Reusable Software Components Object Oriented Embedded Systems Programming In C

If you ally habit such a referred **Reusable Software Components Object Oriented Embedded Systems Programming In C** books that will come up with the money for you worth, get the unconditionally best seller from us currently from several preferred authors. If you desire to entertaining books, lots of novels, tale, jokes, and more fictions collections are then launched, from best seller to one of the most current released.

You may not be perplexed to enjoy every book collections Reusable Software Components Object Oriented Embedded Systems Programming In C that we will categorically offer. It is not in relation to the costs. Its not quite what you obsession currently. This Reusable Software Components Object Oriented Embedded Systems Programming In C , as one of the most effective sellers here will utterly be in the midst of the best options to review.

The Codesign of Embedded Systems: A Unified Hardware/Software Representation - Sanjaya Kumar 2012-12-06
Current practice dictates the separation of the hardware and software development paths early in the design cycle. These paths remain independent with very little interaction occurring between them until system integration. In particular, hardware is often specified without fully appreciating the computational requirements of the software. Also, software development does not influence hardware development and does not track changes made during the hardware design phase. Thus, the ability to explore hardware/software tradeoffs is restricted, such as the movement of functionality from the software domain to the hardware domain (and vice-versa) or the modification of the hardware/software interface. As a result, problems that are encountered during system integration may require modification of the software and/or hardware, resulting in potentially significant cost increases and schedule overruns. To address the problems described above, a cooperative design approach, one that utilizes a unified view of hardware and software, is described. This approach is called hardware/software codesign. The Codesign of Embedded Systems develops several fundamental hardware/software codesign concepts and a methodology that supports them. A unified representation, referred to as a decomposition graph, is presented which can be used to describe hardware or software using either functional abstractions or data abstractions. Using a unified representation based on functional abstractions, an abstract hardware/software model has been implemented in a common simulation environment called ADEPT (Advanced Design Environment Prototyping Tool). This model permits early hardware/software evaluation and tradeoff exploration. Techniques have been developed which support the identification of software bottlenecks and the evaluation of design alternatives with respect to multiple metrics. The application of the model is demonstrated on several examples. A unified representation based on data abstractions is also explored. This work leads to investigations regarding the application of object-oriented techniques to hardware design. The Codesign of Embedded Systems: A Unified Hardware/Software Representation describes a novel approach to a topic of immense importance to CAD researchers and designers alike.
Systems, Controls, Embedded Systems, Energy, and Machines - Richard C. Dorf 2016-04-19
In two editions spanning more than a decade, The Electrical Engineering Handbook stands as the definitive reference to the multidisciplinary field of electrical engineering. Our knowledge continues to grow, and so does the Handbook. For the third edition, it has expanded into a set of six books carefully focused on a specialized area or field of study. Each book represents a concise yet definitive collection of key concepts, models, and equations in its respective domain, thoughtfully gathered for convenient access. Systems, Controls, Embedded Systems, Energy, and Machines explores in detail the fields of energy devices, machines, and systems as well as control systems. It provides all of the fundamental concepts needed for thorough, in-depth understanding of each area and devotes special attention to the emerging area of embedded systems. Each article includes defining terms, references, and sources of further information. Encompassing the work of the world's foremost experts in their respective specialties, Systems, Controls, Embedded Systems, Energy, and Machines features the latest developments, the broadest scope of coverage, and new material on human-computer interaction.
Practical Statecharts in C/C++ - Miro Samek 2002-01-07
Downright revolutionary... the title is a major understatement... 'Quantum Programming' may ultimately change the way embedded software is designed.' -- Michael Barr, Editor-in-Chief, Embedded Systems Programming magazine (Click here
Reliable Object-Oriented Software - Ed Seidewitz 1995
This 1998 book presents the underlying principles associated with object-orientation and its practical application.
Component-Based Software Engineering - Ian Gorton 2006-06-22
This is the refereed proceedings of the 9th International Symposium on Component-Based Software Engineering, CBSE 2006, held in Västerås, Sweden in June/July 2006. The 22 revised full papers and 9 revised short papers presented cover issues concerned with the development of software-intensive systems from reusable parts, the development of reusable parts, and system maintenance and improvement by means of component replacement and customization.
Software Design for Six Sigma - Basem S. El-Haik 2011-02-16
This proposal constitutes an algorithm of design applying the design for six sigma thinking, tools, and philosophy to software design. The algorithm will also include conceptual design frameworks, mathematical derivation for Six Sigma capability upfront to enable design teams to disregard concepts that are not capable upfront, learning the software development cycle and saving development costs. The uniqueness of this book lies in bringing all those methodologies under the umbrella of design and provide detailed description about how these methods, QFD, DOE, the robust method, FMEA, Design for X, Axiomatic Design, TRIZ can be utilized to help quality improvement in software development, what kinds of different roles those methods play in various stages of design and how to combine those methods to form a comprehensive strategy, a design algorithm, to tackle any quality issues in the design stage.
Embedded and Real Time System Development: A Software Engineering Perspective - Mohammad Ayoub Khan 2013-11-19
Nowadays embedded and real-time systems contain complex software. The complexity of embedded systems is increasing, and the amount and variety of software in the embedded products are growing. This creates a big challenge for embedded and real-time software development processes and there is a need to develop separate metrics and benchmarks. “Embedded and Real Time System Development: A Software Engineering Perspective: Concepts, Methods and Principles” presents practical as well as conceptual knowledge of the latest tools, techniques and methodologies of embedded software engineering and real-time systems. Each chapter includes an in-depth investigation regarding the actual or potential role of software engineering tools in the context of the embedded system and real-time system. The book presents state-of-the art and future perspectives with industry experts, researchers, and academicians sharing ideas and experiences including surrounding frontier technologies, breakthroughs, innovative solutions and applications. The book is organized into four parts “Embedded Software Development Process”, “Design Patterns and Development Methodology”, “Modelling Framework” and “Performance Analysis, Power Management and Deployment” with altogether 12 chapters. The book is aiming at (i) undergraduate students and postgraduate students conducting research in the areas of embedded software engineering and real-time systems; (ii) researchers at universities and other institutions working in these fields; and (iii) practitioners in the R&D departments of embedded system. It can be used as an advanced reference for a course taught at the postgraduate level in embedded software engineering and real-time systems.
Programming Embedded Systems - Michael Barr 2006-10-11
Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

Making Embedded Systems - Elecia White 2011-10-25
Interested in developing embedded systems? Since they don't tolerate inefficiency, these systems require a disciplined approach to programming. This easy-to-read guide helps you cultivate a host of good development practices, based on classic software design patterns and new patterns unique to embedded programming. Learn how to build system architecture for processors, not operating systems, and discover specific techniques for dealing with hardware difficulties and manufacturing requirements. Written by an expert who's created embedded systems ranging from urban surveillance and DNA scanners to children's toys, this book is ideal for intermediate and experienced programmers, no matter what platform you use. Optimize your system to reduce cost and increase performance Develop an architecture that makes your software robust in resource-constrained environments Explore sensors, motors, and other I/O devices Do more with less: reduce RAM consumption, code space, processor cycles, and power consumption Learn how to update embedded code directly in the processor Discover how to implement complex mathematics on small processors Understand what interviewers look for when you apply for an embedded systems job "Making Embedded Systems is the book for a C programmer who wants to enter the fun (and lucrative) world of embedded systems. It's very well written—entertaining, even—and filled with clear illustrations." —Jack Ganssle, author and embedded system expert.
Embedded Systems Handbook 2-Volume Set - Richard Zurawski 2018-10-08
During the past few years there has been an dramatic upsurge in research and development, implementations of new technologies, and deployments of actual solutions and technologies in the diverse application areas of embedded systems. These areas include automotive electronics, industrial automated systems, and building automation and control. Comprising 48 chapters and the contributions of 74 leading experts from industry and academia, the Embedded Systems Handbook, Second Edition presents a comprehensive view of embedded systems: their design, verification, networking, and applications. The contributors, directly involved in the creation and evolution of the ideas and technologies presented, offer tutorials, research surveys, and technology overviews, exploring new developments, deployments, and trends. To accommodate the tremendous growth in the field, the handbook is now divided into two volumes. New in This Edition: Processors for embedded systems Processor-centric architecture description languages Networked embedded systems in the automotive and industrial automation fields Wireless embedded systems Embedded Systems Design and Verification Volume I of the handbook is divided into three sections. It begins with a brief introduction to embedded systems design and verification. The book then provides a comprehensive overview of embedded processors and various aspects of system-on-chip and FPGA, as well as solutions to design challenges. The final section explores power-aware embedded computing, design issues specific to secure embedded systems, and web services for embedded devices. Networked Embedded Systems Volume II focuses on selected application areas of networked embedded systems. It covers automotive field, industrial automation, building automation, and wireless sensor networks. This volume highlights implementations in fast-evolving areas which have not received proper coverage in other publications. Reflecting the unique functional requirements of different application areas, the contributors discuss inter-node communication aspects in the context of specific applications of networked embedded systems.
Object-oriented Metrics - Brian Henderson-Sellers 1996
Object-oriented (OO) metrics are an integral part of object technology -- at the research level and in commercial software development projects. This book offers theoretical and empirical tips and facts for creating an OO complexity metrics (measurement) program, based on a review of existing research from the last several years. KEY TOPICS: Covers moving through object-oriented concepts as they related to managing the project lifecycle; the framework in which metrics exist; structural complexity metrics for traditional systems; OO product metrics; and current industrial applications. MARKET: For software developers, programmers, and managers.
Virtual Components Design and Reuse - Ralf Seepold 2013-06-29
Design reuse is not just a topic of research but a real industrial necessity in the microelectronic domain and thus driving the competitiveness of relevant areas like for example telecommunication or automotive. Most companies have already dedicated a department or a central unit that transfer design reuse into reality. All main EDA conferences include a track to the topic, and even specific conferences have been established in this area, both in the USA and in Europe. Virtual Components Design and Reuse presents a selection of articles giving a mature and consolidated perspective to design reuse from different points of view. The authors stem from all relevant areas: research and academia, IP providers, EDA vendors and industry. Some classical topics in design reuse, like specification and generation of components, IP retrieval and cataloguing or interface customisation, are revisited and discussed in depth. Moreover, new hot topics are presented, among them IP quality, platform-based reuse, software IP, IP security, business models for design reuse, and major initiatives like the MEDEA EDA Roadmap.
Reusable Software Components - Ted Van Sickle 1997
Helps real-time embedded systems designers combine the development benefits of the widely-used C language and object-oriented techniques not normally associated with C.Introduces object-oriented programming to microcontroller programmers familiar with C. Shows how objects can be written in C, and developed into classes. Presents useful objects and classes for microcontroller programs, including a class that creates instances of an asynchronous serial port. Shows how to implement components to handle timer functions and input capture. Compiles data sheets for all components derived in the book.Programmers working with real-time embedded systems.
Safe and Secure Software Reuse - John Favaro 2013-06-12
This book constitutes the refereed proceedings of the 13th International Conference on Safe and Secure Software Reuse, ICSR 2013, held in Pisa, Italy, in June 2013. The 27 papers (18 full and 9 short papers) presented were carefully reviewed and selected from various submissions. The papers are organized in topical sections on feature modeling and variability analysis; reuse and testing; architecture and reuse; analysis for reuse; reuse and patterns, short papers, emerging ideas and trends.
A Practical Guide to Testing Object-oriented Software - John D. McGregor 2001
David A. Sykes is a member of Wofford College's faculty.
Model-Driven Domain Analysis and Software Development: Architectures and Functions - Osis, Janis 2010-10-31
"This book displays how to effectively map and respond to the real-world challenges and purposes which software must solve, covering domains such as mechatronic, embedded and high risk systems, where failure could cost human lives"--Provided by publisher.
Object-Oriented Modeling - Jean-Michel Bergé 1996-10-31
Object-oriented techniques and languages have been proven to significantly increase engineering efficiency in software development. Many benefits are expected from their introduction into electronic modeling. Among them are better support for model reusability and flexibility, more efficient system modeling, and more possibilities in design space exploration and prototyping. Object-Oriented Modeling explores the latest techniques in object-oriented methods, formalisms and hardware description language extensions. The seven chapters comprising this book provide an overview of the latest object-oriented techniques for designing systems and hardware. Many examples are given in C++, VHDL and real-time programming languages. Object-Oriented Modeling describes further the use of object-oriented techniques in applications such as embedded systems, telecommunications and real-time systems, using the very latest techniques in object-oriented modeling. It is an essential guide to researchers, practitioners and students involved in software, hardware and system design.
Embedded Systems Design - Bruno Bouyssounouse 2005-02-07
Embedded systems now include a very large proportion of the advanced products designed in the world, spanning transport (avionics, space, automotive, trains), electrical and electronic appliances (cameras, toys, televisions, home appliances, audio systems, and cellular phones), process control (energy production and distribution, factory automation and optimization), telecommunications (satellites, mobile phones and telecom networks), and security (e-commerce, smart cards), etc. The extensive and increasing use of embedded systems and their integration in everyday products marks a significant evolution in information science and technology. We expect that within a short timeframe embedded systems will be a part of nearly all equipment designed or manufactured in Europe, the USA, and Asia. There is now a strategic shift in emphasis for embedded systems designers: from simply achieving feasibility, to achieving optimality. Optimal design of embedded systems means targeting a given market segment at the lowest cost and delivery time possible. Optimality implies seamless integration with the physical and electronic environment while respecting real-world constraints such as hard deadlines, reliability, availability, robustness, power consumption,

and cost. In our view, optimality can only be achieved through the emergence of embedded systems as a discipline in its own right.
Component-Based Software Development for Embedded Systems - Colin Atkinson 2005-11-19
Embedded systems are ubiquitous. They appear in cell phones, microwave ovens, refrigerators, consumer electronics, cars, and jets. Some of these embedded s- tems are safety- or security-critical such as in medical equipment, nuclear plants, and X-by-wire control systems in naval, ground and aerospace transportation - hicles. With the continuing shift from hardware to software, embedded systems are increasingly dominated by embedded software. Embedded software is complex. Its engineering inherently involves a mul- disciplinary interplay with the physics of the embedding system or environment. Embedded software also comes in ever larger quantity and diversity. The next generation of premium automobiles will carry around one gigabyte of binary code. The proposed US DDX submarine is e?ectively a ?oating embedded so- ware system, comprising 30 billion lines of code written in over 100 programming languages. Embedded software is expensive. Cost estimates are quoted at around US$15– 30 per line (from commencement to shipping). In the defense realm, costs can range up to $100, while for highly critical applications, such as the Space Shuttle, the cost per line approximates $1,000. In view of the exponential increase in complexity, the projected costs of future embedded software are staggering.
Embedded Systems - Kiyofumi Tanaka 2012-03-02
Nowadays, embedded systems - the computer systems that are embedded in various kinds of devices and play an important role of specific control functions, have permitted various aspects of industry. Therefore, we can hardly discuss our life and society from now onwards without referring to embedded systems. For wide-ranging embedded systems to continue their growth, a number of high-quality fundamental and applied researches are indispensable. This book contains 19 excellent chapters and addresses a wide spectrum of research topics on embedded systems, including basic researches, theoretical studies, and practical work. Embedded systems can be made only after fusing miscellaneous technologies together. Various technologies condensed in this book will be helpful to researchers and engineers around the world.
Object-Oriented Technology. ECOOP'99 Workshop Reader - Ana Moreira 1999-12-22
ECOOP'99 Workshops, Panels, and Posters Lisbon, Portugal, June 14-18, 1999 Proceedings
Generative Programming and Component Engineering - Gabor Karsai 2004-10-12
This book constitutes the refereed proceedings of the Third International Conference on Generative Programming and Component Engineering, GPCE 2004, held in Vancouver, Canada in October 2004. The 25 revised full papers presented together with abstracts of 2 invited talks were carefully reviewed and selected from 75 submissions. The papers are organized in topical sections on aspect-orientation, staged programming, types for meta-programming, meta-programming, model-driven approaches, product lines, and domain-specific languages and generation.
Design Patterns - Erich Gamma 1995
Software -- Software Engineering.
Component-based Software Development - Kung-Kiu Lau 2004
- First book of its kind (case studies in CBD) - Covers different kinds of components - Covers different component models/technologies - Includes a wide scope of CBD topics - Covers both theoretical and practical work - Includes both formal and informal approaches - Provides a snapshot of current concerns and pointers to future trends
Design Methods and Applications for Distributed Embedded Systems - Bernd Kleinjohann 2006-04-11
The IFIP TC-10 Working Conference on Distributed and Parallel Embedded Systems (DIPES 2004) brings together experts from industry and academia to discuss recent developments in this important and growing field in the splendid city of Toulouse, France. The ever decreasing price/performance ratio of microcontrollers makes it economically attractive to replace more and more conventional mechanical or electronic control systems within many products by embedded real-time computer systems. An embedded real-time computer system is always part of a well-specified larger system, which we call an intelligent product. Although most intelligent products start out as stand-alone units, many of them are required to interact with other systems at a later stage. At present, many industries are in the middle of this transition from stand-alone products to networked embedded systems. This transition requires reflection and architecting: The complexity of the evolving distributed artifact can only be controlled, if careful planning and principled design methods replace the - hoc engineering of the first version of many standalone embedded products.
Advances in Information Technologies - Jean-Yves Roger 1998
Created through a student-tested, faculty-approved review process with input from more than 250 students and faculty, GOVT is an engaging and accessible solution to accommodate the diverse learning styles of today's learners at a value-based price. Focusing on the current and historical conflicts and controversies that define America as a nation, GOVT is a streamlined and extremely current text for the American Government course. Its motivating debate theme and appealing modern format speak directly to today's student. A full suite of learning tools--correlated to the text chapter-by-chapter--are available through CourseMate and include an eBook, Chapter In Review cards, videos, simulations, podcasts, and quizzes that allow students to learn and study wherever they are and whenever they have time.
System Level Design Model with Reuse of System IP - Patrizia Cavalloro 2007-05-08
This book addresses system design, providing a framework for assessing and developing system design practices that observe and utilise reuse of system design know-how. The know-how accumulated in the companies represents an intellectual asset, or property ('IP').
Generative and Component-Based Software Engineering - Krzysztof Czarnecki 2003-06-26
In the past two years, the Smalltalk and Java in Industry and Education C- ference (STJA) featured a special track on generative programming, which was organized by the working group \Generative and Component-Based Software Engineering" of the \Gesellschaft fur¨ Informatik" FG 2.1.9 \Object-Oriented Software Engineering." This track covered a wide range of related topics from domain analysis, software system family engineering, and software product - nes, to extendible compilers and active libraries. The talks and keynotes directed towards this new software engineering paradigm received much attention and - terest from the STJA audience. Hence the STJA organizers suggested enlarging this track, making it more visible and open to wider, international participation. This is how the GCSE symposium was born. The rst GCSE symposium attracted 39 submissions from all over the world. This impressive number demonstrates the international interest in generative programming and related elds. After a careful review by the program comm- tee, fteen papers were selected for presentation. We are very grateful to the members of the program committee, all of them renowned experts, for their dedication in preparing thorough reviews of the submissions. Special thanks go to Elke Pulvermuller¨ and Andreas Speck, who proposed and organized a special conference event, the Young Researches Workshop (YRW). This workshop provided a unique opportunity for young scientists and Ph.D.
Construction and Analysis of Safe, Secure, and Interoperable Smart Devices - Gilles Barthe 2006-04-18
This book constitutes the refereed post-proceedings of the Second International Workshop on Construction and Analysis of Safe, Secure, and Interoperable Smart Devices, CASSIS 2005. The 9 revised full papers presented were carefully selected during two rounds of reviewing and improvement from about 30 workshop talks. The papers are organized in topical sections on research trends in smart devices, Web services, virtual machine technology, security, validation and formal methods, proof-carrying code, and embedded devices.
Embedded Systems Handbook - Richard Zurawski 2018-09-03
Considered a standard industry resource, the Embedded Systems Handbook provided researchers and technicians with the authoritative information needed to launch a wealth of diverse applications, including those in automotive electronics, industrial automated systems, and building automation and control. Now a new resource is required to report on current developments and provide a technical reference for those looking to move the field forward yet again. Divided into two volumes to accommodate this growth, the Embedded Systems Handbook, Second Edition presents a comprehensive view on this area of computer engineering with a currently appropriate emphasis on developments in networking and applications. Those experts directly involved in the creation and evolution of the ideas and technologies presented offer tutorials, research surveys, and technology overviews that explore cutting-edge developments and deployments and identify potential trends. This first self-contained volume of the handbook, Embedded Systems Design and Verification, is divided into three sections. It begins with a brief introduction to embedded systems design and verification. It then provides a comprehensive overview of embedded processors and

various aspects of system-on-chip and FPGA, as well as solutions to design challenges. The final section explores power-aware embedded computing, design issues specific to secure embedded systems, and web services for embedded devices. Those interested in taking their work with embedded systems to the network level should complete their study with the second volume: Network Embedded Systems.

Scientific Engineering of Distributed Java Applications - Nicolas Guelfi 2008-01-04

FIDJI 2004 was an international forum for researchers and practitioners int- estedinthe advancesin,andapplicationsof,softwareengineeringfordistributed application development. Concerning the technologies, the workshop focused on “Java-related” technologies. It was an opportunity to present and observe the latest research, results, and ideas in these areas. Allpaperssubmittedtothisworkshopwerereviewedbyatleasttwomembers of the International Program Committee. Acceptance was based primarily on originality and contribution. We selected, for these post-workshop proceedings, 11 papers amongst 22 submitted, a tutorial and two keynotes.

FIDJI2004aimedatpromotingascienti?capproachtosoftwareengineering. The scope of the workshop included the following topics: – design of distributed applications – development methodologies for software and system engineering – UML-based development methodologies – development of reliable and secure distributed systems – component-based development methodologies – dependability support during system life cycle – fault tolerance re?nement, evolution and decomposition – atomicity and exception handling in system development – software architectures, frameworks and design patterns for developing d- tributed systems – integration of formal techniques in the development process – formal analysis and grounding of modelling notation and techniques (e. g. , UML, metamodelling) – supporting the security and dependability requirements of distributed app- cations in the development process – distributed software inspection – refactoring methods – industrial and academic case studies – development and analysis tools The organization of such a workshop represents an important amount of work.

Scientific and Technical Aerospace Reports - 1994

Computer Aided Systems Theory - EUROCAST'99 - Franz Pichler 2007-03-02

Computer Aided Systems Theory (CAST) deals with the task of contributing to the creation and implementation of tools for the support of usual CAD tools for design and simulation by formal mathematical or logical means in modeling.

Naturally,thebasisfortheconstructionandimplementationofCASTsoftwarei s provided by the existing current knowledge in modeling and by the experience of practitioners in engineering design. Systems Theory, as seen from the viewpoint of CAST research and CAST tool development, has the role of providing formal frameworks and related theoretical knowledge for model-construction and model analysis. We purposely do not distinguish sharply between systems theory and CAST and other similar ?elds of research and tool development such as for example in applied numerical analysis or other computational sciences.

TheheredocumentedEUROCASTconferencewhichtookplaceattheVienna University of Technology re?ects current mainstreams in CAST. As in the p- vious conferences new topics, both theoretical and application oriented, have been addressed. The presented papers show that the ?eld is widespread and that new - velopments in computer science and in information technology are the driving forces.

Theeditorswouldliketothanktheauthorsforprovidingtheirmanuscriptsin hardcopyandinelectronicformontime.Thesta?ofSpringer-VerlagHeidelberg gave, as in previous CAST publications, valuable support in editing this volume.

Advances in Intelligent Modelling and Simulation - Joanna Kołodziej 2012-07-11

One of the most challenging issues in today’s large-scale computational modeling and design is to effectively manage the complex distributed environments, such as computational clouds, grids, ad hoc, and P2P networks operating under various types of users with evolving relationships fraught with uncertainties. In this context, the IT resources and services usually belong to different owners (institutions, enterprises, or individuals) and are managed by different administrators. Moreover, uncertainties are presented to the system at hand in various forms of information that are incomplete, imprecise, fragmentary, or overloading, which hinders in the full and precise resolve of the evaluation criteria, subsequencing and selection, and the assignment scores. Intelligent scalable systems enable the flexible routing and charging, advanced user interactions and the aggregation and sharing of geographically-distributed resources in modern large-scale systems. This book presents new ideas, theories, models, technologies, system architectures and implementation of applications in intelligent scalable computing systems. In 15 chapters, several important Artificial Intelligence-based techniques, such as fuzzy logic, neural networks, evolutionary, and memetic algorithms are studied and implemented. All of those technologies have formed the foundation for the intelligent scalable computing that we know of today. We believe that this book will serve as a reference for students, researchers, and industry practitioners working or interested in joining interdisciplinary research in the areas of intelligent decision systems using emergent distributed computing paradigms. It will also allow newcomers (students and researchers alike) to grasp key issues and potential solutions on the selected topics. This book presents new ideas, theories, models, technologies, system architectures and implementation of applications in intelligent scalable computing systems. In 15 chapters, several important Artificial Intelligence-based techniques, such as fuzzy logic, neural networks, evolutionary, and memetic algorithms are studied and implemented. All of those technologies have formed the foundation for the intelligent scalable computing that we know of today. We believe that this book will serve as a reference for students, researchers, and industry practitioners working or interested in joining interdisciplinary research in the areas of intelligent decision systems using emergent distributed computing paradigms. It will also allow newcomers (students and researchers alike) to grasp key issues and potential solutions on the selected topics.

Telecommunications Engineer's Reference Book - Fraidoon Mazda 2014-06-28

Telecommunications Engineer's Reference Book maintains a balance between developments and established technology in telecommunications. This book consists of four parts. Part 1 introduces mathematical techniques that are required for the analysis of telecommunication systems. The physical environment of telecommunications and basic principles such as the teletraffic theory, electromagnetic waves, optics and vision, ionosphere and troposphere, and signals and noise are described in Part 2. Part 3 covers the political and regulatory environment of the telecommunications industry, telecommunication standards, open system interconnect reference model, multiple access techniques, and network management. The last part deliberates telecommunication applications that includes synchronous digital hierarchy, asynchronous transfer mode, integrated services digital network, switching systems, centrex, and call management. This publication is intended for practicing engineers, and as a supplementary text for undergraduate courses in telecommunications.

Embedded Software and Systems - Zhaohui Wu 2005-08-29

Welcome to the post proceedings of the First International Conference on Embedded Software and Systems (ICESS 2004), which was held in Hangzhou, P. R. China, 9–10 December 2004. Embedded Software and Systems technology is of increasing importance for a wide range of industrial areas, such as aerospace, automotive, telecommunication, and manufacturing automation. Embedded technology is playing an increasingly dominant role in modern society. This is a natural outcome of amazingly fast developments in the embedded field. The ICESS 2004 conference brought together researchers and developers from academia, industry, and government to advance the science, engineering, and technology in embedded software and systems development, and provided them with a forum to present and exchange their ideas, results, work in progress, and experience in all areas of embedded systems research and development. The ICESS 2004 conference attracted much more interest than expected. The total number of paper submissions to the main conference and its three workshops, namely, Pervasive Computing, Automobile Electronics and Tele-communication, was almost 400, from nearly 20 countries and regions. All submissions were reviewed by at least three Program or Technical Committee members or external reviewers. It was extremely difficult to make the final decision on paper acceptance because there were so many excellent, foreseeing, and interesting submissions with brilliant ideas.

Software Engineering with Ada - Grady Booch 1994

Provides complete coverage of the Ada language and Ada programming in general by recognized authorities in Ada software engineering. Demonstrates the power and performance of Ada in the management of large-scale object-oriented systems, and shows how to use Ada features such as generics, packages, and tasking.

Software Reuse: Methods, Techniques, and Tools - Jan Bosch 2004-06-25
This book constitutes the refereed proceedings of the 8th International Conference on Software Reuse, ICSR-8, held in Madrid, Spain in July 2004. The 28 revised full papers presented were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on software variability: requirements; testing reusable software; feature modeling; aspect-oriented software development; component and service development; code level reuse; libraries, classification, and retrieval; model-based approaches; transformation and generation; and requirements.

Embedded Software - Thomas A. Henzinger 2003-06-30
With the omnipresence of micro devices in our daily lifes embedded software has gained tremendous importance in both science and industry.This volume contains 34 invited papers from the First International Workshop on Embedded Systems. They present latest research results from different areas of computer science that are traditionally distinct but relevant to embedded software development (such as, for example, component based design, functional programming, real-time Java, resource and storage allocation, verification). Each paper focuses on one topic, showing the inter-relationship and application to the design and implementation of embedded software systems.

Software Engineering with Reusable Components - Johannes Sametinger 2013-04-17
The book provides a clear understanding of what software reuse is, where the problems are, what benefits to expect, the activities, and its different forms. The reader is also given an overview of what sofware components are, different kinds of components and compositions, a taxonomy thereof, and examples of successful component reuse. An introduction to software engineering and software process models is also provided.