Activity Diagram Reverse Engineering

Object Oriented Modeling And Design With UMLMastering UML with Rational Rose 2002Computational Science and Its Applications -- ICCSA 2015Programming with Microsoft Visual C++ .NETNinth Working Conference on Reverse EngineeringThe C++ ReportISESE '06The Journal of the Computer Society of IndiaApplications of Graph Transformations with Industrial RelevanceUML Weekend Crash CourseWeb EngineeringProceedingsProceedings of the National Conference on Computing for Nation DevelopmentSoftware Engineering and Computer GamesMSDN Magazine11th IEEE International Conference and Workshop on the Engineering of Computer-Based SystemsObject-Oriented Design And PatternsProceedingsReverse EngineeringReverse Engineering of Object Oriented CodeProduct DesignJournal of Engineering EducationBuilding Web Applications with UMLEmerging Technologies for the Evolution and Maintenance of Software ModelsAgile Project Management with ScrumConference ProceedingsBetter Software FasterObject Oriented Modeling & DesignVisual Modeling with Rational Rose 2002 and UMLC/C++ Users JournalVerification and Validation for Quality of UML 2.0 ModelsCOMPSAC 2001Dr. Dobb's JournalProfessional UML Using Visual Studio .NetProceedings of the Third Working Conference on Reverse Engineering, November 8-10, 1996, Monterey, CaliforniaModel-Based Software and Data IntegrationRapid J2EE DevelopmentProceedings of the ASME Design Engineering Technical ConferencesTechnology of Object-oriented Languages and Systems : TOOLS 30ENTERprise Information Systems

Object Oriented Modeling And Design With UML

Mastering UML with Rational Rose 2002

Computational Science and Its Applications -- ICCSA 2015

Here are selected papers of the First International Workshop on Model-Based Software and Data Integration 2008, part of the Berlin Software Integration Week 2008. The 9 revised full papers and 3 invited lectures were carefully picked from numerous submissions.

Programming with Microsoft Visual C++ .NET

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

enhance 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 can be 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). (UML 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 computer 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 UML 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, ICON 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 model and design their applications and systems. This includes companies from all vertical industries, from Coca Cola to Warner Brothers, from CVS Pharmacy to Lockhead Martin Aerospace. You name the company - if they have an IT department, they are using UML.

Ninth Working Conference on Reverse Engineering

The C++ Report

Reverse engineering encompasses a wide spectrum of activities aimed at extracting information on the function, structure, and behavior of man-made or natural artifacts. Increases in data sources, processing power, and improved data mining and processing algorithms have opened new fields of application for reverse engineering. In this book, we present twelve applications of reverse engineering in the software engineering, shape engineering, and medical and life sciences application domains. The book can serve as a guideline to practitioners in the above fields to the state-of-the-art in reverse engineering techniques, tools, and use-cases, as well as an overview of open challenges for reverse engineering researchers.

ISESE '06

The five-volume set LNCS 9155-9159 constitutes the refereed proceedings of the 15th International Conference on Computational Science and Its Applications, ICCSA 2015, held in Banff, AB, Canada, in June 2015. The 232 revised full papers presented in 22 workshops and a general track were carefully reviewed and selected from 780 initial submissions for inclusion in this volume. They cover various areas in computational science ranging from computational science

technologies to specific areas of computational science such as computational geometry and security.

The Journal of the Computer Society of India

What is this book about? If you want to use Visio to create enterprise software, this is the book for you. The integration of Visual Studio .NET Enterprise Architect and Visio for Enterprise Architects provides a formidable tool. Visio offers powerful diagramming capabilities, including such things as creating UML models, mapping out databases with Entity Relationship diagrams, and aiding the development of distributed systems. Its integration with Visual Studio .NET Enterprise Architect means that C# or Visual Basic .NET code can be generated from the UML diagrams, and Visual Studio .NET projects can be reverse engineered to UML models. For the developer already familiar with UML and looking to get the best out of Visio, the Visual Studio .NET and Visio for Enterprise Architects combination is weakly documented, and the quality information needed to realize the time-saving features of Visio just does not seem to be available, until now. This book presumes that you are already familiar with the basic concepts of UML notation — this book will not teach you UML. Instead, this book will take you forward into the Visio environment, showing you how to make the most of its software related features. What does this book cover? In this book, you'll learn how to Diagram business components in Visio Generate code from a UML model Reverse engineer Visual Studio .NET projects into a UML model Reverse engineer into a UML model without source code Document the project with UML and Visio Design distributed applications with Visio's diagrams Work with Entity Relationship database modeling, and round-trip engineering for database design

Applications of Graph Transformations with Industrial Relevance

UML Weekend Crash Course

This book solves the dilemma of wanting to learn Windows-based sorfware engineering without knowing Windows programming. The basics in Windows programming are explained alongside ideas of object-oriented sortware engineering. (Midwest).

Web Engineering

Introduction and summary of web-related technogies; Web application basics; Dynamic clients; Beyond http and html; Security; The process; Defining the architecture; Requirements and use case; Analysis; Design; Implementation.

Proceedings

Proceedings of the National Conference on Computing for Nation Development

Software Engineering and Computer Games

MSDN Magazine

11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems

Product Design presents an in-depth study of structured designed processes and methods. KEY TOPICS: Fundamental approach is that reverse engineering and teardowns offer a new better paradigm for design instruction, permitting a modern learning cycle of experience, hypothesis, understanding, and then execution. MARKET: For practicing engineers interested in learning about mechanical design. FEATURES/BENEFITS Fundamental approach is that reverse engineering and teardowns offer a new better paradigm for design instruction, permitting a modern learning cycle of experience, hypothesis, understanding, and then execution. Concrete experiences with hands-on products. Applications of contemporary technologies. Studies of systematic experimentation. Exploration of the boundaries of design methodology. Decision making for real product development. Discusses the foundation material of product design, including a philosophy for learning and implementing product design methods. Each chapter includes both basic and advanced techniques for particular phases of product development.

Object-Oriented Design And Patterns

Model-driven software development drastically alters the software development process, which is characterized by a high degree of innovation and productivity. Emerging Technologies for the Evolution and Maintenance of Software Models contains original academic work about current research and research projects related to all aspects affecting the maintenance, evolution, and reengineering (MER), as well as long-term management, of software models. The mission of this book is to present a comprehensive and central overview of new and emerging trends in software model research and to provide concrete results from ongoing developments in the field.

Proceedings

Papers from an October 2001 address such themes as requirements engineering, component-based development, protocols and harmonization, quality management, software architecture, workflow systems, and software testing, distributed systems, UML, commercial off-the-shelf components, e-learning applicat

Reverse Engineering

The August 1999 conference concentrated on the delivery of high-quality software

on schedule and within budget, offering practical experience from both industry and academia. The 37 technical papers provide insights from lessons learned on real projects, covering such topics as databases, object-ori

Reverse Engineering of Object Oriented Code

OMG Standards : MDA, MOF, XMI, CORBA, UML 2.0. UML History, UML 2.0 New Features. Rational Unified Process emphasizing Inception, Elaboration, Construction, Transition Phases. 4+1 View architecture, Architectural approaches : Use case Centric, Architecture driven, Iterative approach, OO Concepts Review.UML : UML MetaModel. Extensibility mechanisms like stereotypes, tagged values, constraints and profiles. OCL. Overview of all diagrams in UML 2.0.Object diagrams, CRC method, Review of OO concepts. Class diagrams, Classes and Relationships, Interfaces and ports, Templates, Active Objects, Advanced relationships generalization, association, aggregation, dependencies. Composite structure diagrams including composite structures, collaborations.Interaction diagrams. Interaction Overview diagrams including interactions, signals, exceptions, regions, partitions, Sequence diagrams, Communication diagrams.State Machine diagrams, States, encapsulation of states, transitions, submachine, state generalization. Timing diagrams, Activity diagrams, Activities, sub activities, signals, exceptions, partitions, regions.Support for modeling Architecture in UML. Package diagrams, Component diagrams, Deployment diagrams. Applications of UML in embedded systems, Web applications, commercial applications. All diagrams are to be assumed for UML 2.0 for each diagram the need, purpose, Concepts, Notation, Forward Engineering, Reverse Engineering and Application must be considered.

Product Design

This three-volume-set (CCIS 219, CCIS 220, and CCIS 221) constitutes the refereed proceedings of the International Conference on ENTERprise Information Systems, CENTERIS 2011, held in Vilamoura, Portugal, in September 2011. The approx. 120 revised full papers presented in the three volumes were carefully reviewed and selected from 180 submissions. The papers are organized in topical sections on knowledge society, EIS adoption and design, EIS implementation and impact, EIS applications, social aspects and IS in education, IT/IS management, telemedicine and imaging technologies, healthcare information management, medical records and business processes, decision support systems and business intelligence in health and social care contexts, architectures and emerging technologies in healthcare organizations, as well as m-health.

Journal of Engineering Education

Building Web Applications with UML

TogetherSoft's integrated lifecycle tools allow software teams to achieve breakthrough quality, efficiency, and performance. In Better Software Faster, two leading Together experts share insights, examples, and techniques for succeeding

with Together every step of the way: through planning, requirements, modeling, design, architecture, development, debugging, implementation, and beyond. Contains solutions for every team member: analysts, architects, designers, developers, and managers.

Emerging Technologies for the Evolution and Maintenance of Software Models

Cay Horstmann offers readers an effective means for mastering computing concepts and developing strong design skills. This book introduces object-oriented fundamentals critical to designing software and shows how to implement design techniques. The author's clear, hands-on presentation and outstanding writing style help readers to better understand the material.· A Crash Course in Java· The Object-Oriented Design Process· Guidelines for Class Design· Interface Types and Polymorphism· Patterns and GUI Programming· Inheritance and Abstract Classes· The Java Object Model· Frameworks· Multithreading· More Design Patterns

Agile Project Management with Scrum

Conference Proceedings

The J2EE developer's practical introduction and cookbook to cost saving software engineering solutions.

Better Software Faster

Design More Efficient Applications with the Leading Visual Modeler Mastering UML with Rational Rose 2002 offers expert instruction in both areas you need to master if you want to develop flexible object-oriented applications: the Unified Modeling Language and the latest version of Rational Rose, the world's leading visual modeling tool. But this book goes far beyond modeling. It teaches you to use Rose to turn your UML diagrams into code--automatically--in the language of your choice. And it's newly expanded to provide valuable information on business modeling, web modeling, new Java functionality, and XML DTDs. Coverage includes: * Understanding UML, with a bonus "Getting Started with UML" appendix * Finding your way around Rational Rose * Creating UML diagrams of all kinds * Creating a detailed object model * Creating a detailed data model * Modeling your XML DTDs * Generating code automatically * Handling language-specific code-generation issues * Reverse-engineering an existing application * Using round-trip engineering techniques

Object Oriented Modeling & Design

A practical approach to enhancing quality in software models using UML Version 2.0 "Despite its increasing usage, many companies are not taking the best advantage of UML and, occasionally, individuals have experienced frustration in applying its standards. Perhaps this is because they have not yet read this book!" —From the Foreword by Prof. Brian Henderson-Sellers This book presents a

practical checklist approach to enhancing the quality of software models created with the Unified Modeling Language (UML) Version 2.0. The foundation for quality is set by the discussion on the nature and creation of UML models. This is followed by a demonstration of how to apply verification and validation checks to these models with three foci: syntactical correctness, semantic meaningfulness, and aesthetic symmetry. The quality work is carried out within three distinct yet related modeling spaces: Model of problem space (MOPS) Model of solution space (MOSS) Model of background space (MOBS) Readers can then choose a specific quality approach according to their roles in their projects. Verification and validation checks are also organized according to these three modeling spaces, making it easier for the reader to focus on the appropriate diagrams and quality checks corresponding to their modeling space. In addition, a major element of this publication is the Strengths, Weaknesses, Objectives, and Traps (SWOT) analysis. This analysis is performed on each UML diagram, enabling readers to fully comprehend these diagrams, their advantages and limitations, and the way in which they can be used in practical projects for modeling. A consistent case study of the Lucky Insurance System is provided throughout the chapters to illustrate the creation of good quality UML diagrams, followed by application of quality checks to them. With its emphasis on quality in UML-based projects, this book is an essential resource for all quality professionals, including quality analysts, process consultants, quality managers, test designers, and testers.

Visual Modeling with Rational Rose 2002 and UML

This book constitutes the thoroughly refereed post-proceedings of the International Workshop on Graph Transformation with Industrial Relevance, AGTIVE'99, held in Kerkrade, The Netherlands, in June 1999. The 28 revised full papers presented went through an iterated process of reviewing and revision. Also included are three invited papers, 10 tool demonstrations, a summary of a panel discussion, and lists of graph transformation systems and books on graph transformations. The papers are organized in sections on modularization concepts, distributed systems modeling, software architecture: evolution and reengineering, visual graph transformation languages, visual language modeling and tool development, knowledge modeling, image recognition and constraint solving, process modeling and view integration, and visualization and animation tools.

C/C++ Users Journal

Verification and Validation for Quality of UML 2.0 Models

COMPSAC 2001

The Unified Modeling Language™ (UML®) is inherently object-oriented modeling language and was designed for use in object-oriented software applications. The applications could be based on the object-oriented technologies recommended by the Object Management Group (OMG), which owns the UML. The initial versions of UML (UML 1.x) were based on three leading object-oriented methods - Booch, OMT,

and OOSE, to represent "the culmination of best practices in practical object-oriented modeling". UML 2.x is still object-oriented in its core (though there were some apparently unsuccessful attempts to extend UML to support other development methods). This book provides practical guidance on the modeling and design of object-oriented systems. Its specific goals are the following: ■ To provide a sound understanding of the fundamental concepts and historical evolution of the object model. ■ To facilitate a mastery of the notation and process of object-oriented modelling and design. ■ To teach the realistic application of object-oriented modelling and design within a variety of problem domains. The concepts presented all stand on a solid theoretical foundation, but this is primarily a pragmatic book that addresses the practical needs and concerns of software engineering practitioners, from the architect to the software developer.

Dr. Dobb's Journal

Describes how to design object-oriented code and accompanying algorithms that can be reverse engineered for greater flexibility in future code maintenance and alteration. Provides essential object-oriented concepts and programming methods for software engineers and researchers.

Professional UML Using Visual Studio .Net

Proceedings of the Third Working Conference on Reverse Engineering, November 8-10, 1996, Monterey, California

The rules and practices for Scrum—a simple process for managing complex projects—are few, straightforward, and easy to learn. But Scrum’s simplicity itself—its lack of prescription—can be disarming, and new practitioners often find themselves reverting to old project management habits and tools and yielding lesser results. In this illuminating series of case studies, Scrum co-creator and evangelist Ken Schwaber identifies the real-world lessons—the successes and failures—culled from his years of experience coaching companies in agile project management. Through them, you’ll understand how to use Scrum to solve complex problems and drive better results—delivering more valuable software faster. Gain the foundation in Scrum theory—and practice—you need to: Rein in even the most complex, unwieldy projects Effectively manage unknown or changing product requirements Simplify the chain of command with self-managing development teams Receive clearer specifications—and feedback—from customers Greatly reduce project planning time and required tools Build—and release—products in 30-day cycles so clients get deliverables earlier Avoid missteps by regularly inspecting, reporting on, and fine-tuning projects Support multiple teams working on a large-scale project from many geographic locations Maximize return on investment!

Model-Based Software and Data Integration

This title is the complete programming reference for intermediate and expert developers who want to create .NET applications with Visual C++ and the .NET

Framework. Focusing on core programming techniques, instructions, and solutions, this book is designed to help developers who are already familiar with Visual C++.

Rapid J2EE Development

Proceedings of the ASME Design Engineering Technical Conferences

Comprises the proceedings of the Third Working Conference on Reverse Engineering held in Monterey in November 1996. The 30 contributions contained in this volume cover a range of topics including experiments with large systems, experiments for evaluation, user interface migration, reverse engineering binary and assembler code, object model transformation, reengineering infrastructure, wrapping, data reverse engineering, visualizing recovered architectures, recovering objects, recognition, and domain-oriented recovery. Lacks a subject index. Annotation copyrighted by Book News, Inc., Portland, OR.

Technology of Object-oriented Languages and Systems : TOOLS 30

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.

ENTERprise Information Systems