# Uml 2 0 In A Nutshell In A Nutshell Oreilly

Gain the skills to effectively plan software applications andsystems using the latest version of UML UML 2 represents a significant update to the UML specification,from providing more robust mechanisms for modeling workflow andactions to making the modeling language more executable. Now in itssecond edition, this bestselling book provides you with all thetools you'll need for effective modeling with UML 2. The authorsget you up to speed by presenting an overview of UML and its mainfeatures. You'll then learn how to apply UML to produce effectivediagrams as you progress through more advanced topics such asuse-case diagrams, classes and their relationships, dynamicdiagrams, system architecture, and extending UML. The authors takeyou through the process of modeling with UML so that you cansuccessfully deliver a software product or information managementsystem. With the help of numerous examples and an extensive case study,this book teaches you how to: * Organize, describe, assess, test, and realize use cases * Gain substantial information about a system by usingclasses * Utilize activity diagrams, state machines, and interactiondiagrams to handle common issues * Extend UML features for specific environment or domains * Use UML as part of a Model Driven Architecture initiative * Apply an

effective process for using UML The CD-ROM contains all of the UML models and Java?TM code for acomplete application, Java?TM 2 Platform, Standard Edition, Version1.4.1, and links to the Web sites for vendors of UML 2 tools.
The extension UMLsec of the Unified Modeling Language for secure systems development is presented in this text. The book is written in a way which keeps the first part accessible to anyone with a basic background on object-oriented systems. The second part covers the mathematical tools needed to use the UMLsec approach to verify UML specifications against security requirements. It can also be used as part of a general course on applying UML or on computer security. A practically relevant example is used throughout the book to demonstrate the presented methods.
Concise and easy-to-understand guidelines and standards for creating UML 2.0 diagrams.
Includes articles in topic areas such as autonomic computing, operating system architectures, and open source software technologies and applications.
We would like to welcome you to the proceedings of the workshops held in c-junction with the 27th International Conference on Conceptual Modeling (ER 2008). While the ER main conference covers a wide spectrum of conceptual modeling research, increasingly complex real-world problems demand new p-

spectives and activeresearchin new applications.The ER workshopsattempt to provideresearchers,students, andindustry professionalswitha forumto present and discuss emerging hot topics related to conceptual modeling. We received 13 excellent proposals for workshops to be held with ER 2008. We accepted the following seven based on peer reviews: 1. The Second International Workshop on Conceptual Modeling for Life S- ences Applications (CMLSA 2008), organized by Yi-Ping Phoebe Chen and Sven Hartmann. 2. The 5th International Workshopon Evolution and Change in Data Mana- ment (ECDM 2008), organized by Fabio Grandi. 3. The 4th International Workshop on Foundations and Practices of UML (FP-UML 2008), organized by Juan Trujillo and Andreas L. Opdahl. 4. The First International Workshop on Modeling Mobile Applications and Services (M2AS 2008), organized by Fernando Ferri, Patrizia Grifoni, and Maria Chiara Caschera. 5. The Second International Workshop on Requirements, Intentions and Goals in Conceptual Modeling (RIGiM 2008), organized by Colette Rolland, C- son Woo, and Camille Salinesi. 6. The Second International Workshop on Semantic and Conceptual Issues in Geographic Information Systems (SeCoGIS 2008), organized by Esteban Zim´ anyi and Christophe Claramunt. 7. The 5th International Workshop on Web Information Systems Modeling (WISM 2008), organized by Flavius Frasincar, Geert-Jan Houben, and

Philippe Thiran.

The complexity of most real-time and embedded systems often exceeds that of other types of systems since, in addition to the usual spectrum of problems inherent in software, they need to deal with the complexities of the physical world. That world—as the proverbial Mr. Murphy tells us—is an unpredictable and often unfriendly place. Consequently, there is a very strong motivation to investigate and apply advanced design methods and technologies that could simplify and improve the reliability of real-time software design and implementation. As a result, from the first versions of UML issued in the mid 1990's, designers of embedded and real-time systems have taken to UML with vigour and enthusiasm. However, the dream of a complete, model-driven design flow from specification through automated, optimised code generation, has been difficult to realise without some key improvements in UML semantics and syntax, specifically targeted to the real-time systems problem. With the enhancements in UML that have been proposed and are near standardisation with UML 2. 0, many of these improvements have been made. In the Spring of 2003, adoption of a formalised UML 2. 0 specification by the members of the Object Management Group (OMG) seems very close. It is therefore very appropriate to review the status of UML as a set of notations for embedded real-time systems - both the

state of the art and best practices achieved up to this time with UML of previous generations - and where the changes embodied in the 2.

This journal is devoted to all facets of aspect-oriented software development (AOSD) techniques in the context of all phases of the software life cycle, from requirements and design to implementation, maintenance and evolution. The focus is on approaches for systematic identification, modularization, representation and composition of crosscutting concerns, i.e., the aspects and evaluation of such approaches and their impact on improving quality attributes of software systems.

Covers UML 2.0.

Uses friendly, easy-to-understand For Dummies style to helpreaders learn to model systems with the latest version of UML, themodeling language used by companies throughout the world to developblueprints for complex computer systems Guides programmers, architects, and business analysts throughapplying UML to design large, complex enterprise applications thatenable scalability, security, and robust execution Illustrates concepts with mini-cases from different businessdomains and provides practical advice and examples Covers critical topics for users of UML, including objectmodeling, case modeling, advanced dynamic and functional modeling,and component and deployment modeling

This comprehensive guide has been fully revised to cover UML 2.0, today's standard method for modelling software systems. Filled with concise information, it's been crafted to help IT professionals read, create, and understand system artefacts expressed using UML. Includes an example-rich tutorial for those who need familiarizing with the system.

This book constitutes the refereed proceedings of the 15 IFIP International Conference on Testing of Communicating Systems, TestCom 2003, held in Sophia Antipolis, France in May 2003. The 19 revised full papers presented together with three invited contributions were carefully reviewed and selected from 53 submissions. The papers are organized in topical section on next generation networks, IP and UMTS; TTCN-3; automata-based test methodology; and test design, tools, and methodology.

A coherent and integrated account of the leading UML 2 semantics work and the practical applications of UML semantics development With contributions from leading experts in the field, the book begins with an introduction to UML and goes on to offer in-depth and up-to-date coverage of: The role of semantics Considerations and rationale for a UML system model Definition of the UML system model UML descriptive semantics Axiomatic semantics of UML class diagrams The object constraint language Axiomatic semantics of state machines

A coalgebraic semantic framework for reasoning about interaction designs Semantics of activity diagrams Verification of UML models State invariants Model transformation specification and verification Additionally, readers are provided with expert guidance on how to resolve semantic problems and a section on applications of UML semantics with model analysis. UML 2 Semantics and Applications is an ideal resource for researchers and tool-builders working in UML, among others. It is also an excellent textbook for postgraduate teaching and research.

* Examples are easy to understand; diagrams aren't overly busy. * Written in user-friendly style author is known for. * Condensed, distilled presentation of the UML Superstructure document will get you up to speed with UML 2.0.

The idea for this conference came from a meeting of the IFIP (International Federation for Information Processing) Technical Committee for Information Systems (TC8) in Guimares, Portugal in June 2005. Our goal is to build an IFIP forum among the different Information Systems Communities of TC8 dealing with the increasing important area of Enterprise Information Systems. In this particular meeting the committee members intensively discussed the innovative and unique characteristics of Enterprise Information Systems as scientific sub-discipline. Hence, in this meeting it was decided by the TC8 members that the IFIP TC8

First International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS 2006) would be held in April 2006 in Vienna, Austria. Dr. Li Xu (USA) and Dr. A Min Tjoa (IFIP TC8) were assigned to propose a concept for this conference in order to establish an IFIP platform for EIS researchers and practitioners in the field to share experience, and discussing opportunities and challenges. We are very pleased therefore to have this conference organised by the help of the Austrian Computer Society (OCG). OCG supports the idea of this conference due to the urgent need of research and dissemination of new techniques in this key area. We received 180 papers from more than 30 countries for CONFENIS and the Program Committee eventually selected xx papers or extended abstracts, making an acceptance rate of xx% of submitted papers. Each paper was thoroughly reviewed by at least two qualified reviewers.

With its clear introduction to the Unified Modeling Language (UML) 2.0, this tutorial offers a solid understanding of each topic, covering foundational concepts of object-orientation and an introduction to each of the UML diagram types.

This book constitutes the refereed proceedings of the 6th International Workshop on Next Generation Information Technologies and Systems, NGITS 2006, held in Kibbutz Shefayim, Israel, July 2006. The book presents 28 revised full papers

and four revised short papers together with three invited papers. Topical sections include information integration, next generation applications, information systems development, security and privacy, semi-structured data, frameworks, models and taxonomies, simulation and incremental computing, and more.

"This book manages to convey the practical use of UML 2 in clear and understandable terms with many examples and guidelines. Even for people not working with the Unified Process, the book is still of great use. UML 2 and the Unified Process, Second Edition is a must-read for every UML 2 beginner and a helpful guide and reference for the experienced practitioner." --Roland Leibundgut, Technical Director, Zuehlke Engineering Ltd. "This book is a good starting point for organizations and individuals who are adopting UP and need to understand how to provide visualization of the different aspects needed to satisfy it. " --Eric Naiburg, Market Manager, Desktop Products, IBM Rational Software

This thoroughly revised edition provides an indispensable and practical guide to the complex process of object-oriented analysis and design using UML 2. It describes how the process of OO analysis and design fits into the software development lifecycle as defined by the Unified Process (UP). UML 2 and the Unified Process contains a wealth of practical, powerful, and useful techniques that you can apply immediately. As you progress through the text, you will learn

OO analysis and design techniques, UML syntax and semantics, and the relevant aspects of the UP. The book provides you with an accurate and succinct summary of both UML and UP from the point of view of the OO analyst and designer. This book provides Chapter roadmaps, detailed diagrams, and margin notes allowing you to focus on your needs Outline summaries for each chapter, making it ideal for revision, and a comprehensive index that can be used as a reference New to this edition: Completely revised and updated for UML 2 syntax Easy to understand explanations of the new UML 2 semantics More real-world examples A new section on the Object Constraint Language (OCL) Introductory material on the OMG's Model Driven Architecture (MDA) The accompanying website provides A complete example of a simple e-commerce system Open source tools for requirements engineering and use case modeling Industrial-strength UML course materials based on the book

UML stands for Unified Modeling Language used for creating object-oriented, meaningful documentation models for any software system present. It provides us a way to develop rich models that describe the working of any software/hardware systems. UML serves a great way of creating professional documentation which is a necessary part of any project development. Here is what is covered in the book - Chapter 1: UML Diagrams: Versions, Types,

History, Tools, Examples What is UML? Why use UML? UML Versions Characteristics of UML Conceptual model UML Diagrams UML Tools Chapter 2: UML Notation Tutorial: Symbol with Examples What is a model? UML Building Blocks Things Relationships Diagrams Chapter 3: UML Relationships with EXAMPLE: Dependency, Generalization, Realization Association Dependency Generalization Realization Composition Aggregation Chapter 4: UML Association vs Aggregation vs Composition with EXAMPLE Association Composition Aggregation Association vs. Aggregation vs. Composition Chapter 5: UML Class Diagram Tutorial with Examples What is Class? What is Class Diagram? Benefits of Class Diagram Essential elements of A UML class diagram Aggregation vs. Composition Abstract Classes Example of UML Class Diagram Chapter 6: What is UML Object Diagram? Tutorial with Example What is a Class Diagram? What is an Object Diagram? How to draw an object diagram? Purpose of an object diagram Applications of Object Diagrams Chapter 7: UML Use Case Diagram: Tutorial with EXAMPLE What is the Use Case Diagram? Why Use-Case diagram? Use-case diagram notations How to draw a use-case diagram? Tips for drawing a use-case diagram Chapter 8: State Machine Diagram: UML Tutorial with EXAMPLE What is a State Machine Diagram? Why State Machine Diagram? Notation and Symbol for State Machine Types of State How to draw a

Statechart diagram? When to use State Diagrams? Chapter 9: UML Activity Diagram: What is, Components, Symbol, EXAMPLE What is an Activity Diagram? Components of Activity Diagram Why use Activity Diagrams? Activity Diagram Notations How to draw an activity diagram? Chapter 10: Interaction, Collaboration, Sequence Diagrams with EXAMPLES What is Interaction diagram? Purpose of an Interaction Diagram Important terminology Types of Interaction diagram and Notations Sequence Diagram What is the Collaboration diagram? Timing diagram Chapter 11: Component Diagram: UML Tutorial with EXAMPLE What is Component Diagram? Component diagram Notations What is a Component? Why use Component Diagram? When to use Component Diagram? Chapter 12: Deployment Diagram: UML Tutorial with EXAMPLE What is Deployment Diagram? Purpose of a deployment diagram Deployment Diagram Symbol and notations What is an artifact? What is a node? How to draw a deployment diagram? Click the BUY button now and download the book now to start learning UML. Learn it fast and learn it well. Pick up your copy today by clicking the BUY NOW button at the top of this page!

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.

Model-Driven Software Development (MDSD) is currently a highlyregarded development paradigm among developers and researchers.With the advent of OMG's MDA and Microsoft's Software Factories,the MDSD approach has moved

to the centre of the programmer'sattention, becoming the focus of conferences such as OOPSLA, JAOOand OOP. MDSD is about using domain-specific languages to create models thatexpress application structure or behaviour in an efficient anddomain-specific way. These models are subsequently transformed intoexecutable code by a sequence of model transformations. This practical guide for software architects and developers ispeppered with practical examples and extensive case studies.International experts deliver: * A comprehensive overview of MDSD and how it relates to industrystandards such as MDA and Software Factories. * Technical details on meta modeling, DSL construction,model-to-model and model-to-code transformations, and softwarearchitecture. * Invaluable insight into the software development process, plusengineering issues such as versioning, testing and product lineengineering. * Essential management knowledge covering economic andorganizational topics, from a global perspective. Get started and benefit from some practical support along the way!

Information Systems Development (ISD) progresses rapidly, continually creating new challenges for the professionals involved. New concepts, approaches and techniques of systems development emerge constantly in this field. Progress in ISD comes from research as well as from practice. This conference will discuss

issues pertaining to information systems development (ISD) in the inter-networked digital economy. Participants will include researchers, both experienced and novice, from industry and academia, as well as students and practitioners. Themes will include methods and approaches for ISD; ISD education; philosophical, ethical, and sociological aspects of ISD; as well as specialized tracks such as: distributed software development, ISD and knowledge management, ISD and electronic business / electronic government, ISD in public sector organizations, IOS.

This book constitutes the refereed proceedings of the 7th International Conference on the Unified Modeling Language, UML 2004, held in Lisbon, Portugal, in October 2004. The 30 revised full papers presented together with summaries on the workshops and tutorials were carefully reviewed and selected from 135 technical paper submissions. The papers are organized in topical sections on metamodeling, aspects, profiles and extensions, OCL, model transformation, verification and model consistency, security, and methodology.

UML2.0??????????

Covering the breadth of a large topic, this book provides a thorough grounding in object-oriented concepts, the software development process, UML and multi-tier technologies. After covering some basic ground work underpinning OO software

projects, the book follows the steps of a typical development project (Requirements Capture - Design - Specification & Test), showing how an abstract problem is taken through to a concrete solution. The book is programming language agnostic - so code is kept to a minimum to avoid detail and deviation into implementation minutiae. A single case study running through the text provides a realistic example showing development from an initial proposal through to a finished system. Key artifacts such as the requirements document and detailed designs are included. For each aspect of the case study, there is an exercise for the reader to produce similar documents for a different system.

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

and manage projects and the UML systems engineering profile.
A practical approach to enhancing quality in software models usingUML Version 2.0 "Despite its increasing usage, many companies are not taking thebest advantage of UML and, occasionally, individuals haveexperienced frustration in applying its standards. Perhaps this isbecause they have not yet read this book!" -From the Foreword by Prof. Brian Henderson-Sellers This book presents a practical checklist approach to enhancing thequality of software models created with the Unified ModelingLanguage (UML) Version 2.0. The foundation for quality is set bythe discussion on the nature and creation of UML models. This isfollowed by a demonstration of how to apply verification andvalidation checks to these models with three foci: syntacticalcorrectness, semantic meaningfulness, and aesthetic symmetry. Thequality work is carried out within three distinct yet relatedmodeling spaces: * Model of problem space (MOPS) * Model of solution space (MOSS) * Model of background space (MOBS) Readers can then choose a specific quality approach according totheir roles in their projects. Verification and validation checks are also organized according tothese three modeling spaces, making it easier for the reader tofocus on the appropriate diagrams and quality checks correspondingto their modeling space. In addition, a major element of thispublication is the Strengths, Weaknesses, Objectives, and Traps(SWOT) analysis. This analysis is performed on each UML diagram,enabling readers to fully comprehend these diagrams, theiradvantages and limitations, and the

way in which they can be usedin practical projects for modeling. A consistent case study of the Lucky Insurance System is providedthroughout the chapters to illustrate the creation of good qualityUML diagrams, followed by application of quality checks to them.With its emphasis on quality in UML-based projects, this book is anessential resource for all quality professionals, including qualityanalysts, process consultants, quality managers, test designers,and testers.

The popular Unified Modeling Language (UML) is both a language and notation developed by the Object Management Group (OMG) used to design and create specifications for software systems. With the recent release of version 2.0 UML, the OMG has started the OMG-Certified UML Professional Program to provide an objective measure of UML knowledge. As a certified UML professional a developer has an important credential to present to employers and clients. Certification also benefits companies looking for skilled UML practitioners by giving them a basis for making hiring and promotion decisions. UML 2 Certification Guide is the only official study guide to passing the new UML exams. This book systematically covers all of the topics covered in the exams, and has been carefully reviewed by the OMG. The book begins by assuming only a basic knowledge of UML and then progresses far enough to allow a reader to pass both the fundamental and the intermediate level exams. Along the way the book also covers topics that are not in introductory books on UML but that are necessary to pass the exams. Tim Weilkiens is considered one of the top ten experts

on UML, and both authors have extensive experience training developers to successfully take the exams. The official certification resource Assumes a basic knowledge of UML so that you can focus immediately on the exams Written by two authors known for their skill as trainers, consultants, and developers Developed systematically to enable you to master all exam topics—without exception Covers the use of UML for applications, as required by the exams, both inside and outside of the realm of software development Includes a practice exam, glossary, list of books, and website information

This title provides a forum where expert insights are presented on the subject of linking three current phenomena: software evolution, UML and XML.

Globe-trotting travelers have long resorted to handy, pocket-size dictionaries as an aid to communicating across the language barrier. Dan Pilone's UML 2.0 Pocket Reference is just such an aid for on-the-go developers who need to converse in the Unified Modeling Language (UML). Use this book to decipher the many UML diagrams you'll encounter on the path to delivering a modern software system. Updated to cover the very latest in UML, you'll find coverage of the following UML 2.0 diagram types: Class diagrams Component diagrams* Sequence diagrams* Communication diagrams* Timing diagrams* Interaction Overview diagrams* Package diagrams* Deployment diagrams* Use case diagrams Composite structure diagrams* Activity diagrams* Statechart diagrams* * New or expanded coverage in this edition Also new in this

edition is coverage of UML's Object Constraint Language (OCL). Using OCL, you can specify more narrowly the functionality described in a given diagram by recording limits that are the result of business rules and other factors. The UML 2.0 Pocket Reference travels well to meetings and fits nicely into your laptop bag. It's near impossible to memorize all aspects of UML, and with this book along, you won't have to.

System developers have used modeling languages for decades to specify, visualize, construct, and document systems. The Unified Modeling Language (UML) is one of those languages. UML makes it possible for team members to collaborate by providing a common language that applies to a multitude of different systems. Essentially, it enables you to communicate solutions in a consistent, tool-supported language.Today, UML has become the standard method for modeling software systems, which means you're probably confronting this rich and expressive language more than ever before. And even though you may not write UML diagrams yourself, you'll still need to interpret diagrams written by others.UML 2.0 in a Nutshell from O'Reilly feels your pain. It's been crafted for professionals like you who must read, create, and understand system artifacts expressed using UML. Furthermore, it's been fully revised to cover version 2.0 of the language.This comprehensive new edition not only provides a quick-reference to all UML 2.0 diagram types, it also explains key concepts in a way that appeals to readers already familiar with UML or object-oriented programming concepts.Topics include: The role and value of UML in projects The object-oriented paradigm and its

relation to the UML An integrated approach to UML diagrams Class and Object, Use Case, Sequence, Collaboration, Statechart, Activity, Component, and Deployment Diagrams Extension Mechanisms The Object Constraint Language (OCL) If you're new to UML, a tutorial with realistic examples has even been included to help you quickly familiarize yourself with the system.

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

This title contains standards and guidelines for creating UML diagrams that are concise and easy to understand.

This new book is the definitive primer for UML, and starts with the foundational concepts of object-orientation in order to provide the proper context for explaining UML.

Provides a collection of authoritative articles from distinguished international researchers in information technology and Web engineering.

UML 2.0 Pocket Reference"O'Reilly Media, Inc."

The acclaimed beginner's book on object technology now presents UML 2.0, Agile Modeling, and the latest in object development techniques.

"Since its original introduction in 1997, the Unified Modeling Language has revolutionized software development. Every integrated software development

environment in the world--open-source, standards-based, and proprietary--now supports UML and, more importantly, the model-driven approach to software development. This makes learning the newest UML standard, UML 2.0, critical for all software developers--and there isn't a better choice than this clear, step-by-step guide to learning the language." --Richard Mark Soley, Chairman and CEO, OMG If you're like most software developers, you're building systems that are increasingly complex. Whether you're creating a desktop application or an enterprise system, complexity is the big hairy monster you must manage. The Unified Modeling Language (UML) helps you manage this complexity. Whether you're looking to use UML as a blueprint language, a sketch tool, or as a programming language, this book will give you the need-to-know information on how to apply UML to your project. While there are plenty of books available that describe UML, Learning UML 2.0 will show you how to use it. Topics covered include: Capturing your system's requirements in your model to help you ensure that your designs meet your users' needs Modeling the parts of your system and their relationships Modeling how the parts of your system work together to meet your system's requirements Modeling how your system moves into the real world, capturing how your system will be deployed Engaging and accessible, this book shows you how to use UML to craft and communicate your project's design. Russ

Miles and Kim Hamilton have written a pragmatic introduction to UML based on hard-earned practice, not theory. Regardless of the software process or methodology you use, this book is the one source you need to get up and running with UML 2.0. Russ Miles is a software engineer for General Dynamics UK, where he works with Java and Distributed Systems, although his passion at the moment is Aspect Orientation and, in particular, AspectJ. Kim Hamilton is a senior software engineer at Northrop Grumman, where she's designed and implemented a variety of systems including web applications and distributed systems, with frequent detours into algorithms development.

This book constitutes the refereed proceedings of the 10th International Conference on Algebraic Methodology and Software Technology, AMAST 2004, held in Stirling, Scotland, UK in July 2004. The 35 revised full papers presented together with abstracts of 5 invited talks and an invited paper were carefully reviewed and selected from 63 submissions. Among the topics covered are all current issues in formal methods related to algebraic approaches to software engineering including abstract data types, process algebras, algebraic specification, model checking, abstraction, refinement, model checking, state machines, rewriting, Kleene algebra, programming logic, etc.