

Uml Distilled Applying The Standard Object Modelling Language Object Technology Series

Software architecture is a primary factor in the creation and evolution of virtually all products involving software. It is a topic of major interest in the research community where numerous formalisms, processes, and technologies are under development. Architecture is also of major interest in industry because it is recognized as a significant leverage point for manipulating such basic development factors as cost, quality, and interval. Its importance is attested to by the fact that there are several international workshop series as well as major conference sessions devoted to it. The First Working IFIP Conference on Software Architecture (WICSA1) provided a focused and dedicated forum for the international software architecture community to unify and coordinate its effort to advance the state of practice and research. WICSA 1 was organized to facilitate information exchange between practising software architects and software architecture researchers. The conference was held in San Antonio, Texas, USA, from February 22nd to February 24th, 1999; it was the initiating event for the new IFIP TC-2 Working Group on Software Architecture. This proceedings document contains the papers accepted for the conference. The papers in this volume comprise both experience reports and technical papers. The proceedings reflect the structure of the conference and are divided into six sections corresponding to the working groups established for the conference.

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 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.

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.

UML Distilled A Brief Guide to the Standard Object Modeling Language

A UML Pattern Language

Practical E-business Applications

The Unified Modeling Language Reference Manual

Applied Software Architecture

9th International Conference, RTCSA 2003, Tainan, Taiwan, February 18-20, 2003. Revised Papers

8th Ada-Europe International Conference on Reliable Software Technologies, Toulouse, France, June 16-20, 2003, Proceedings

Advances in Computing and Information Technology

The objective of the workshops associated with the ER2000 19th International Conference on Conceptual Modeling was to give participants the opportunity to present and discuss emerging, hot topics, thus adding new perspectives to conceptual modeling. This attracts communities which have begun to or which have already recognized the importance of conceptual modeling for solving their problems. To meet this objective, we selected the following two topics: { Conceptual Modeling Approaches for E-Business (eCOMO2000) aimed at studying the application of conceptual modeling techniques specially to e-business. { The World Wide Web and Conceptual Modeling (WCM2000) which analyzes how conceptual modeling can help address the challenges of Web development, management, and use. eCOMO2000 is the first international workshop on Conceptual Modeling - approaches for E-Business. It was intended to work out and to discuss the actual state of research on conceptual modeling aspects and methods within the realm of the network economy, which is driven by both traditionally organized enterprises and dynamic networks. Following the philosophy of the ER workshops, the selection of eCOMO contributions was done very carefully and restrictively (six accepted papers out of thirteen submissions) in order to guarantee an excellent workshop program. We are deeply indebted to the authors and to the members of the program committee, whose work resulted in this outstanding program.

Up until a few years ago there were over 150 different modelling languages available to software developers. This vast array of choice however, only served to severely hinder effective communication. Therefore, to combat this, every methodologist and many companies agreed to speak the same language, hence the birth of the unified modelling language (UML). The UML offers a means to communicate complex information in a simple way using visual modelling; i.e. drawing diagrams to create a model of a system. This fully revised edition, based on a training course given by the author, coincides with the release of UML version 2 by the standard body, the Object Management Group, and covers the

significant changes that have occurred since its release. It also includes material on life cycle management, examining the way the UML can be used to control and manage projects and the UML systems engineering profile.

The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not easily implemented. Common failures in enterprise applications often occur because their developers do not understand the architectural lessons that experienced object developers have learned. Patterns of Enterprise Application Architecture is written in direct response to the stiff challenges that face enterprise application developers. The author, noted object-oriented designer Martin Fowler, noticed that despite changes in technology--from Smalltalk to CORBA to Java to .NET--the same basic design ideas can be adapted and applied to solve common problems. With the help of an expert group of contributors, Martin distills over forty recurring solutions into patterns. The result is an indispensable handbook of solutions that are applicable to any enterprise application platform. This book is actually two books in one. The first section is a short tutorial on developing enterprise applications, which you can read from start to finish to understand the scope of the book's lessons. The next section, the bulk of the book, is a detailed reference to the patterns themselves. Each pattern provides usage and implementation information, as well as detailed code examples in Java or C#. The entire book is also richly illustrated with UML diagrams to further explain the concepts. Armed with this book, you will have the knowledge necessary to make important architectural decisions about building an enterprise application and the proven patterns for use when building them. The topics covered include

- Dividing an enterprise application into layers
- The major approaches to organizing business logic
- An in-depth treatment of mapping between objects and relational databases
- Using Model-View-Controller to organize a Web presentation
- Handling concurrency for data that spans multiple transactions
- Designing distributed object interfaces

Gathering customer requirements is a key activity for developing software that meets the customer's needs. A concise and practical overview of everything a requirement's analyst needs to know about establishing customer requirements, this first-of-its-kind book is the perfect desk guide for systems or software development work. The book enables professionals to identify the real customer requirements for their projects and control changes and additions to these requirements. This unique resource helps practitioners understand the importance of requirements, leverage effective requirements practices, and better utilize resources. The book also explains how to strengthen interpersonal relationships and communications which are major contributors to project effectiveness. Moreover, analysts find clear examples and checklists to help them implement best practices.

Third Pacific Rim International Workshop on Multi-Agents, PRIMA 2000 Melbourne, Australia, August 28-29, 2000 Proceedings

The Elements of UML(TM) 2.0 Style

APPLYING UML & PATTERNS 3RD EDITION

Applying the Standard Object Modeling Language

Software Architecture

Refactoring

The Unified Modeling Language User Guide

Second Edition of the UML video course based on the book Applying UML and Patterns. This VTC will focus on object-oriented analysis and design, not just drawing UML.

This volume constitutes the refereed proceedings of the Third Pacific Rim International Workshop on Multi-Agents, PRIMA 2000, held in Melbourne, Australia in August 2000 in conjunction with F. The 13 revised full papers presented were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on coordination, negotiation, and learning; architecture specification, and allocation; and applications in control and process management.

ABOUT THE TECHNOLOGY What it is: UML (Unified Modeling Language) is a graphical modeling language used to specify, visualize, construct, and document applications and software systems, which are implemented with components and object-oriented programming languages, such as Java, C++, and Visual Basic. UML incorporates the object-oriented community's consensus on core modeling concepts and provides a standard way for developers to communicate the details of system design and development. In addition to object-oriented modeling of applications, UML is also used for business-process modeling, data modeling, and XML modeling. Purpose of modeling: Models for software systems are as important as having a blueprint for a large building, or an outline for a book. Good models enable communication among project teams and assure architectural soundness. The more complex the software system, the more important it is to have models that accurately describe the system and are understood by everyone. UML helps provide this via a standard for graphical diagrams. Just like an architect can understand the notations on any blueprint, UML enables software engineers and business managers to understand the design of any software system, even if the original designers have long left the company. Organization behind it: Object Management Group (OMG) (www.omg.org). (The Resource Page at OMG Web site is www.omg.org/uml.) The OMG produces and maintains the UML standard, an internationally recognized standard. The OMG is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. Its membership roster (about 800) includes just about every large company in the industry and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing are represented on the OMG's Board of Directors. Companies that contributed to the Standard: Realizing that UML would be strategic to their business, the following companies contributed their ideas to the first UML standard: Digital Equipment Corp, HP, i-Logix, IntelliCorp, IBM, Intel, Lotus Computing, MCI, Microsoft, Oracle, Rational Rose, TI, and Unisys. Companies that use UML: It is safe to say that all Fortune 1000 companies are currently using UML, or are moving toward UML to analyze and design their applications and systems. This includes companies from all vertical industries, from Coca Cola to Warner Brothers, from CVS Pharmacy to Lockheed Martin Aerospace. You name the company - if they have an IT department, they are using UML.

Explains how to leverage Java's architecture and mechanisms to design enterprise applications and considers code modularity, nonduplication, network efficiency, maintainability, and reusability.

An Introduction to Object-Oriented Modeling

Best Practices and Design Strategies

A Project-based Tutorial

Improving the Design of Existing Code

Planning Extreme Programming

Software Engineering Models, Patterns and Architectures for HCI

Design and Applications of Intelligent Agents

Users can dramatically improve the design, performance, and manageability of object-oriented code without altering its interfaces or behavior.

"Refactoring" shows users exactly how to spot the best opportunities for refactoring and exactly how to do it, step by step.

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.

A guide to XP leads the developer, project manager, and team leader through the software development planning process, offering real world examples and tips for reacting to changing environments quickly and efficiently.

Concise and easy-to-understand guidelines and standards for creating UML 2.0 diagrams.

UML 2. 0 in Action

A Brief Guide to the Standard Object Modeling Language

Fowler

"Applying Uml and Patterns:An Introduction to Object-Oriented Analysis and Design and the Unified Process with Uml Distilled:A Brief Guide to the Standard Object Modeling Language

Conceptual Modeling for E-Business and the Web

Reliable Software Technologies -- Ada-Europe 2003

Offers comprehensive coverage of all major modeling viewpoints Provides details of collaboration and class diagrams for filling in the design-level models

This book constitutes the refereed proceedings of workshops, held at the 30th International Conference on Conceptual Modeling, ER 2011, in Brussels, Belgium in October/November 2011. The 31 revised full papers presented together with 9 posters and demonstrations (out of 88 submissions) for the workshops and the 6 papers (out of 11 submissions) for the industrial track were carefully reviewed and selected. The papers are organized in sections on the workshops Web Information Systems Modeling (WISM); Modeling and Reasoning for Business Intelligence (MORE-BI); Software Variability Management (Variability@ER); Ontologies and Conceptual Modeling (Onto.Com); Semantic and Conceptual Issues in GIS (SeCoGIS); and Foundations and Practices of UML (FP-UML).

"Designing a large software system is an extremely complicated undertaking that requires juggling differing perspectives and differing goals, and evaluating differing options. Applied Software Architecture is the best book yet that gives guidance as to how to sort out and organize the conflicting pressures and produce a successful design." -- Len Bass, author of Software Architecture in Practice. Quality software architecture design has always been important, but in today's fast-paced, rapidly changing, and complex development environment, it is essential. A solid, well-thought-out design helps to manage complexity, to resolve trade-offs among conflicting requirements, and, in general, to bring quality software to market in a more timely fashion. Applied Software Architecture provides practical guidelines and techniques for producing quality software designs. It gives an overview of software architecture basics and a detailed guide to architecture design tasks, focusing on four fundamental views of architecture--conceptual, module, execution, and code. Through four real-life case studies, this book reveals the insights and best practices of the most skilled software architects in designing software architecture. These case studies, written with the masters who created them, demonstrate how the book's concepts and techniques are embodied in state-of-the-art architecture design. You will learn how to: create designs flexible enough to incorporate tomorrow's technology; use architecture as the basis for meeting performance, modifiability, reliability, and safety requirements; determine priorities among conflicting requirements and arrive at a successful solution; and use software architecture to help integrate system components. Anyone involved in software architecture will find this book a valuable compendium of best practices and an insightful look at the critical role of architecture in software development. 0201325713B07092001

The Definitive Refactoring Guide, Fully Revamped for Ruby With refactoring, programmers can transform even the most chaotic software into well-designed systems that are far easier to evolve and maintain. What's more, they can do it one step at a time, through a series of simple, proven steps. Now, there's an authoritative and extensively updated version of Martin Fowler's classic refactoring book that utilizes Ruby examples and idioms throughout-not code adapted from Java or any other environment. The authors introduce a detailed catalog of more than 70 proven Ruby refactorings, with specific guidance on when to apply each of them, step-by-step instructions for using them, and example code illustrating how they work. Many of the authors' refactorings use powerful Ruby-specific features, and all code samples are available for download. Leveraging Fowler's original concepts, the authors show how to perform refactoring in a controlled, efficient, incremental manner, so you methodically improve your code's structure without introducing new bugs. Whatever your role in writing or maintaining Ruby code, this book will be an indispensable resource. This book will help you * Understand the core principles of refactoring and the reasons for doing it * Recognize "bad smells" in your Ruby code * Rework bad designs into well-designed code, one step at a time * Build tests to make sure your refactorings work properly * Understand the challenges of refactoring and how they can be overcome * Compose methods to package code properly * Move features between objects to place responsibilities where they fit best * Organize data to make it easier to work with * Simplify conditional expressions and make more effective use of polymorphism * Create interfaces that are easier to understand and use * Generalize more effectively * Perform larger refactorings that transform entire software systems and may take months or years * Successfully refactor Ruby on Rails code

First International Workshop, Mulhouse, France, June 3-4, 1998, Selected Papers

Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India - Volume 2

Use Case Driven Object Modeling with UML Theory and Practice

Applying UML and Patterns Training Course

ER 2011 Workshops FP-UML, MoRE-BI, Onto-CoM, SeCoGIS, Variability@ER, WISM, Brussels, Belgium, October 31 - November 3, 2011

The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions

Human-Centered Software Engineering

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.

A detailed and practical book and eBook walk-through showing how to apply UML to real world development projects

The refereed proceedings of the 8th International Conference on Reliable Software Technologies, Ada-Europe 2003, held in Toulouse, France in June 2003. The 29 revised full papers presented together with 3 invited papers were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on Ravenscar, language issues, static analysis, distributed information systems, software metrics, software components, formal specification, real-time kernel, software testing, and real-time systems design.

For the last two decades, IS researchers have conducted empirical studies leading to a better understanding of the impact of Systems Analysis and Design methods in business, managerial, and cultural contexts. SA&D research has established a balanced focus not only on technical issues, but also on organizational and social issues in the information society..This volume presents the very latest, state-of-the-art research by well-known figures in the field. The chapters are grouped into three categories: techniques, methodologies, and approaches.

A Desktop Seminar from Craig Larman

Uml Distilled: A Brief Guide To The Standard Object Modeling Language, 3/E

UML for Systems Engineering

Learning UML 2.0

Theory and Practice

Pattern Enterpr Applica Arch

UML @ Classroom

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.

Ebook: Object-Oriented Systems Analysis and Design Using UML

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

"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.

UML in Practice

TC2 First Working IFIP Conference on Software Architecture (WICSA1) 22-24 February 1999, San Antonio, Texas, USA

The Requirements Engineering Handbook

Learning UML

UML Weekend Crash Course

Advances in Conceptual Modeling. Recent Developments and New Directions

The Unified Modeling Language. "UML"'98: Beyond the Notation

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

Now widely adopted as the de facto industry standard and sanctioned by the Object Management Group, the Unified Modeling Language (UML) is a notation all software developers need to know and understand. However, the UML is a big language, and not all of it is equally important. The award-winning first edition of UML Distilled was widely praised for being a concise guide to the core parts of the UML and has proved extremely successful in helping developers get up and running quickly. UML Distilled, Second Edition, maintains the concise format with significantly updated coverage of use cases and activity diagrams, and expanded coverage of collaborations. It also includes a new appendix detailing the changes between UML versions. Written for those with a basic understanding of object-oriented analysis and design, this book begins with a summary of UML's history, development, and rationale and then moves into a discussion of how the UML can be integrated into the object-

oriented development process. The primary author profiles the various modeling techniques in the UML--such as use cases, class diagrams, and interaction diagrams--and describes the notation and semantics clearly and succinctly. He also outlines useful non-UML techniques such as CRC cards and patterns. These descriptions are made even more relevant with a collection of best practices based on the primary author's experience and a brief Java programming example demonstrating the implementation of a UML-based design. With this tour of the key parts of the UML, readers will be left with a firm foundation upon which to build models and develop further knowledge of the Unified Modeling Language. Praise for the First Edition " UML Distilled is a recipient of the prestigious 1997 Software Development Magazine Productivity Award in the Books category. Addison-Wesley congratulates authors Martin Fowler and Kendall Scott for their outstanding work." "This book is a godsend. It is packed with solid advice presented in a concise and highly readable way. The essence of the notations is explained very well indeed but the author goes beyond this to give very clear insights into the application of UML techniques."--Jennifer Stapleton, Vice President Technical, British Computer Society " UML Distilled is well written, knowledgeable about both systems development and UML, and disarmingly honest." -- Robert L. Glass, The Software Practitioner (March 1998) " UML Distilled proves that you can say a lo ...

This volume contains mainly the revised versions of papers presented at the workshop '98, "Beyond the Notation", that took place in Mulhouse, France on June 3-4, 1998. We thank all those that have made this possible, and particularly all the people in Mulhouse that worked hard to make this meeting a success, with such a short delay between the announcement and the realization. We are specially grateful to Nathalie Gaertner, who put in a tremendous amount of effort in the initial preparation of the workshop. We were pleasantly surprised of the quality of the submitted material and of the level of the technical exchanges at the Mulhouse meeting. More than one hundred attendees, from about twenty different countries, representing the main actors in the UML research and development scene, gathered in Mulhouse for two full study days. We would like to express our deepest appreciation to the authors of submitted papers, the editorial committee for this volume, the program committee for the initial workshop, the external referees, and many others who contributed towards the final contents of this volume. April 1999 Jean Bézivin Pierre-Alain Muller This book constitutes the thoroughly refereed post-proceedings of the 9th International Conference on Real-Time and Embedded Systems and Applications, RTCSA 2003, held in Tainan, Taiwan, in February 2003. The 28 revised full papers and 9 revised short papers presented were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on scheduling, networking and communication, embedded systems and environments, pervasive and ubiquitous computing, systems and architectures, resource management, file systems and databases, performance analysis, and tools and development.

Ebook: Object-Oriented Systems Analysis and Design Using UML

ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and the World Wide Web and Conceptual Modeling, Salt Lake City, Utah, USA, October 9-12, 2000 Proceedings

Modeling XML Applications with UML

SysML Distilled

Core J2EE Patterns

Real-Time and Embedded Computing Systems and Applications

A Brief Guide to the Systems Modeling Language

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 be successful.

Activity theory is a way of describing and characterizing the structure of human activity of all kinds. First introduced by Russian psychologists Rubinshtein, Leontiev, and Vigotsky in the early part of the last century, activity theory has more recently gained increasing attention among interaction designers and others in the human-computer interaction and usability communities (see, for example, Gay and H-brooke, 2004). Interest was given a significant boost when Donald Norman suggested activity-theory and activity-centered design as antidotes to some of the putative ills of "human-centered design" (Norman, 2005). Norman, who has been credited with coining the phrase "user-centered design," suggested that too much attention focused on human users may be harmful, that to design better tools designers need to focus not so much on users as on the activities in which users are engaged and the tasks they seek to perform within those activities. Although many researchers and practitioners claim to have used or been influenced by activity theory in their work (see, for example, Nardi, 1996), it is often difficult to trace precisely where or how the results have actually been shaped by activity theory. In many cases, even detailed case studies report results that seem only distantly related, if at all, to the use of activity theory. Contributing to the lack of precise and traceable impact is that activity theory, in spite of its name, is not truly a formal and proper theory.

The international conference on Advances in Computing and Information technology (ACITY 2012) provides an excellent international forum for both academics and professionals for sharing knowledge and results in theory, methodology and applications of Computer Science and Information Technology. The Second International Conference on Advances in Computing and Information technology (ACITY 2012), held in Chennai, India, during July 13-15, 2012, covered a number of topics in all major fields of Computer Science and Information Technology including: networking and communications, network security and applications, web and internet computing, ubiquitous computing, algorithms, bioinformatics, digital image processing and pattern recognition, artificial intelligence, soft computing and applications. Upon a strength review process, a number of high-quality, presenting not only innovative ideas but also a founded evaluation and a strong argumentation of the same, were selected and collected in the present proceedings, that is composed of three different volumes.

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.

Watching the Wheels, 2nd Edition

Ruby Edition: Ruby Edition

Systems Analysis and Design: People, Processes, and Projects

A Brief Guide to the Standard Object Modeling Language, Second Edition

UML Distilled

XML is rapidly becoming the standard platform for delivering e-Business information and integrating e-Business systems. XML developers desperately need

mature software development processes and tools for developing effective applications. David Carlson fills the gap, showing exactly how to leverage the worldwide UML standard for modeling complex systems in advanced XML development. In *Modeling XML Applications with UML*, he presents the first comprehensive framework for modeling communications in any B2B software system. Carlson presents in-depth coverage of UML-based analysis, design, and modeling of XML content within e-Business environments. The book includes detailed coverage of using UML to support the creation of new XML-based B2B vocabularies and industry portals that reflect the requirements of several key stakeholder communities, including consumers, business analysts, web application specialists, system integration specialists, and content developers. Carlson presents several B2B use cases, and then decomposes them into scenarios illustrated with class diagrams, sequence diagrams, and activity diagrams showing how XML fits into an overall e-Business solution. Each chapter concludes with "steps for success" that distill UML's general principles into specific recommendations for action.

"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, interactive overview, and timing diagrams."--P. [4] of cover.

This second edition provides updates to the UML and is the best resource for quick, no-nonsense explanations of using UML. The major strength is its short, concise presentation of the essentials of UML and where it fits within the software-development process.