

## Component Based Product Line Engineering With Uml Component Based Development Series

Many approaches to creating Software Product Lines have emerged that are based on Model-Driven Engineering. This book introduces both Software Product Lines and Model-Driven Engineering, which have separate success stories in industry, and focuses on the practical combination of them. It describes the challenges and benefits of merging these two software development trends and provides the reader with a novel approach and practical mechanisms to improve software development productivity. The book is aimed at engineers and students who wish to understand and apply software product lines and model-driven engineering in their activities today. The concepts and methods are illustrated with two product line examples: the classic smart-home systems and a collection manager information system.

As a result of the open-source movement there is now a great deal of reusable software available in the public domain. This offers significant functionality that commercial software vendors can use in their software projects. Open-source approaches to software development have illustrated that complex, mission critical software can be developed by distributed teams of developers sharing a common goal. Commercial software vendors have an opportunity to both learn from the open-source community as well as leverage that knowledge for the benefit of its commercial clients. Nonetheless, the open-source movement is a diverse collection of ideas, knowledge, techniques, and solutions. As a result, it is far from clear how these approaches should be applied to commercial software engineering. This paper has looked at many of the dimensions of the open-source movement, and provided an analysis of the different opportunities available to commercial software vendors. References and Notes 1. It can be argued that the open-source community has produced really only two essential products -- Apache (undeniably the most popular web server) and Linux although both are essentially reincarnations of prior systems. Both are also somewhat products of their times: Apache filled a hole in the then emerging Web, at a time no platform vendor really knew how to step in, and Linux filled a hole in the fragmented Unix market, colored by the community's general anger against Microsoft. 2. Evans Marketing Services, Linux Developers Survey, Volume 1, March 2000.

This work provides a comprehensive overview of research and practical issues relating to component-based development information systems (CBIS). Spanning the organizational, developmental, and technical aspects of the subject, the original research included here provides fresh insights into successful CBIS technology and application. Part I covers component-based development methodologies and system architectures. Part II analyzes different aspects of managing component-based development. Part III investigates component-based development versus commercial off-the-shelf products (COTS), including the selection and trading of COTS products. This book constitutes the proceedings of the 5th International Conference on Web Information Systems Engineering, WISE 2004, held in Brisbane, Australia in November 2004. The 45 revised full papers and 29 revised short papers presented together with 3 invited contributions were carefully reviewed and selected from 198 submissions. The papers are organized in topical sections on Web information modeling; payment and security; information extraction; advanced applications; performance issues; linkage

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

analysis and document clustering; Web caching and content analysis; XML query processing; Web search and personalization; workflow management and enterprise information systems; business processes; deep Web and dynamic content; Web information systems design; ontologies and applicatoins; multimedia, user interfaces, and languages; and peer-to-peer and grid systems.

This book constitutes the refereed proceedings of the Third International Conference on Generative and Component-Based Software Engineering, GCSE 2001, held in Erfurt, Germany, in September 2001. The 14 revised full papers presented together with one invited paper were carefully reviewed and selected from 43 submissions. The papers are organized in topical sections on software product lines, aspects, generic and generative approaches, and components and architectures.

This book focuses on a specialized branch of the vast domain of software engineering: component-based software engineering (CBSE). *Component-Based Software Engineering: Methods and Metrics* enhances the basic understanding of components by defining categories, characteristics, repository, interaction, complexity, and composition. It divides the research domain of CBSE into three major sub-domains: (1) reusability issues, (2) interaction and integration issues, and (3) testing and reliability issues. This book covers the state-of-the-art literature survey of at least 20 years in the domain of reusability, interaction and integration complexities, and testing and reliability issues of component-based software engineering. The aim of this book is not only to review and analyze the previous works conducted by eminent researchers, academicians, and organizations in the context of CBSE, but also suggests innovative, efficient, and better solutions. A rigorous and critical survey of traditional and advanced paradigms of software engineering is provided in the book. Features: In-interactions and Out-Interactions both are covered to assess the complexity. In the context of CBSE both white-box and black-box testing methods and their metrics are described. This work covers reliability estimation using reusability which is an innovative method. Case studies and real-life software examples are used to explore the problems and their solutions. Students, research scholars, software developers, and software designers or individuals interested in software engineering, especially in component-based software engineering, can refer to this book to understand the concepts from scratch. These measures and metrics can be used to estimate the software before the actual coding commences.

This book constitutes the thoroughly refereed post-proceedings of the 5th International Workshop on Product-Family Engineering, PFE 2003, held in Siena, Italy in November 2003. The 36 revised full papers presented together with an introductory overview and 3 keynote presentations were carefully selected during two rounds of reviewing and improvement. The papers are organized in topical sections on variation mechanisms, requirements analysis and management, product derivation, transition to family development, industrial experience, evolution, and decision and derivation.

Following the tradition of previous editions of the MODELS conference, many satellite events were organized in co-location with the MODELS conference in Toulouse in 2008: 12 workshops, 3 symposia, 9 tutorials, a poster session, and a tools exhibition. The selection of the workshops was organized by a Workshop Selection Committee, which consisted of the following experts: – Michel R. V. Chaudron, Leiden University, The Netherlands (Chair) – Jochen Kuster , IBM Research Zurich, Switzerland – Henry

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

Muccini, University of L'Aquila, Italy – Holger Giese, Hasso-Plattner-Institute, Germany – Hans Vangheluwe, McGill University, Canada Some workshops have been running for several years as MODELS satellite events, but each year some workshops end. Furthermore, there are always new developments, and hence there is room for new workshops. Therefore, the Workshop Selection Committee very much welcomes new proposals. The workshops enabled groups of participants to exchange recent and/or preliminary results, to conduct intensive discussions, or to coordinate efforts between representatives of a technical community. They served as forums for lively discussion of innovative ideas, recent progress, or practical experience on model-driven engineering for specific aspects, specific problems, or domain-specific needs. The three symposia this year were: the Doctoral Symposium, the Educators' Symposium, and the Research Projects Symposium. The Doctoral Symposium provided specific support for PhD students to discuss their work and receive guidance for the completion of their dissertation research.

A Product Line is a set of products with common elements and variable features. Including Product Lines in an overall development strategy tailored to the commercial and/or industrial context delivers significant benefits: products that are more suitable, reduction in cost, shorter development timescales, quality improvement, etc. This work, *Systems Product Line Engineering*, brings together a summary of the state-of-the-art with lessons learnt from industrial experience in implementing Product Lines of various kinds, in terms of marketplace, number of applications, degree of variability, etc. It is resolutely practical, and is intended to complement existing Systems Engineering manuals; indeed, it adopts the same process structures. It includes: • Definitions and examples: Product Line, Product Lines organizations, Product Line Engineering, • Processes, from needs analysis through to disposal, • Systems Engineering methods, particularly Model-Based Product Line Systems Engineering, • Organization: development in silos, development in platforms, • Implementation strategies and management processes. This work is intended for practitioners: engineers, project managers, instructors, researchers, students and developers of systems that fit into this approach. Elected IncoSE Product of the Year 2015.

"Reports on the recent advances in UML and XML based software evolution in terms of a wider range of techniques and applications"--Provided by publisher.

Software product lines are emerging as a critical new paradigm for software development. Product lines are enabling organizations to achieve impressive time-to-market gains and cost reductions. With the increasing number of product lines and product-line researchers and practitioners, the time is right for a comprehensive examination of the issues surrounding the software product line approach. The Software Engineering Institute at Carnegie Mellon University is proud to sponsor the first conference on this important subject. This book comprises the proceedings of the First Software Product Line Conference (SPLC1), held August 28-31, 2000, in Denver, Colorado, USA. The twenty-seven papers of the conference technical program present research results and experience reports that cover all aspects of software product lines. Topics include business issues, enabling technologies, organizational issues, and life-cycle issues. Emphasis is placed on experiences in the development and fielding of product lines of complex systems, especially those that expose problems in the design, development, or evolution of software product lines. The book will be essential reading for researchers and practitioners alike.

This book constitutes the thoroughly refereed post-proceedings of the international conference NetObjectDays 2002, held in Erfurt, Germany, in October 2002. The 26 revised full papers presented were carefully selected during two rounds of reviewing and revision. The papers are organized in topical sections on embedded and distributed systems; components and MDA;

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

Java technology; Web services; aspect-oriented software design; agents and mobility; software product lines; synchronization; testing, refactoring, and CASE tools.

Component-based software development, CBSD, is no longer just one more new paradigm in software engineering, but is effectively used in development and practice. So far, however, most of the efforts from the software engineering community have concentrated on the functional aspects of CBSD, leaving aside the treatment of the quality issues and extra-functional properties of software components and component-based systems. The 16 revised chapters presented were carefully reviewed and selected for inclusion in the book; together with an introductory survey, they give a coherent and competent survey of the state of the art in the area. The book - the first to focus on quality issues of components and component-based systems - is organized in topical parts on COTS selection, testing and certification, software component quality models, formal models to quality assessment, and CBSD management.

Annotation The instruction put forth in this new book is all related to successfully using Select Perspective, a process conceived and marketed by Select Business solutions, a division of Aonix. Select Perspective is a pragmatic, component-based software development process that can be implemented by all roles in software development, and includes the business people that specify, accept, verify and use software solutions. Every individual who is involved in the specification, acceptance, construction, testing, delivery or budgetary control of software solutions will benefit from this book. The authors have helped organizations realize the benefit of component-based development with Select Perspective, and this book shows how it can be done, taking into account varying team sizes, uneven skill levels, and different industries. The book uses the UML for expression of designs, and will allow the reader to meet the demands of web services.

Over the last decade, software product line engineering (SPLE) has emerged as one of the most promising software development paradigms for increasing productivity in IT-related industries. Detailing the various aspects of SPLE implementation in different domains, Applied Software Product Line Engineering documents best practices with regard to system development. Expert contributors from academia and industry come together and focus on core asset development, product development, and management, addressing the process, technical, and organizational issues needed to meet the growing demand for information. They detail the adoption and diffusion of SPLE as a primary software development paradigm and also address technical and managerial issues in software product line engineering. Providing an authoritative perspective of the latest research and practice in SPLE, the text: Presents in-depth discussions and many industry / case studies Covers applications in various domains including automotive, business process management, and defense Organized according to the organizational, process, and technical aspects of software product lines within an organization Provides the expertise of a distinguished panel of global contributors Ever-increasing global competition coupled with a fragile world economy means that the pressure is on for software engineers and software process improvement professionals to find ways to meet the needs of expanding markets—with greater efficiency and effectiveness. This book arms readers with the insight needed to harness the power of SPLE to increase productivity, reduce time to market, and to handle the growing diversity in the quickly evolving global marketplace.

This book covers research into the most important practices in product line organization. Contributors offer experience-based knowledge on the domain and application engineering, the modeling and management of variability, and the design and use of tools to support the management of product line-related knowledge.

The book describes a method for developing the testing of components in parallel with their functionality based on models. UML models are used to derive the testing architecture for an application, the testing interfaces and the component testers. The

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

method provides a process and guidelines for modeling and developing these artifacts. The book also discusses the implications of built-in contract testing with other component-based development technologies such as product-line engineering, middleware platforms, reuse principles etc. Still further, it describes a new method for specifying and checking real-time properties of object-oriented, component-based real-time systems that are based on dynamic execution time analysis with optimization algorithms.

Software product lines represent perhaps the most exciting paradigm shift in software development since the advent of high-level programming languages. Nowhere else in software engineering have we seen such breathtaking improvements in cost, quality, time to market, and developer productivity, often registering in the order-of-magnitude range. Here, the authors combine academic research results with real-world industrial experiences, thus presenting a broad view on product line engineering so that both managers and technical specialists will benefit from exposure to this work. They capture the wealth of knowledge that eight companies have gathered during the introduction of the software product line engineering approach in their daily practice. Component-based software development (CBD) is an emerging discipline that promises to take software engineering into a new era. Building on the achievements of object-oriented software construction, CBD aims to deliver software engineering from a cottage industry into an industrial age for Information Technology, wherein software can be assembled from components, in the manner that hardware systems are currently constructed from kits of parts. This volume provides a survey of the current state of CBD, as reflected by activities that have been taking place recently under the banner of CBD, with a view to giving pointers to future trends. The contributions report case studies - self-contained, fixed-term investigations with a finite set of clearly defined objectives and measurable outcomes - on a sample of the myriad aspects of CBD. The book includes chapters dealing with COTS (commercial off-the-shelf) components; methodologies for CBD; compositionality, i.e. how to calculate or predict properties of a composite from those of its constituents; component software testing; and grid computing.

Software product lines are emerging as an important new paradigm for software development. Product lines are enabling organizations to achieve impressive time-to-market gains and cost reductions. In 1997, we at the Software Engineering Institute (SEI) launched a Product Line Practice Initiative. Our vision was that product line development would be a low-risk, high-return proposition for the entire software engineering community. It was our hope from the beginning that there would eventually be sufficient interest to hold a conference. The First Software Product Line Conference (SPLC1) was the realization of that hope. Since SPLC1, we have seen a growing interest in software product lines. Companies are launching their own software product line initiatives, product line technical and business practices are maturing, product line tool vendors are emerging, and books on product lines are being published. Motivated by the enthusiastic response to SPLC1 and the increasing number of software product lines and product line researchers and practitioners, the SEI is proud to sponsor this second conference dedicated to software product lines. We were gratified by the submissions to SPLC2 from all parts of the globe, from government and commercial organizations. From these submissions we were able to assemble a rich and varied

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

conference program with unique opportunities for software product line novices, experts, and those in between. This collection represents the papers selected from that response and includes research and experience reports.

This book constitutes the refereed proceedings of the 9th International Conference on Software Reuse, ICSR 2006, held in Torino, Italy, in June 2006. The book presents 27 revised full papers and 13 revised short papers, carefully reviewed and selected from numerous submissions. The Coverage includes COTS selection and integration; product lines, domain analysis, and variability; reengineering maintenance; programming languages and retrieval; aspect-oriented software development; approaches and models; and components.

This book constitutes the thoroughly refereed post-proceedings of the Second International Symposium on Generative and Component-Based Software Engineering, GCSE 2000, held in Erfurt, Germany in October 2000. The twelve revised full papers presented with two invited keynote papers were carefully reviewed and selected from 29 submissions. The book offers topical sections on aspects and patterns, models and paradigms, components and architectures, and Mixin-based composition and metaprogramming.

**Annotation** This work provides a comprehensive overview of research and practical issues relating to component-based information systems (CBIS). Spanning the organizational, developmental, and technical aspects of the subject, the original research included here provides fresh insights into successful CBIS technology and application, including the selection and trading of commercial off-the shelf products (COTS).

This book constitutes the refereed proceedings of the 5th European Conference on Software Architecture, ECSA 2011, held in Essen, Germany, in September 2011. The 13 revised full papers presented together with 24 emerging research papers, and 7 research challenge poster papers were carefully reviewed and selected from over 100 submissions. The papers are organized in topical sections on requirements and software architectures; software architecture, components, and compositions; quality attributes and software architectures; software product line architectures; architectural models, patterns and styles; short papers; process and management of architectural decisions; software architecture run-time aspects; ADLs and metamodels; and services and software architectures.

Software product line engineering has proven to be the methodology for developing a diversity of software products and software intensive systems at lower costs, in shorter time, and with higher quality. In this book, Pohl and his co-authors present a framework for software product line engineering which they have developed based on their academic as well as industrial experience gained in projects over the last eight years. They do not only detail the technical aspect of the development, but also an integrated view of the business, organisation and process aspects are given. In addition, they explicitly point out the key differences of software product line engineering compared to traditional single software system development, as the need for two distinct development processes for domain and application engineering respectively, or the need to define and manage variability.

Model-Driven Software Development (MDS) is currently a highly regarded development paradigm among developers and researchers. With the advent of OMG's MDA and Microsoft's Software Factories, the MDS approach has moved to the centre of the programmer's attention, becoming the focus of conferences such as OOPSLA, JAOO and OOP. MDS is about using

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

domain-specific languages to create models that express application structure or behaviour in an efficient and domain-specific way. These models are subsequently transformed into executable code by a sequence of model transformations. This practical guide for software architects and developers is peppered with practical examples and extensive case studies. International experts deliver:

- \* A comprehensive overview of MDSD and how it relates to industry standards such as MDA and Software Factories.
- \* Technical details on meta modeling, DSL construction, model-to-model and model-to-code transformations, and software architecture.
- \* Invaluable insight into the software development process, plus engineering issues such as versioning, testing and product line engineering.
- \* Essential management knowledge covering economic and organizational topics, from a global perspective. Get started and benefit from some practical support along the way!

This volume constitutes the refereed proceedings of the 14th International Software Product Line Conference, SPLC 2010, held on Jeju Island, South Korea, in September 2010.

This volume constitutes the refereed proceedings of nine international workshops, EI2N+NSF ICE, ICSP, INBAST, ISDE, MONET, ORM, SeDeS, SWWS, and VADER 2011, held as part of OTM 2011 in Hersonissos on the island of Crete, Greece, in October 2011. The 64 revised full papers presented were carefully reviewed and selected from a total of 104 submissions. The volume also includes three papers from the On the Move Academy (OTMA) 2011 and five ODBASE 2011 poster papers. Topics of the workshop papers are enterprise integration and semantics, information centric engineering, interoperability, industrial and business applications of semantic Web applications, information systems in distributed environments, process management in distributed information system development, distributed information systems: implementation issues and applications, industrial applications of fact-oriented modeling, data warehouse modeling, extensions to fact-oriented modeling, model validation procedures, schema transformations and mapping, semantic Web and Web semantics, ontology development, deployment and interoperability, data access and efficient computation, efficient information processing, exchange and knowledge synthesis algorithms, mobile and networking technologies for social applications, semantic and decision support, variability in software architecture, and dynamic and adaptive architectures.

"Designing Software Product Lines with UML is well-written, informative, and addresses a very important topic. It is a valuable contribution to the literature in this area, and offers practical guidance for software architects and engineers." --Alan Brown Distinguished Engineer, Rational Software, IBM Software Group "Gomaa's process and UML extensions allow development teams to focus on feature-oriented development and provide a basis for improving the level of reuse across multiple software development efforts. This book will be valuable to any software development professional who needs to manage across projects and wants to focus on creating software that is consistent, reusable, and modular in nature."

--Jeffrey S Hammond Group Marketing Manager, Rational Software, IBM Software Group "This book brings together a good range of concepts for understanding software product lines and provides an organized method for developing product lines using object-oriented techniques with the UML. Once again, Hassan has done an excellent job in balancing the needs of both experienced and novice software engineers." --Robert G. Pettit IV, Ph.D. Adjunct Professor of Software Engineering, George Mason University "This breakthrough book provides a comprehensive step-by-step approach on how to develop software product lines, which is of great strategic benefit to industry. The development of software product lines enables significant reuse of software architectures. Practitioners will benefit from the well-defined PLUS process and rich case studies." --Hurley V. Blankenship II Program Manager, Justice and Public Safety, Science Applications International Corporation "The Product Line UML based Software engineering (PLUS) is leading edge. With the author's wide experience and deep knowledge, PLUS is well harmonized with architectural and design pattern technologies."

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

--Michael Shin Assistant Professor, Texas Tech University Long a standard practice in traditional manufacturing, the concept of product lines is quickly earning recognition in the software industry. A software product line is a family of systems that shares a common set of core technical assets with preplanned extensions and variations to address the needs of specific customers or market segments. When skillfully implemented, a product line strategy can yield enormous gains in productivity, quality, and time-to-market. Studies indicate that if three or more systems with a degree of common functionality are to be developed, a product-line approach is significantly more cost-effective. To model and design families of systems, the analysis and design concepts for single product systems need to be extended to support product lines. Designing Software Product Lines with UML shows how to employ the latest version of the industry-standard Unified Modeling Language (UML 2.0) to reuse software requirements and architectures rather than starting the development of each new system from scratch. Through real-world case studies, the book illustrates the fundamental concepts and technologies used in the design and implementation of software product lines. This book describes a new UML-based software design method for product lines called PLUS (Product Line UML-based Software engineering). PLUS provides a set of concepts and techniques to extend UML-based design methods and processes for single systems in a new dimension to address software product lines. Using PLUS, the objective is to explicitly model the commonality and variability in a software product line. Hassan Gomaa explores how each of the UML modeling views--use case, static, state machine, and interaction modeling--can be extended to address software product families. He also discusses how software architectural patterns can be used to develop a reusable component-based architecture for a product line and how to express this architecture as a UML platform-independent model that can then be mapped to a platform-specific model. Key topics include: Software product line engineering process, which extends the Unified Development Software Process to address software product lines Use case modeling, including modeling the common and variable functionality of a product line Incorporating feature modeling into UML for modeling common, optional, and alternative product line features Static modeling, including modeling the boundary of the product line and information-intensive entity classes Dynamic modeling, including using interaction modeling to address use-case variability State machines for modeling state-dependent variability Modeling class variability using inheritance and parameterization Software architectural patterns for product lines Component-based distributed design using the new UML 2.0 capability for modeling components, connectors, ports, and provided and required interfaces Detailed case studies giving a step-by-step solution to real-world product line problems Designing Software Product Lines with UML is an invaluable resource for all designers and developers in this growing field. The information, technology, and case studies presented here show how to harness the promise of software product lines and the practicality of the UML to take software design, quality, and efficiency to the next level. An enhanced online index allows readers to quickly and easily search the entire text for specific topics. This book contains the proceedings of the 5th International Workshop on Product Family Engineering, PFE-5. This workshop was held in Siena, Italy, November 4–6, 2003. This workshop was the fifth in the series, with the same subject, software product family engineering. These workshops have been held initially irregularly about every 18 months since 1996. Since 1999 the workshop has been held every second year in the fall. The proceedings of the second, third and fourth workshops were published as Springer LNCS volumes 1429, 1951 and 2290. The workshops were organized within co-operation projects of European industry. The first two were organized by ARES (Esprit IV 20.477) 1995–1999; this project had 3 industrial and 3 academic partners, and studied software architectures for product families. Some of the partners continued in the ITEA project if99005 ESAPS (1999–2001). ITEA is the software development programme (?! 2023) within the European Eureka initiative. ITEA

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

projects last for 2 years, and ESAPS was succeeded by CAFÉ (ITEA if00004) for 2001–2003 and FAMILIES (ITEA if02009). This fifth workshop was initially prepared within CAFÉ and the preparation continued in FAMILIES. As usual Henk Obbink was the workshop chair, and Linda Northrop and Sergio Bandinelli were the co-chairs.

The size, complexity, and integration level of software systems is increasing constantly. Companies in all domains identify that software defines the competitive edge of their products. These developments require us to constantly search for new approaches to increase the productivity and quality of our software development and to decrease the cost of software maintenance. Generative and component-based technologies hold considerable promise with respect to achieving these goals. GCSE 2001 constituted another important step forward and provided a platform for academic and industrial researchers to exchange ideas. These proceedings represent the third conference on generative and component-based software engineering. The conference originated as a special track on generative programming from the Smalltalk and Java in Industry and Education Conference (STJA), organized by the working group “Generative and Component-Based Software Engineering” of the “Gesellschaft für Informatik” FG 2.1.9 “Object-Oriented Software Engineering.” However, the conference has evolved substantially since then, with its own, independent stature, invited speakers, and, most importantly, a stable and growing community. This year’s conference attracted 43 submissions from all over the world, indicating the broad, international interest in the research field. Based on careful review by the program committee, 14 papers were selected for presentation. I would like to thank the members of the program committee, all renowned experts, for their dedication in preparing thorough reviews of the submissions.

The 2009 Symposium on Component-Based Software Engineering (CBSE 2009) was the 12th in a series of successful events that have grown into the main forum for industrial and academic experts to discuss component technology. Component-based software engineering (CBSE) has emerged as the underlying technology for the assembly of flexible software systems. In essence, CBSE is about composing computational building blocks to construct larger building blocks that fulfill client needs. Most software engineers are involved in some form of component-based development. Nonetheless, the implications of CBSE adoption are wide-reaching and its challenges grow in tandem with its uptake, continuing to inspire our scientific speculation. Component-based development necessarily involves elements of software architecture, modular software design, software verification, testing, configuration and deployment. This year’s submissions represent a cross-section of CBSE research that touches upon all these aspects. The theoretical foundations of component specification, composition, analysis, and verification continue to pose research challenges. What exactly constitutes an adequate semantics for communication and composition so that bigger things can be built from smaller things? How can formal approaches facilitate predictable assembly through better analysis? We have grouped the proceedings into two sub-themes that deal with these issues: component models and communication and composition. At the same time, the world is changing.

This book constitutes the refereed proceedings of the 11th International ACM SIGSOFT Symposium on Component-Based Software Engineering, CBSE 2008, held in Karlsruhe, Germany in October 2008. The 20 revised full papers and 3 short papers presented were carefully reviewed and selected from 70 submissions. The papers feature new trends in global software services and distributed systems architectures to push the limits of established and tested component-based methods, tools and platforms. The papers are organized in topical sections on performance engineering; extra-functional properties: security

and energy; formal methods and model checking; verification techniques; run-time infrastructures; methods of design and development; component models. This book constitutes the refereed proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2007, held in Trondheim, Norway. It covers goal-driven requirements engineering (RE), products and product-lines, value-based RE and the value of RE, requirements elicitation, requirements specification, industrial experience of RE, and requirements quality and quality requirements.

Business Component-Based Software Engineering, an edited volume, aims to complement some other reputable books on CBSE, by stressing how components are built for large-scale applications, within dedicated development processes and for easy and direct combination. This book will emphasize these three facets and will offer a complete overview of some recent progresses.

Projects and works explained herein will prompt graduate students, academics, software engineers, project managers and developers to adopt and to apply new component development methods gained from and validated by the authors. The authors of Business Component-Based Software Engineering are academic and professionals, experts in the field, who will introduce the state of the art on CBSE from their shared experience by working on the same projects. Business Component-Based Software Engineering is designed to meet the needs of practitioners and researchers in industry, and graduate-level students in Computer Science and Engineering.

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 believes 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

## Read Free Component Based Product Line Engineering With Uml Component Based Development Series

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.

This book constitutes the thoroughly refereed post-proceedings of the Second International Workshop on Rapid Integration of Software Engineering Techniques, RISE 2005. The book presents 19 revised full papers together with the abstract of a keynote paper. Among the topics addressed are modelling safety case evolution, practical approaches in model mapping, context-aware service composition, techniques for representing product line core assets for automation, formal development of reactive fault-tolerant systems, and more. Industrial development of software systems needs to be guided by recognized engineering principles. Commercial-off-the-shelf (COTS) components enable the systematic and cost-effective reuse of prefabricated tested parts, a characteristic approach of mature engineering disciplines. This reuse necessitates a thorough test of these components to make sure that each works as specified in a real context. Beydeda and Gruhn invited leading researchers in the area of component testing to contribute to this monograph, which covers all related aspects from testing components in a context-independent manner through testing components in the context of a specific system to testing complete systems built from different components. The authors take the viewpoints of both component developers and component users, and their contributions encompass functional requirements such as correctness and functionality compliance as well as non-functional requirements like performance and robustness. Overall this monograph offers researchers, graduate students and advanced professionals a unique and comprehensive overview of the state of the art in testing COTS components and COTS-based systems.

