled . This third edition is the best resource for quick, no-nonsense insights into understanding and using UML 2.0 and prior versions of the UML. Some readers will want to quickly get up to speed with the UML 2.0 and learn the essentials of the UML. Others will use this book as a handy, quick reference to the most common parts of the UML. The author delivers on both of these promises in a short, concise, and focused presentation. This book describes all the major UML diagram types, what they're used for, and the basic notation involved in creating and deciphering them. These diagrams include class, sequence, object, package, deployment, use case, state machine, activity, communication, composite structure, component, interaction overview, and timing diagrams. The examples are clear and the explanations cut to the fundamental design logic. Includes a quick reference to the most useful parts of the UML notation and a useful summary of diagram types that were added to the UML 2.0. If you are like most developers, you don't have time to keep up with all the new innovations in software

engineering. This new edition of Fowler's classic work gets you acquainted with some of the best thinking about efficient object-oriented software design using the UML--in a convenient format that will be essential to anyone who designs software professionally.

Provides complete coverage of the Ada language and Ada programming in general by recognized authorities in Ada software engineering. Demonstrates the power and performance of Ada in the management of large-scale object-oriented systems, and shows how to use Ada features such as generics, packages, and tasking.

"If you are a serious user of UML, there is no other book quite like this one. I have been involved with the UML specification process for some time, but I still found myself learning things while reading through this book--especially on the changes and new capabilities that have come with UML." -Ed Seidewitz, Chief Architect, IntelliData Technologies Corporation

The latest version of the Unified Modeling Language-UML 2.0-has increased its capabilities as the standard notation for modeling software-intensive systems. Like most standards documents, however, the official UML specification is difficult to read and navigate. In addition, UML 2.0 is far more complex than previous versions, making a thorough reference book more essential than ever. In this significantly updated and expanded edition of the definitive reference to the standard, James Rumbaugh, Ivar Jacobson, and Grady Booch--the UML's creators--clearly and completely describe UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Whether you are capturing requirements, developing software architectures, designing implementations, or trying to understand existing systems, this is the book for you. Highlights include:

- Alphabetical dictionary of articles covering every UML concept
- Integrated summary of UML concepts by diagram type
- Two-color diagrams with extensive annotations in blue
- Thorough coverage of both semantics and notation, separated in each article for easy reference
- Further explanations of concepts whose meaning or purpose is obscure in the original specifications
- Discussion sections offering usage advice and additional insight into tricky concepts
- Notation summary, with references to individual articles
- An enhanced online index available on the book's web site allowing readers to quickly and easily search the entire text for specific topics

The result is an indispensable resource for anyone who needs to understand the inner workings of the industry standard modeling language.

Learning UML

SysML Distilled

UML 2000 - The Unified Modeling Language: Advancing the Standard

Developing Software with UML

UML Explained

The Unified Modeling Language has become the industry standard for the expression of software designs. The Java programming language continues to grow in popularity as the language of choice for the serious application developer. Using UML and Java together would appear to be a natural marriage, one that can produce considerable benefit. However, there are nuances that the seasoned developer needs to keep in mind when using UML and Java together. Software expert Robert Martin presents a concise guide, with numerous examples, that will help the programmer leverage the power of both development concepts. The author ignores features of UML that do not apply to java programmers, saving the reader time and effort. He provides direct guidance and points the reader to real-world usage scenarios. The overall practical approach of this book brings key information related to Java to the many presentations. The result is an highly practical guide to using the UML with Java.