

Use Case Driven Object Modeling With UML Theory And Practice

This compact book helps application developers bridge the gap between the theory of the newly created Unified Software Development Process and the practical realities necessary to design and build a software system. The authors present the key ingredients of the Unified Process and demonstrate how the process was conceived to work with UML, emphasizing the application of Use Cases as a primary design tool. The book incorporates a wealth of practical experience showcased by four case studies -- a hospital information system, a video on demand system, a portfolio management system, and a vehicle navigation (IVHS) system.

The first UML book to cover Rational Rose 2000, this brand-new edition reviews the three key interrelated components of state-of-the-art software system design: the Rational Unified process, the Unified Modeling Language, and Rational Rose 2000. Then, through a simplified case study, it walks developers through a real-world business system. Includes screen shots demonstrating UML at work in the Rational Rose 2000 modeling tool.

This book covers all you need to know to model and design software applications from use cases to software architectures in UML and shows how to apply the COMET UML-based modeling and design method to real-world problems. The author describes architectural patterns for various architectures, such as broker, discovery, and transaction patterns for service-oriented architectures, and addresses software quality attributes including maintainability, modifiability, testability, traceability, scalability, reusability, performance, availability, and security.

Online Library Use Case Driven Object Modeling With UML Theory And Practice

Complete case studies illustrate design issues for different software architectures: a banking system for client/server architecture, an online shopping system for service-oriented architecture, an emergency monitoring system for component-based software architecture, and an automated guided vehicle for real-time software architecture. Organized as an introduction followed by several short, self-contained chapters, the book is perfect for senior undergraduate or graduate courses in software engineering and design, and for experienced software engineers wanting a quick reference at each stage of the analysis, design, and development of large-scale software systems.

This book offers a unique insight into a revolution in software development that allows model specifications to be fully and efficiently translated into code. Using the most widely adopted, industry standard, software modelling language, UML, the reader will learn how to build robust specifications based on OMG's Model Driven Architecture (MDA). From there, the authors describe the steps needed to translate the Executable UML (xUML) models to any platform-specific implementation. The benefits of this approach go well beyond simply reducing or eliminating the coding stage - it also ensures platform independence, avoids obsolescence (programming languages may change, the model doesn't) and allows full verification of the models by executing them in a test and debug xUML environment. This is an excellent reference for anyone embarking on what is surely the future of software development for medium and large scale projects.

Use Case Driven Object Modeling with UML Theory and Practice

Online Library Use Case Driven Object Modeling With UML Theory And Practice

Applying Use Cases

Second Edition

Executable UML

An Introduction to Metaphysics

The Written Language Bias in Linguistics

Linguists routinely emphasise the primacy of speech over writing. Yet, most linguists have analysed spoken language, as well as language in general, applying theories and methods that are best suited for written language. Accordingly, there is an extensive 'written language bias' in traditional and present day linguistics and other language sciences. In this book, this point is argued with rich and convincing evidence from virtually all fields of linguistics.

This book discusses how model-based approaches can improve the daily practice of software professionals. This is known as Model-Driven Software Engineering (MDSE) or, simply, Model-Driven Engineering (MDE). MDSE practices have proved to increase efficiency and effectiveness in software development, as demonstrated by various quantitative and qualitative studies. MDSE adoption in the software industry is foreseen to grow exponentially in the near future, e.g., due to the convergence of software development and business analysis. The aim of this book is to provide you with an agile and flexible tool to introduce you to the MDSE world, thus allowing you to quickly understand its basic principles and techniques and to

Online Library Use Case Driven Object Modeling With UML Theory And Practice

choose the right set of MDSE instruments for your needs so that you can start to benefit from MDSE right away. The book is organized into two main parts. The first part discusses the foundations of MDSE in terms of basic concepts (i.e., models and transformations), driving principles, application scenarios, and current standards, like the well-known MDA initiative proposed by OMG (Object Management Group) as well as the practices on how to integrate MDSE in existing development processes. The second part deals with the technical aspects of MDSE, spanning from the basics on when and how to build a domain-specific modeling language, to the description of Model-to-Text and Model-to-Model transformations, and the tools that support the management of MDSE projects. The second edition of the book features: a set of completely new topics, including: full example of the creation of a new modeling language (IFML), discussion of modeling issues and approaches in specific domains, like business process modeling, user interaction modeling, and enterprise architecture complete revision of examples, figures, and text, for improving readability, understandability, and coherence better formulation of definitions, dependencies between concepts and ideas addition of a complete index of book content In addition to the contents of the book, more resources are provided on the book's website <http://www.mdse-book.com>, including the examples presented in the book.

The acclaimed beginner's book on object technology now presents UML 2.0, Agile

Online Library Use Case Driven Object Modeling With UML Theory And Practice

Modeling, and the latest in object development techniques.

The 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.

Domain-driven Design

Software Modeling and Design

Using the Unified Modeling Language

Using CRC Cards

An Annotated E-commerce Example

Model Driven Architecture with Executable UML

Conallen introduces architects and designers and client/server systems to issues and techniques of developing software for the Web. He expects readers to be familiar with object-oriented principles and concepts, particularly with UML (unified modeling language), and at least one Web application architecture or environment. The second edition incorporates both technical developments and his experience since 1999. He does not provide a bibliography. Annotation copyrighted by Book News, Inc., Portland, OR

The groundbreaking book Design Driven Testing brings sanity back to the software development process by flipping around the concept of Test Driven Development

Online Library Use Case Driven Object Modeling With UML Theory And Practice

(TDD)—restoring the concept of using testing to verify a design instead of pretending that unit tests are a replacement for design. Anyone who feels that TDD is “Too Damn Difficult” will appreciate this book. Design Driven Testing shows that, by combining a forward-thinking development process with cutting-edge automation, testing can be a finely targeted, business-driven, rewarding effort. In other words, you’ll learn how to test smarter, not harder. Applies a feedback-driven approach to each stage of the project lifecycle. Illustrates a lightweight and effective approach using a core subset of UML. Follows a real-life example project using Java and Flex/ActionScript. Presents bonus chapters for advanced DDTers covering unit-test antipatterns (and their opposite, “test-conscious” design patterns), and showing how to create your own test transformation templates in Enterprise Architect.

"This is the fourth report on mothers and babies in NSW to combine the annual reports of the NSW Midwives Data Collection (MDC), the Neonatal Intensive Care Units' Data Collection and the NSW Birth Defects Register."--Page 9.

Introduces CRC (Class, Responsibility, Collaborator) cards and describes how they can be used in interactive sessions to develop an object-oriented model of an application.

Object-oriented Design and Architecture

The Object Primer

A Brief Guide to the Standard Object Modeling Language

The Economic and Fiscal Consequences of Immigration

Online Library Use Case Driven Object Modeling With UML Theory And Practice

Building Web Applications with UML A Practical Guide

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

ICONIX Process has a long track record of helping companies avoid analysis paralysis on a multitude of projects, and is best suited for developing Web and GUI-based systems. This resource contains a treasure-trove of tailored roadmaps, proven on demanding real-life projects.

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

Online Library Use Case Driven Object Modeling With UML Theory And Practice

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.

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

Online Library Use Case Driven Object Modeling With UML Theory And Practice

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.

The Application Developer's Guide to Object-Oriented and the UML Design Driven Testing

UML, Use Cases, Patterns, and Software Architectures

Fast Track UML 2.0

Extreme Programming Refactored

Diagramming and process are important topics in today's software development world, as the UML diagramming language has come to be almost universally accepted. Yet process is necessary; by themselves, diagrams are of little use. Use Case Driven Object Modeling with UML - Theory and Practice combines the notation of UML with a lightweight but effective process - the ICONIX process - for designing and developing software systems. ICONIX has developed a growing following over the years. Sitting between the free-for-all of Extreme Programming and overly rigid processes such as RUP, ICONIX offers just enough structure to

Online Library Use Case Driven Object Modeling With UML Theory And Practice

be successful.

Describes ways to incorporate domain modeling into software development.

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.

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.

Use Case Modeling

Its Nature, Origins and Transformations

A Foundation for Model-driven Architecture

Aspect-Oriented Requirements Engineering

People, Process, and Pragmatism

A Practical Guide to Testing Object-oriented Software

Use Case Driven Object Modeling with UML Theory and Practice shows how to drive an object-oriented software design from use case all the way through coding and testing, based on the minimalist, UML-based ICONIX process. In addition to a comprehensive explanation of the foundations of the approach, the book makes extensive use of examples and provides

Online Library Use Case Driven Object Modeling With UML Theory And Practice

exercises at the back of each chapter.· Introduction to ICONIX Process· Domain Modeling· Use Case Modeling· Requirements Review· Robustness Analysis· Preliminary Design Review· Technical Architecture· Sequence Diagrams· Critical Design Review· Implementation: Getting from Detailed Design to Code· Code Review and Model Update· Design-Driven Testing· Addressing Requirements

Broadly-scoped requirements such as security, privacy, and response time are a major source of complexity in modern software systems. This is due to their tangled inter-relationships with and effects on other requirements. Aspect-Oriented Requirements Engineering (AORE) aims to facilitate modularisation of such broadly-scoped requirements, so that software developers are able to reason about them in isolation - one at a time. AORE also captures these inter-relationships and effects in well-defined composition specifications, and, in so doing exposes the causes for potential conflicts, trade-offs, and roots for the key early architectural decisions. Over the last decade, significant work has been carried out in the field of AORE. With this book the editors aim to provide a consolidated overview of these efforts and results. The individual contributions discuss how aspects can be identified, represented, composed and reasoned about, as well as how they are used in specific domains and in industry. Thus, the book does not present one particular AORE approach, but conveys a broad understanding of the aspect-

Online Library Use Case Driven Object Modeling With UML Theory And Practice

oriented perspective on requirements engineering. The chapters are organized into five sections: concern identification in requirements, concern modelling and composition, domain-specific use of AORE, aspect interactions, and AORE in industry. This book provides readers with the most comprehensive coverage of AORE and the capabilities it offers to those grappling with the complexity arising from broadly-scoped requirements - a phenomenon that is, without doubt, universal across software systems. Software engineers and related professionals in industry, as well as advanced undergraduate and post-graduate students and researchers, will benefit from these comprehensive descriptions and the industrial case studies.

*** Examples are easy to understand; diagrams aren't overly busy. * Written in user-friendly style author is known for. * Condensed, distilled presentation of the UML Superstructure document will get you up to speed with UML 2.0.**

For courses in Software Engineering, Software Development, or Object-Oriented Design and Analysis at the Junior/Senior or Graduate level. This text can also be utilized in short technical courses or in short, intensive management courses. Shows students how to use both the principles of software engineering and the practices of various object-oriented tools, processes, and products. Using a step-by-step case study to illustrate the concepts and topics in each chapter, Bruegge and Dutoit emphasize

Online Library Use Case Driven Object Modeling With UML Theory And Practice

learning object-oriented software engineer through practical experience: students can apply the techniques learned in class by implementing a real-world software project. The third edition addresses new trends, in particular agile project management (Chapter 14 Project Management) and agile methodologies (Chapter 16 Methodologies).

The Unified Modeling Language User Guide

Theory and Practice

A Practical Approach

The Art of Objects

Visual Modeling with Rational Rose 2000 and UML

The Case Against XP

Discusses how to define and organize use cases that model the user requirements of a software application. The approach focuses on identifying all the parties who will be using the system, then writing detailed use case descriptions and structuring the use case model. An ATM example runs throughout the book. The authors work at Rational Software.

Annotation copyrighted by Book News, Inc., Portland, OR

The basic principles that Bergson articulates, especially his way of thinking about reality as a dynamic process and his view of human beings as creative and evolving, should be helpful to anyone who seeks to go beyond

simply dealing with the practical demands of daily life and consider the nature of things. Of special importance is Bergson's claim that it is both possible and necessary to know from the inside rather than confining our attention to external perspectives and points of view. Intuition is able to get beyond what is relative and place us inside reality. This essay is, as the title says, an introduction. But if we think there is more to a human being — and even to nature itself — than material structures alone, perhaps the time has come to take a fresh look at Bergson's essay. In "An Introduction to Metaphysics," Bergson traces the demise of metaphysics to the failure of both scientific materialism and dogmatism and to the immense success of a kind of pragmatism that promised liberation from the fruitless battles among various schools of philosophy. He also rejects relativism and criticizes the vacuum that is created when philosophers refuse to inquire about the nature of reality. To avoid metaphysics easily leads to a worldview shaped by unexamined ideas and hidden presuppositions. Henri Bergson was born in the year that Darwin published the Origin of the Species. He could not have imagined the philosophical impact of evolutionary theory, which is now so great that Bergson's philosophy, which emphasizes "creative evolution," is experiencing a significant

revival. The basic principles that Bergson articulates, especially his way of thinking about reality as a dynamic process and his view of human beings as creative and evolving, should be helpful to anyone who seeks to go beyond simply dealing with the practical demands of daily life and consider the nature of things.

For nearly ten years, the Unified Modeling Language (UML) has been the industry standard for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. As the de facto standard modeling language, the UML facilitates communication and reduces confusion among project stakeholders. The recent standardization of UML 2.0 has further extended the language's scope and viability. Its inherent expressiveness allows users to model everything from enterprise information systems and distributed Web-based applications to real-time embedded systems. In this eagerly anticipated revision of the best-selling and definitive guide to the use of the UML, the creators of the language provide a tutorial to its core aspects in a two-color format designed to facilitate learning. Starting with an overview of the UML, the book explains the language gradually by introducing a few concepts and notations in each chapter. It also illustrates the application of the UML to complex

Online Library Use Case Driven Object Modeling With UML Theory And Practice

modeling problems across a variety of application domains. The in-depth coverage and example-driven approach that made the first edition of The Unified Modeling Language User Guide an indispensable resource remain unchanged. However, content has been thoroughly updated to reflect changes to notation and usage required by UML 2.0. Highlights include: A new chapter on components and internal structure, including significant new capabilities for building encapsulated designs New details and updated coverage of provided and required interfaces, collaborations, and UML profiles Additions and changes to discussions of sequence diagrams, activity diagrams, and more Coverage of many other changes introduced by the UML 2.0 specification With this essential guide, you will quickly get up to speed on the latest features of the industry standard modeling language and be able to apply them to your next software project. Venturing beyond C++ programming, this text shows how to engineer software products using object-oriented principles. It covers gathering requirements, specifying objects, object verification, defining relations between objects, translating object design into code, object testing, and software maintenance.

Object Oriented Systems Development

Online Library Use Case Driven Object Modeling With UML Theory And Practice

Step-By-Step Guidance for Soa, Embedded, and Algorithm-Intensive Systems

Model-Driven Software Engineering in Practice

Agile Model-Driven Development with UML 2.0

APPLYING UML & PATTERNS 3RD EDITION

Object-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International Edition

Based on Objectory which is the first commercially available comprehensive object-oriented process for developing large scale industrial systems.

****Describes an agile process that works on large projects *Ideal for hurried developers who want to develop software in teams *Incorporates real-life C#/.NET web project; can compare this with cases in book***

Scott Ambler, author of Building Object Applications that Work, Process Patterns, and More Process Patterns, has revised his acclaimed first book, The Object Primer. Long prized in its original edition by both students and professionals as the best introduction to object-oriented technology, now this book is completely up-to-date with new material in every chapter. There are also new chapters on good OO programming techniques and OO software testing. All modeling notation has been rewritten in UML notation. Review questions at the end of each chapter allow readers to test their newly acquired knowledge. In

Online Library Use Case Driven Object Modeling With UML Theory And Practice

addition, the author takes time to reflect on the lessons learned over the past few years by discussing the proven benefits and drawbacks of the technology. This is the perfect book for any software development professional or student seeking an introduction to the concepts and terminology of object technology.

The Economic and Fiscal Consequences of Immigration finds that the long-term impact of immigration on the wages and employment of native-born workers overall is very small, and that any negative impacts are most likely to be found for prior immigrants or native-born high school dropouts. First-generation immigrants are more costly to governments than are the native-born, but the second generation are among the strongest fiscal and economic contributors in the U.S. This report concludes that immigration has an overall positive impact on long-run economic growth in the U.S. More than 40 million people living in the United States were born in other countries, and almost an equal number have at least one foreign-born parent. Together, the first generation (foreign-born) and second generation (children of the foreign-born) comprise almost one in four Americans. It comes as little surprise, then, that many U.S. residents view immigration as a major policy issue facing the nation. Not only does immigration affect the environment in which everyone lives, learns, and works, but it also interacts with nearly every policy area of concern, from jobs and the economy, education, and health care, to federal, state, and local government budgets. The

Online Library Use Case Driven Object Modeling With UML Theory And Practice

changing patterns of immigration and the evolving consequences for American society, institutions, and the economy continue to fuel public policy debate that plays out at the national, state, and local levels. The Economic and Fiscal Consequences of Immigration assesses the impact of dynamic immigration processes on economic and fiscal outcomes for the United States, a major destination of world population movements. This report will be a fundamental resource for policy makers and law makers at the federal, state, and local levels but extends to the general public, nongovernmental organizations, the business community, educational institutions, and the research community.

A Use Case Driven Approach

Use Case Driven Object Modeling With Uml: Theory And Practice

UML Explained

UML Distilled

Tackling Complexity in the Heart of Software

Test Smarter, Not Harder

From the beginning of software time, people have wondered why it isn't possible to accelerate software projects by simply adding staff. This is sometimes known as the "nine women can't make a baby in one month" problem. The most famous treatise declaring this to be impossible is Fred Brooks' 1975 book The Mythical Man-Month, in which he declares that

Online Library Use Case Driven Object Modeling With UML Theory And Practice

“adding more programmers to a late software project makes it later,” and indeed this has proven largely true over the decades. Aided by a domain-driven code generator that quickly creates database and API code, Parallel Agile (PA) achieves significant schedule compression using parallelism: as many developers as necessary can independently and concurrently develop the scenarios from initial prototype through production code. Projects can scale by elastic staffing, rather than by stretching schedules for larger development efforts. Schedule compression with a large team of developers working in parallel is analogous to hardware acceleration of compute problems using parallel CPUs. PA has some similarities with and differences from other Agile approaches. Like most Agile methods, PA “gets to code early” and uses feedback from executable software to drive requirements and design. PA uses technical prototyping as a risk-mitigation strategy, to help sanity-check requirements for feasibility, and to evaluate different technical architectures and technologies. Unlike many Agile methods, PA does not support “design by refactoring,” and it doesn't drive designs from unit tests. Instead, PA uses a minimalist UML-based design approach (Agile/ICONIX) that starts out with a domain model to facilitate communication across the development team, and partitions the system along use case boundaries,

Online Library Use Case Driven Object Modeling With UML Theory And Practice

which enables parallel development. Parallel Agile is fully compatible with the Incremental Commitment Spiral Model (ICSM), which involves concurrent effort of a systems engineering team, a development team, and a test team working alongside the developers. The authors have been researching and refining the PA process for several years on multiple test projects that have involved over 200 developers. The book's example project details the design of one of these test projects, a crowdsourced traffic safety system.

Use Case Driven Object Modeling with UML Theory and Practice Theory and Practice Apress

The Art of Objects offers an extensive overview of the long-standing principles of object technology, along with leading-edge developments in the field. It will give you a greater understanding of design patterns and the know-how to use them to find effective solutions to a wide range of design challenges. And because the book maintains an approach independent of specific programming languages, the concepts and techniques presented here can be applied to any object-oriented development environment. Using the Unified Modeling Language (UML), The Art of Objects examines numerous static and dynamic practical object design patterns, illustrated by real-life case studies that demonstrate how to put the patterns to work. You will also find

Online Library Use Case Driven Object Modeling With UML Theory And Practice

*discussion of basic concepts of database management and persistent objects, and an introduction to advanced topics in object modeling and interface design patterns. Moving beyond the design level, the book also covers important concepts in object-oriented architecture. Specific topics include: *Object creation and destruction, associations and links, aggregation, inheritance, and other object design fundamentals *UML notation basics for static and dyna*

Stephens and Rosenberg examine XP in the context of existing methodologies and processes such as RUP, ICONIX, Spiral, RAD, DSDM, etc - and show how XP goals can be achieved using these existing processes.

Use Case Driven Object Modeling with UML

Object-oriented Software Engineering

Agile Development with ICONIX Process

An Annotated E-commerce Example

Iconix Process Roadmaps

The Unified Software Development Process

David A. Sykes is a member of Wofford College's faculty.

Object - Oriented Modeling And Design With Uml, 2/E

An Informal Approach to Object-Oriented Development

Online Library Use Case Driven Object Modeling With UML Theory And Practice

Parallel Agile – faster delivery, fewer defects, lower cost
Applying Use Case Driven Object Modeling with UML