Uml Diagram For Platform Assignment Railway

This book highlights cutting-edge research presented at the third installment of the International Conference on Smart City Applications (SCA2018), held in Tétouan, Morocco on October 10–11, 2018. It presents original research results, new ideas, and practical lessons learned that touch on all aspects of smart city applications. The respective papers share new and highly original results by leading experts on IoT, Big Data, and Cloud technologies, and address a broad range of key challenges in smart cities, including Smart Education and Intelligent Learning Systems, Smart Healthcare, Smart Building and Home Automation, Smart Environment and Smart Agriculture, Smart Economy and Digital Business, and Information Technologies and Computer Science, among others. In addition, various novel proposals regarding smart cities are discussed. Gathering peer-reviewed chapters written by prominent researchers from around the globe, the book offers an invaluable instructional and research tool for courses on computer and urban sciences; students and practitioners in computer science, information science, technology studies and urban management studies will find it particularly useful. Further, the book is an excellent reference guide for professionals and researchers working in mobility, education, governance, energy, the environment and computer sciences.

Software -- Software Engineering.

Software engineering for complex systems requires abstraction, multi-domain expertise, separation of concerns, and reuse. Domain experts rarely are software engineers and should formulate solutions using their domain's vocabulary instead of general purpose programming languages (GPLs). Successful integration of domain-specific languages (DSLs) into a software system requires a separation of concerns between domain issues and integration issues while retaining a loose enough coupling to support DSL reuse in different contexts. Component-based software engineering (CBSE) increases reuse and separation of concerns by encapsulating functionalities in components. Components are GPL artifacts, which raises accidental complexities. Model-driven engineering (MDE) abstracts from GPLs by lifting models to primary development artifacts. Models can be abstract and better comprehensible by using domain vocabulary instead of a GPL. They can be platform-independent and translated into GPLs for different target platforms. Component & connector (C&C) architecture description languages (ADLs) combine CBSE and MDE to compose of architectures from component models. We present concepts for engineering software systems with exchangeable component behavior languages. The concepts are realized in a software architecture modeling infrastructure that comprises modeling languages to develop applications based on C&C software architectures with exchangeable component behavior DSLs. It supports transformations from platform-independent to platform-specific software architectures and compositional code generation. With this, it enables domain experts to (re-)use the most appropriate component behavior DSL and facilitates composition of domain solutions through encapsulation in components.

Computers as Components: Principles of Embedded Computing System Design, Third Edition, presents essential knowledge on embedded systems technology and techniques. Updated for today's embedded systems design methods, this volume features new examples including digital signal processing, multimedia, and cyber-physical systems. It also covers the latest processors from Texas Instruments, ARM, and Microchip Technology plus software, operating systems, networks, consumer devices, and more. Like the previous editions, this textbook uses real processors to demonstrate both technology and techniques; shows readers how to apply principles to actual design practice; stresses necessary fundamentals that can be applied to evolving technologies; and helps readers gain facility to design large, complex embedded systems. Updates in this edition include: description of cyber-physical systems; exploration of the PIC and TI OMAP processors; high-level representations of systems using signal flow graphs; enhanced material on interprocess communication and buffering in operating systems; and design examples that include an audio player, digital camera, and cell phone. The author maintains a robust ancillary site at http://www.marilynwolf.us/CaC3e/index.html which includes a variety of support materials for instructors and students, including PowerPoint slides for each chapter; lab assignments developed for multiple systems including the ARM-based BeagleBoard computer; downloadable exercises solutions and source code; and links to resources and additional information on hardware, software, systems, and more. This book will appeal to students in an embedded systems design course as well as to researchers and savvy professionals schooled in hardware or software design. Description of cyber-physical systems: physical systems with integrated computation to give new capabilities Exploration of the PIC and TI OMAP multiprocessors High-level representations of systems using signal flow graphs Enhanced material on interprocess communication and buffering in operating systems Design examples include an audio player, digital camera, cell phone, and more

APPLYING UML & PATTERNS 3RD EDITION

Elements of Reusable Object-Oriented Software

Software Technologies

Guide to Web Application and Platform Architectures

From Concepts to Code

The Art, Philosophy, and Science of Object-oriented Programming

Model-Based Engineering of Embedded Real-Time Systems

The professional programmer's Deitel® guide to Java™ development and the powerful Java platform Written for programmers with a background in high-level language programming, this book applies the Deitel signature live-code approach to teaching programming and explores the Java language and Java APIs in depth. The book presents concepts in the context of fully tested programs, complete with syntax shading, code highlighting, line-by-line code walkthroughs and program outputs. The book features 200+ complete Java programs with 18,000+ lines of proven Java code, and hundreds of tips that will help you build robust applications. Start with an introduction to Java using an early classes and objects approach, then rapidly move on to more advanced topics, including GUI, graphics, exception handling, generics, collections, JDBC™, web-application development with JavaServer™ Faces, web services and more. You'll enjoy the Deitels' classic treatment of object-oriented programming and the OOD/UML® ATM case study, including a complete Java implementation. When you're finished, you'll have everything you need to build object-oriented Java applications.

Presents instructions on using MySQL, covering such topics as installation, querying, user management, security, and backups and recovery.

This book constitutes the refereed proceedings of the First European Conference, Workshops on Model Driven Architecture - Foundations and Applications, ECMDA-FA 2005, held in Nuremberg, Germany in November 2005. The 24 revised full papers presented, 9 papers from the applications track and 15 from the foundations track, were carefully reviewed and selected from 82 submissions. The latest and most relevant information on model driven software engineering in the industrial and academic spheres is provided. The papers are organized in topical sections on MDA development processes, MDA for embedded and real-time systems, MDA and component-based software engineering, metamodelling, model transformation, and model synchronization and consistency.

This book constitutes the thoroughly refereed proceedings of the 46th International Conference on Objects, Components, Models and Patterns, TOOLS EUROPE 2008, held in Zurich, Switzerland, in June/July 2008. The 21 papers presented in this book were carefully reviewed and selected from 58 submissions. TOOLS played a major role in the spread of object-oriented and component technologies. It has now broadened its scope beyond the original topics of object technology and component-based development to encompass all modern, practical approaches to software development. At the same time, TOOLS kept its traditional spirit of technical excellence, its acclaimed focus on practicality, its well-proven combination of theory and applications, and its reliance on the best experts from academia and industry.

46th International Conference, TOOLS EUROPE 2008, Zurich, Switzerland, June 30-July 4, 2008, Proceedings

Sun Certified Enterprise Architect for Java EE Study Guide

Learning MySQL

ANEMONA

5th International Workshop, AOSE 2004, New York, NY, USA, July 2004, Revised Selected Papers

Computers as Components

Objects, Components, Models and Patterns

This volume contains the papers presented at the 12th SDL Forum, Grimstad, Norway. The SDL Forum was ?rst held in 1982, and then every two years from 1985. Initially the Forum was concerned only with the Speci?cation and Description Language that was ?rst standardized in the 1976 Orange Book of the Inter- tional Telecommunication Union (ITU). Since then, many developments took place and the language has undergone several changes. However, the main underlying paradigm has survived, and it is the reason for the success of the Speci?cation and Description Language in many projects. This paradigm is based on the following important principles of distributed - plications: Communication: large systems tend to be described using smaller parts that communicate with each other; State: the systems are described on the basis of an explicit notion of state; State change: the behavior of the system is described in terms of (local) changes of the state. The original language is not the only representative for this kind of paradigm, so the scope of the SDL Forum was extended quite soon after the ?rst few events to also include other ITU standardized languages of the same family, such as MSC, ASN.1 and TTCN. This led to the current scope of System Design Languages coveringallstagesofthedevelopmentprocessincludinginparticularSDL,MSC, UML, ASN.1, eODL, TTCN, and URN. The focus is clearly on the advantages to users, and how to get from these languages the same advantage given by the ITU Speci?cation and Description Language: code generation from high-level speci?cations.

Thetopicof"Model-BasedEngineeringofReal-TimeEmbeddedSystems"brings together a challenging problem domain (real-time embedded systems) and a - lution domain (model-based engineering). It is also at the forefrontof integrated software and systems engineering, as software in this problem domain is an essential tool for system implementation and integration. Today, real-time - bedded software plays a crucial role in most advanced technical systems such as airplanes, mobile phones, and cars, and has become the main driver and - cilitator for innovation. Development, evolution, veri?cation, con?guration, and maintenance of embedded and distributed software nowadays are often serious challenges as drastic increases in complexity can be observed in practice. Model-based engineering in general, and model-based software engineering in particular, advocates the notion of using models throughout the development and life-cycle of an engineered system. Model-based software engineering re- forces this notion by promoting models not only as the tool of abstraction, but also as the tool for veri?cation, implementation, testing, and maintenance. The application of such model-based engineering techniques to embedded real-time systems appears to be a good candidate to tackle some of the problems arising in the problem domain.

This book constitutes the thoroughly refereed proceedings of the 11th International Joint Conference on Software Technologies, ICSOFT 2016, held in Lisbon, Portugal, in July 2016. The 13 revised full papers presented together with 3 short papers presented were carefully reviewed and selected from 84 submissions. The papers selected to be included in this book contribute to the understanding of relevant trends of current research on software technologies, including: Modelling for mobile devices Software and system testing Model-driven software development Reengineering systems for multi-tenancy Embedded and real-time systems reconguration Domain-specic languages and modelling Software and systems quality Context-aware and dynamically adapting software systems

This book presents the Proceedings of the Tenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, focusing on the theoretical aspects of intelligent systems research as well as extensions of theory of intelligent thinking machines.

Agent-Oriented Software Engineering V

The Common Component Modeling Example

Java¿ for Programmers

Software Engineering and Computer Systems, Part I

The Proceedings of the Third International Conference on Smart City Applications

Using UML, Patterns and Java

Beginning Java Objects

This two-volume set constitutes the refereed proceedings of the 16th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2022, held as part of the 24th International Conference, HCI International 2022, held as a virtual event, in June-July 2022. A total of 1271 papers and 275 posters included in the 39 HCII 2022 proceedings volumes. UAHCI 2022 includes a total of 73 papers; they focus on topics related to universal access methods, techniques and practices, studies on accessibility, design for all, usability, UX and technology acceptance, emotion and behavior recognition for universal access, accessible media, access to learning and education, as well universal access to virtual and intelligent assistive environments.

Definitive, Comprehensive SCEA Exam Prep–Straight from Sun's Exam Developers! This book delivers complete, focused review for Sun's new Sun Certified Enterprise Architect (SCEA) for Java EE certification exam—straight from two of the exam's creators! SCEA lead developer/assessor Mark Cade and SCEA lead developer/assessor Humphrey Sheil offer powerful insights, real-world architectural case studies, and challenging sample questions that systematically prepare you for the actual exam. For every question, the authors show why the right answers are right—and why the other answers are wrong. Cade and Sheil cover every SCEA exam topic, skill, and technique, including: Understanding system architecture and its goals Decomposing larger systems into components organized by tiers or layers Addressing requirements for scalability, maintainability, reliability, availability, extensibility, performance, and security Building effective web (presentation) tiers, and analyzing tradeoffs associated with using web frameworks Leveraging EJB 3's enhancements for business tier development Covering new enhancements in the JEE 5 platform Choosing and architecting the best integration and messaging components for your system Using the Java security model to enforce confidentiality, integrity, authorization, authentication, and non-repudiation Using the most powerful and useful Java EE architecture patterns Documenting Java EE architectures through visual models and narratives The authors also present detailed guidance for handling every element of the SCEA exam—including your development and defense of a complete real-world architectural solution.

The practicing programmer's DEITEL® guide to C# and the powerful Microsoft .NET Framework Written for programmers with a background in C++, Java, or other high-level languages, this book applies the Deitel signature live-code approach to teaching programming and explores Microsoft's C# language and the new .NET 2.0 in depth. The book is updated for Visual Studio® 2005 and C# 2.0, and presents C# concepts in the context of fully tested programs, complete with syntax shading, detailed line-by-line code descriptions, and program outputs. The book features 200+ C# applications with 16,000+ lines of proven C# code, as well as 300+ programming tips that will help you build robust applications. Start with a concise introduction to C# fundamentals using an early classes and objects approach, then rapidly move on to more advanced topics, including multithreading, XML, ADO.NET 2.0, ASP.NET 2.0, Web services, network programming, and .NET remoting. Along the way you will enjoy the Deitels' classic treatment of object-oriented programming and a new, OOD/UML™ ATM case study, including a complete C# implementation. When you are finished, you will have everything you need to build next-generation Windows applications, Web applications, and Web services. Dr. Harvey M. Deitel and Paul J. Deitel are the founders of Deitel & Associates, Inc., the internationally recognized programming languages content-creation and corporate-training organization. Together with their colleagues at Deitel & Associates, Inc., they have written many international best-selling programming languages textbooks that millions of people worldwide have used to master C, C++, Java™, C#, XML, Visual Basic®, Perl, Python, and Internet and Web programming. The DEITEL® Developer Series is designed for practicing programmers. The series presents focused treatments of emerging technologies, including .NET, J2EE, Web services, and more. Practical, Example-Rich Coverage Of: C# 2.0, .NET 2.0, FCL ASP.NET 2.0, Web Forms and Controls Database, SQL, and ADO.NET 2.0 Networking and .NET Remoting XML, Web Services Generics, Collections GUI/Windows® Forms OOP: Classes, Inheritance, and Polymorphism OOD/UML™ ATM Case Study Graphics and Multimedia Multithreading Exception Handling And more... VISIT WWW.DEITEL.COM Download code examples To receive updates on this book, subscribe to the free DEITEL® BUZZ ONLINE e-mail newsletter at www.deitel.com/newsletter/subscribe.html Read archived Issues of the DEITEL® BUZZ ONLINE Get corporate training information

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 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 Universal Access Handbook

4th International Conference, S-BPM ONE 2012, Vienna, Austria, April 4-5, 2012. Proceedings

Principles of Embedded Computing System Design

Application of Web Service Technologies on a B2B Communication Platform by Means of a Pattern and UML Based Software Development Process

Getting Started with BPM

12th International SDL Forum, Grimstad, Norway, June 20-23, 2005, Proceedings

Design Patterns

Java For Artists: The Art, Philosophy, and Science of Object-Oriented Programming is a Java programming language text/tradebook that targets beginner and intermediate Java programmers.

This book constitutes the proceedings of the 5th International IFIP Working Conference on Enterprise Interoperability (IWEI 2013), held in Enschede, The Netherlands, during March 27–28, 2013. The theme for IWEI 2013 was "Information Services and Processes for the Interoperable Economy and Society." The 15 full papers presented in this volume were carefully selected from 35 submissions. The selection was based on a thorough review process, in which each paper was reviewed by at least three experts in the field. The papers are representative of the current research activities in the area of enterprise interoperability and are grouped into six sections: enterprise service interoperability, enterprise interoperability in sectors, interoperability methodology, interoperability for specific application types, strategic and tactical aspects of enterprise interoperability, and ontology-based interoperability. Two keynotes and an invited paper complete this volume.

In recent years, the field of Universal Access has made significant progress in consolidating theoretical approaches, scientific methods and technologies, as well as in exploring new application domains. Increasingly, professionals in this rapidly maturing area require a comprehensive and multidisciplinary resource that addresses current principles, methods, and tools. Written by leading international authorities from academic, research, and industrial organizations and nonmarket institutions, The Universal Access Handbook covers the unfolding scientific, methodological, technological, and policy issues involved in the process of achieving universal access in the information society. In a collection of 61 chapters, the book discusses how to systematically apply universal design principles to information technologies. It explains the various dimensions of diversity in the technological platforms and contexts of use, including trends in mobile interaction and ambient intelligence environments. The implications of Universal Access on the development life cycle of interactive applications and services are unfolded, addressing user interface architectures and related components. Novel interaction methods and techniques for Universal Access are analyzed, and a variety of applications in diverse domains are discussed. The book reflects recent developments, consolidates present knowledge, and points towards new perspectives for the future. A quick glance through the contents demonstrates not only the breadth and depth of coverage but also the caliber of the contributions. An indispensable source of information for

interdisciplinary and cross-thematic study, the book provides a baseline for further in-depth studies, as well as an important educational tool in an increasingly globalized research and development environment.

This practical "how-to" guide to both using the ARIS Design Platform and how to use it to create real business models, follows Rob Davis' hugely successful Business Process Modelling with ARIS (Springer 2001). This second volume describes the new release of ARIS 7 Design Platform including ARIS Business Architect and ARIS Business Designer. Containing tips, techniques and short cuts gained from practical experience, this book show how to use ARIS in an easy way, supporting smart methods and smart models, and displays how ARIS can be used as a powerful tool for BPM. This book is a must-have guide and reference for all existing and new users of ARIS.

16th International Conference, UAHCI 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26 – July 1, 2022, Proceedings, Part I

UML Distilled

Real-Time Object Uniform Design Methodology with UML

ARIS Design Platform

S-BPM ONE - Education and Industrial Developments

Enterprise Interoperability

Cloud Computing with the Windows Azure Platform

This multi-function volume starts off as an ideal basic textbook for teaching object modeling, fundamental concepts learning and system designing with thirteen UML diagrams. But it also contains a whole section devoted to advanced research topics, samples and case studies. It is an essential work for any system developer or graduate student in a discipline that requires the power of object modeling as part of a development methodology.

Behavioral Specifications of Businesses and Systems deals with the reading, writing and understanding of specifications. The papers presented in this book describe useful and sometimes elegant concepts, good practices (in programming and in specifications), and solid underlying theory that is of interest and importance to those who deal with increased complexity of business and systems. Most concepts have been successfully used in actual industrial projects, while others are from the forefront of research. Authors include practitioners, business thinkers, academics and applied mathematicians. These seemingly different papers address different aspects of a single problem - taming complexity. Behavioral Specifications of Businesses and Systems emphasizes simplicity and elegance in specifications without concentrating on particular methodologies, languages or tools. It shows how to handle complexity, and, specifically, how to succeed in understanding and specifying businesses and systems based upon precise and abstract concepts. It promotes reuse of such concepts, and of constructs based on them, without taking reuse for granted. Behavioral Specifications of Businesses and Systems is the second volume of papers based on a series of workshops held alongside ACM's annual conference on Object-Oriented Programming Systems Languages and Applications (OOPSLA) and European Conference on Object-Oriented Programming (ECOOP). The first volume, Object-Oriented Behavioral Specifications, edited by Haim Kilov and William Harvey, was published by Kluwer Academic Publishers in 1996.

The explosive growth of application areas such as electronic commerce, ent- prise resource planning and mobile computing has profoundly and irreversibly changed our views on software systems. Nowadays, software is to be based on open architectures that continuously change and evolve to accommodate new components and meet new requirements. Software must also operate on di?- ent platforms, without recompilation, and with minimal assumptions about its operating environment and its users. Furthermore, software must be robust and` autonomous, capable of serving a naive user with a minimum of overhead and interference. Agent concepts hold great promise for responding to the new realities of software systems. They o?er higher-level abstractions and mechanisms which address issues such as knowledge representation and reasoning, communication, coordination, cooperation among heterogeneous and autonomous parties, p- ception, commitments, goals, beliefs, and intentions, all of which need conceptual modelling. On the one hand, the concrete implementation of these concepts can lead to advanced functionalities, e.g., in inference-based query answering, tra- action control, adaptive work?ows, brokering and integration of disparate inf- mation sources, and automated communication processes. On the other hand, their rich representational capabilities allow more faithful and ?exible treatments of complex organizational processes, leading to more e?ective requirements an- ysis and architectural/detailed design. The current trend of learner centeredness in education has been challenging many of the current ways of working, especially in higher education institutions. This rapid change in educational institutions demands educators acquire new sets of skills via continuous reflective practices. Hence, educators in higher education institutions are actively involved in research-driven teaching and learning practices. This change of role from mere content delivery to learning facilitators could be better achieved through a strong research-driven community of practice. Preparing 21st Century Teachers for Teach Less, Learn More (TLLM) Pedagogies is a pivotal reference source that provides vital research on the application of practice-based learning techniques in higher education institutions. This publication establishes a platform for academics to share their best practices to promote teach less, learn more pedagogies and learn reciprocally from the community of practice. While highlighting topics such as interactive learning, experiential technology, and logical thinking skills, this book is ideally designed for teachers, instructional designers, higher education faculty, deans, researchers, professionals, universities, academicians, and students seeking current research on transformative learning and future teaching practices.

Preparing 21st Century Teachers for Teach Less, Learn More (TLLM) Pedagogies

Java for Programmers _p2

First European Conference, ECMDA-FA 2005, Nuremberg, Germany, November 7-10, 2005, Proceedings

Proceedings of the Tenth International Conference

Proceedings

Industrial and Engineering Applications of Artificial Intelligence and Expert Systems

A Multi-agent Methodology for Holonic Manufacturing Systems

"This thoroughly explored text teaches students or industry R & D practitioners to successfully negotiate the terrain for building and maintaining large, complex software systems. The authors introduce the basic skills needed for a developer to apply software engineering techniques. Next, they focus on methods and technologies that enable developers to specify, design, and implement complex systems. Finally, the authors show how to support the system changes throughout the software life cycle."--BOOK JACKET.Title Summary field provided by Blackwell North America, Inc. All Rights Reserved

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.

ANEMONA is a multi-agent system (MAS) methodology for holonic manufacturing system (HMS) analysis and design. ANEMONA defines a mixed top-down and bottom-up development process, and provides HMS-specific guidelines to help designers identify and implement holons. The analysis phase is defined in two stages: System Requirements Analysis, and Holon Identification and Specification. This analysis provides high-level HMS specifications, adopting a top-down recursive approach which provides a set of elementary elements and assembling rules. The next stage is Holon Design, a bottom-up process to produce the system architecture from the analysis models. The Holons Implementation stage produces an Executable Code for the SetUp and Configuration stage. Finally, maintenances functions are executed in the Operation and Maintenance stage. The book will be of interest to researchers and students involved in artificial intelligence and software engineering, and manufacturing engineers in industry and academia.

This Three-Volume-Set constitutes the refereed proceedings of the Second International Conference on Software Engineering and Computer Systems, ICSECS 2011, held in Kuantan, Malaysia, in June 2011. The 190 revised full papers presented together with invited papers in the three volumes were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on software engineering; network; bioinformatics and e-health; biometrics technologies; Web engineering; neural network; parallel and distributed; e-learning; ontology; image processing; information and data management; engineering; software security; graphics and multimedia; databases; algorithms; signal processing; software design/testing; e- technology; ad hoc networks; social networks; software process modeling; miscellaneous topics in software engineering and computer systems.

Model Driven Architecture - Foundations and Applications

An Extensible Component & Connector Architecture Description Infrastructure for Multi-Platform Modeling

SDL 2005: Model Driven

Java for Artists

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

Component-based Product Line Engineering with UML

Export author Barker covers information key for proficiency with an OO programming language like Java, and shows how to really create reusable code and extensible applications.

Inhaltsangabe:Abstract: There are about 230.000 establishments in the Spanish hotel, restaurant and catering industry accounting for a turnover of about 15.6 thousand million euros. All of them sell food to private consumers. On the other hand suppliers and traders supply the catering trade with food and beverage products. Usually the different players in this food distribution network trade products by means of orders which are placed manually. This is a process which can be enhanced through the application of computer and Internet technology. A project lately also supported by the Spanish government is supposed to fill this gap. This project is called Catanet and has been established three years ago. Actually the Catanet platform is used by more than 100 customers, amongst them very important industry players like Lauren Films, Pepsi, Unilever and Nestle. Some of them will carry out a significant part of their overall food orders by the Catanet platform, which corresponds to a turnover volume of many million euros. In the former version of the Catanet platform clients had to apply a web page based interface in order to use the Catanet services. As this approach prohibited the full exploitation of the benefits the use of computer assistance provides (e.g. human participation still constitutes an inevitable and crucial part of the transaction, the interaction is completely asynchronous) an additional level is being added to the Catanet platform eliminating these shortcomings. During the time of this work the number of Catanet customers has grown explosively increasing also the diversity of the customer s computer systems. Additionally new subprojects could be launched due to the acquisition of a government grant. These encompassed among others new value added services demanded by the customers like an instant messaging module and a module for the automatic update of the local product catalogue. The characteristics of the IT infrastructure of the new customers which will carry out transactions with a serious turnover via the Catanet platform and the necessity to integrate the new subprojects required an adaptation of the design of the platform prototype which had been developed by this time and which is described in this work. Within this context the decision has been done to use .NET Framework based programs on the customer side instead of Java which had been used so far. The reasons for this were besides the easier integration with the IT [...]

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.

A cutting-edge, UML-based approach to software development and maintenance that integrates component-based and product-line engineering methods. - ripe market: development of component-based technologies is a major growth area - CBD viewed as a faster, more flexible way of building systems that can easily be adapted to meet rapidly-changing business needs and integrate legacy and new applications (e.g. Forrester report in June 1998 predicted that by 2001 "half of packaged apps vendors will deliver component-based apps"; e.g. Butler Group Management Briefing (2000): "Butler Group is now advising that all new-build and significant modification activity should be based on component architectures...Butler Group belives that Component-Based Development is one of the most important events in the evolution of information technology" e.g. Gartner Group estimates that "by 2003, 70% of new applications will be deployed as a combination of pre-assembled and newly created components integrated to form complex business-systems. The book defines, describes and shows how to use a method for component-based product-line engineering, supported by UML. This method aims to dramatically increase the level of reuse in software development by integrating the strengths of both of these approaches. UML is used to describe components during the analysis, design & implementation stages and capture their characteristics and relationships.This method includes two new kinds of extensions to the UML: new stereotypes to capture KobrA-specific concepts and new metamodel elements to capture variabilities. The method makes components the focus of the entire software development process, not just the implementation and deployment phases. The method has grown out of work by two companies in industry (Softlab & Psipenta) and two research organizations (GMD FIRST & Fraunhofer IESE) called the KobrA project. It is influenced by a number of successful existing methods e.g. Fusion method, Cleanroom method, Catalysis & Rational Unified Process, integrated with new ideas in an innovative way. Benefits for the reader: - gain a clear understanding of the product-line and component-based approaches to software development - learn how to use UML to describe components in analysis, design and implementation of components - learn how to develop and apply component-based frameworks in product-lines - learn how to build new systems from pre-existing components and ensure that components are of a high quality The book also includes: - case studies: library system example running throughout the chapters; ERP/business software system as appendix or separate chapter - bibliography - glossary - appendices covering: UML profiles, concise process description in the form of UML activity diagrams, refinement/translation patterns AUDIENCE Software engineers, architects & project managers. Software engineers working in the area of distributed/enterprise systems who want a method for applying a component-based or product-line engineering approach in practice.

Web Engineering for Workflow-based Applications

Object-oriented Software Engineering

C# for Programmers

Proceedings of the NATO Advanced Study Institute on Engineering Theories of Software Intensive Systems, Marktoberdorf, Germany, from 3 to 15 August 2004

Comparing Software Component Models

Universal Access in Human-Computer Interaction. Novel Design Approaches and Technologies

A Brief Guide to the Standard Object Modeling Language

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

This book constitutes the thoroughly refereed proceedings of the industrial track of the 4th International Conference on Subject-Oriented Business Process Management, S-BPM ONE 2012, held in Vienna, Austria, in April 2012. S-BPM as a discipline is characterized by a seamless approach toward the analysis, modeling, implementation, execution, and maintenance of business processes, with an explicit stakeholder focus. The 19 papers included were selected from the practically oriented submissions, and they have gone through the same rigorous peer-review process as their scientific counterparts.

Software engineering has over the years been applied in many different fields, ranging from telecommunications to embedded systems in car and aircraft industry as well as in production engineering and computer networks. Foundations in software technology lie in models allowing to capture application domains, detailed requirements, but also to understand the structure and working of software systems like software architectures and programs. These models have to be expressed in techniques based on discrete mathematics, algebra and logics. However, according to the very specific needs in applications of software technology, formal methods have to serve the needs and the quality of advanced software engineering methods, especially taking into account security aspects in Information Technology. This book presents mathematical foundations of software engineering and state-of-the-art engineering methods in their theoretical substance in the step towards practical applications to examine software engineering techniques and foundations used for industrial tasks. The contributions in this volume emerged from lectures of the 25th International Summer School on Engineering Theories of Software Intensive Systems, held at Marktoberdorf, Germany from August 3 to August 15, 2004.

New concepts and technologies are being introduced continuously for application development in the World-Wide Web. Selecting the right implementation strategies and tools when building a Web application has become a tedious task, requiring in-depth knowledge and significant experience from both software developers and software managers. The mission of this book is to guide the reader through the opaque jungle of Web technologies. Based on their long industrial and academic experience, Stefan Jablonski and his coauthors provide a framework architecture for Web applications which helps choose the best strategy for a given project. The authors classify common technologies and standards like .NET, CORBA, J2EE, DCOM, WSDL and many more with respect to platform, architectural layer, and application package, and guide the reader through a three-phase development process consisting of preparation, design, and technology selection steps. The whole approach is exemplified using a real-world case: the architectural design of an order-entry management system.

Engineering Theories of Software Intensive Systems

International Dagstuhl Workshop, Dagstuhl Castle, Germany, November 4-9, 2007. Revised Selected Papers

Behavioral Specifications of Businesses and Systems

11th International Joint Conference, ICSOFT 2016, Lisbon, Portugal, July 24-26, 2016, Revised Selected Papers

Second International Conference, ICSECS 2011, Kuantan, Malaysia, June 27-29, 2011. Proceedings, Part I

An Introduction to Object-Oriented Modeling

Models, Systems and Methodologies

Based on the 2007 Dagstuhl Research Seminar CoCoME, this book defines a common example for modeling approaches of component-based systems. The book makes it possible to compare different approaches and to validate existing models.

UML @ Classroom
Innovations in Smart Cities Applications Edition 2
5th International IFIP Working Conference, IWEI 2013, Enschede, The Netherlands, March 27-28, 2013, Proceedings