

Modelica Development Tooling For Eclipse

Principles of Object-Oriented Modeling and Simulation with Modelica 3.3
Content Ontology Design Patterns: Qualities, Methods, and Tools
System Design, Modeling, and Simulation Using Ptolemy II
Automotive Embedded Systems Handbook
Simulation, Modeling, and Programming for Autonomous Robots
Cyber Physical Systems. Model-Based Design
Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4
Tools and Methods for Analysis, Debugging, and Performance Improvement of Equation-Based Models
Environmental Modelling, Software and Decision Support
Software Composition
Model-Driven Engineering and Software Development
Information Science and Applications (ICISA) 2016
SysML Distilled
Wearable Electronics Sensors
Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica
Model-based System and Architecture Engineering with the Arcadia Method
Model-Based Engineering of Embedded Systems
Simulation and Modeling Methodologies, Technologies and Applications
Introduction to Physical Modeling with Modelica
Smart Connected Buildings Design Automation
Topological UML Modeling
Model-based Engineering with AADL
Simulation, Modeling, and Programming for Autonomous Robots
Simulation Tools and Techniques
System Analysis and Modeling. Languages, Methods, and Tools for Systems Engineering
Simulation and Modeling Methodologies, Technologies and Applications
Theory and Practice of Model Transformations
Handbook of Research on Embedded Systems Design
Collaborative Design for Embedded Systems
Decision Management Systems
The MATHEMATICA ® Book, Version 4
Modeling and Simulation of Thermal Power Plants with ThermoSysPro
Compiling Natural Semantics
Complex Systems Design & Management
Principles of Object-Oriented Modeling and Simulation with Modelica 2.1
Systems Architecture Modeling with the Arcadia Method
IEC 61131-3: Programming Industrial Automation Systems
Handbook of Knowledge Representation
The Definition of Standard ML
Literate Programming

Principles of Object-Oriented Modeling and Simulation with Modelica 3.3

This book contains all refereed papers that were accepted to the third edition of the « Complex Systems Design & Management » (CSD&M 2012) international conference that took place in Paris (France) from December 12-14, 2012. (Website: <http://www.csdm2012.csdm.fr>) These proceedings cover the most recent trends in the emerging field of complex systems sciences & practices from an industrial and academic perspective, including the main industrial domains (transport, defense & security, electronics, energy & environment, e-services), scientific & technical topics (systems fundamentals, systems architecture & engineering, systems metrics & quality, systemic tools) and system types (transportation systems, embedded systems, software & information systems, systems of systems, artificial ecosystems). The CSD&M 2012 conference is organized under the guidance of the CESAMES non-profit organization (<http://www.cesames.net>).

Content Ontology Design Patterns: Qualities, Methods, and Tools

This book includes extended and revised versions of a set of selected papers from the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2013) which was co-organized by the Reykjavik University (RU) and sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC). SIMULTECH 2013 was held in cooperation with the ACM SIGSIM - Special Interest Group (SIG) on Simulation and Modeling (SIM), Movimento Italiano Modellazione e Simulazione (MIMOS) and AIS Special Interest Group on Modeling and Simulation (AIS SIGMAS) and technically co-sponsored by the Society for Modeling & Simulation International (SCS), Liophant Simulation, Simulation Team and International Federation for Information Processing (IFIP). This proceedings brings together researchers, engineers, applied mathematicians and practitioners working in the advances and applications in the field of system simulation.

System Design, Modeling, and Simulation Using Ptolemy II

One of the most significant challenges in the development of embedded and cyber-physical systems is the gap between the disciplines of software and control engineering. In a marketplace, where rapid innovation is essential, engineers from both disciplines need to be able to explore system designs collaboratively, allocating responsibilities to software and physical elements, and analyzing trade-offs between them. To this end, this book presents a framework that allows the very different kinds of design models – discrete-event (DE) models of software and continuous time (CT) models of the physical environment – to be analyzed and simulated jointly, based on common scenarios. The individual chapters provide introductions to both sides of this co-simulation technology, and give a step-by-step guide to the methodology for designing and analyzing co-models. They are grouped into three parts: Part I introduces the technical basis for collaborative modeling and simulation with the Crescendo technology. Part II continues with different methodological guidelines for creating co-models and analyzing them in different ways using case studies. Part III then delves into more advanced topics and looks into the potential future of this technology in the area of cyber-physical systems. Finally various appendices provide summaries of the VDM and 20-sim technologies, a number of valuable design patterns applicable for co-models, and an acronym list along with indices and references to other literature. By combining descriptions of the underlying theory with records of real engineers' experience in using the framework on a series of case studies the book appeals to scientists and practitioners alike. It is complemented by tools, examples, videos, and other material on www.crescendotool.org. Scientists/researchers and graduate students working in embedded and cyber-physical systems will learn the semantic foundations for collaborative modeling and simulation, as well as the current capabilities and limitations of methods and tools in this field. Practitioners will be able to develop an appreciation of the capabilities of the co-modeling techniques, to assess the benefits of more collaborative approaches to modeling and simulation, and will benefit from the included

guidelines and modeling patterns.

Automotive Embedded Systems Handbook

The Systems Modeling Language (SysML) extends UML with powerful systems engineering capabilities for modeling a wider spectrum of systems and capturing all aspects of a system's design. SysML Distilled is the first clear, concise guide for everyone who wants to start creating effective SysML models. (Drawing on his pioneering experience at Lockheed Martin and NASA, Lenny Delligatti illuminates SysML's core components and provides practical advice to help you create good models and good designs. Delligatti begins with an easy-to-understand overview of Model-Based Systems Engineering (MBSE) and an explanation of how SysML enables effective system specification, analysis, design, optimization, verification, and validation. Next, he shows how to use all nine types of SysML diagrams, even if you have no previous experience with modeling languages. A case study running through the text demonstrates the use of SysML in modeling a complex, real-world sociotechnical system. Modeled after Martin Fowler's classic UML Distilled, Delligatti's indispensable guide quickly teaches you what you need to know to get started and helps you deepen your knowledge incrementally as the need arises. Like SysML itself, the book is method independent and is designed to support whatever processes, procedures, and tools you already use. Coverage Includes Why SysML was created and the business case for using it Quickly putting SysML to practical use What to know before you start a SysML modeling project Essential concepts that apply to all SysML diagrams SysML diagram elements and relationships Diagramming block definitions, internal structures, use cases, activities, interactions, state machines, constraints, requirements, and packages Using allocations to define mappings among elements across a model SysML notation tables, version changes, and sources for more information

Simulation, Modeling, and Programming for Autonomous Robots

Software -- Programming Languages.

Cyber Physical Systems. Model-Based Design

IEC 61131-3 gives a comprehensive introduction to the concepts and languages of the new standard used to program industrial control systems. A summary of the special programming requirements and the corresponding features in the IEC 61131-3 standard make it suitable for students as well as PLC experts. The material is presented in an easy-to-understand form using numerous examples, illustrations, and summary tables. There is also a purchaser's guide and a CD-ROM containing two reduced but functional versions of programming systems.

Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4

Natural Semantics has become a popular tool among programming language researchers for specifying many aspects of programming languages. However, due to the lack of practical tools for implementation, the natural semantics formalism has so far largely been limited to theoretical applications. This book introduces the rational meta-language RML as a practical language for natural semantics specifications. The main part of the work is devoted to the problem of compiling natural semantics, actually RML, into highly efficient code. For this purpose, an effective compilation strategy for RML is developed and implemented in the rml2c compiler. This compiler ultimately produces low-level C code. Benchmarking results show that rml2c-produced code is much faster than code resulting from compilers based on alternative implementation approaches.

Tools and Methods for Analysis, Debugging, and Performance Improvement of Equation-Based Models

Indispensable reference manual, from the creator of Mathematica.

Environmental Modelling, Software and Decision Support

This book highlights a set of selected, revised and extended papers from the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2017), held in Madrid, Spain, on July 26 to 28, 2017. The conference brought together researchers, engineers and practitioners whose work involves methodologies in and applications of modeling and simulation. The papers showcased here represent the very best papers from the Conference, and report on a broad range of new and innovative solutions.

Software Composition

This book presents ARCADIA—a tooled method devoted to systems and architecture engineering, especially for those dealing with strong constraints to be reconciled (cost, performance, safety, security, reuse, consumption, weight). The book describes the detailed reasoning necessary to: understand the real customer need; define and share the product architecture among all engineering stakeholders; early validate its design and justify it; and ease and master integration, validation, verification and qualification (IVVQ). Offers a comprehensive examination of systems engineering, including the use of models to support it Not only yet another book on modeling, but rather a journey in systems engineering, enlightening the use of models to support it. Focuses on solitary modeling tasks while also covering prime collaborations

between engineering stakeholders Examines modeling techniques to capture and share architecture and to early verify it against need and non-functional constraints Addresses subjects not usually covered by model-based system engineering (MBSE) methods, such as co-engineering with specialties, system/sub-system co-engineering, integration verification and validation Features a powerful, dedicated tool (Capella) Covers a range of topics, including an introduction to system engineering issues, an introduction to MBSE, a presentation of the method for beginners and a handy reference manual for advanced users

Model-Driven Engineering and Software Development

3. 8 Problems . . . 66 4 ENABLING REUSE 69 4. 1 Concepts 69 4. 2 Exploiting commonality 70 4. 3 Reusable building blocks 71 4. 4 Allowing replaceable components 75 4. 5 Other replaceable entities 79 4. 6 Limiting flexibility . . . 82 4. 7 Other considerations . . 84 4. 8 Language fundamentals 85 4. 9 Problems 88 5 FUNCTIONS 91 5. 1 Concepts 91 5. 2 Introduction to functions 92 5. 3 An interpolation function 94 5. 4 Multiple return values 96 97 5. 5 Passing records as arguments 5. 6 Using external subroutines 100 5. 7 Language fundamentals 102 5. 8 Problems 110 6 USING ARRAYS 113 6. 1 Concepts 113 6. 2 Planetary motion: Arrays of components . . 113 6. 3 Simple ID heat transfer: Arrays of variables 120 6. 4 Using arrays with chemical systems 132 6. 5 Language fundamentals 143 6. 6 Problems 152 7 HYBRID MODELS 155 7. 1 Concepts 155 7. 2 Modeling digital circuits 155 7. 3 Bouncing ball 162 7. 4 Sensor modeling 166 7. 5 Language fundamentals 178 7. 6 Problems 186 8 EXPLORING NONLINEAR BEHAVIOR 189 8. 1 Concepts . . . 189 8. 2 An ideal diode 189 8. 3 Backlash . . . 193 8. 4 Thermal properties 199 Contents vii 8. 5 Hodgkin-Huxley nerve cell models 203 8. 6 Language fundamentals 206 8. 7 Problems 210 9 MISCELLANEOUS 213 9. 1 Lookup rules 213 9. 2 Annotations . . 225 Part II Effective Modelica 10 MULTI-DOMAIN MODELING 231 10. 1 Concepts 231 231 10. 2 Conveyor system

Information Science and Applications (ICISA) 2016

A Clear Outline of Current Methods for Designing and Implementing Automotive Systems Highlighting requirements, technologies, and business models, the Automotive Embedded Systems Handbook provides a comprehensive overview of existing and future automotive electronic systems. It presents state-of-the-art methodological and technical solutions in the areas of in-vehicle architectures, multipartner development processes, software engineering methods, embedded communications, and safety and dependability assessment. Divided into four parts, the book begins with an introduction to the design constraints of automotive-embedded systems. It also examines AUTOSAR as the emerging de facto standard and looks at how key technologies, such as sensors and wireless networks, will facilitate the conception of partially and fully autonomous vehicles. The next section focuses on networks and protocols, including CAN, LIN, FlexRay, and TTCAN. The

third part explores the design processes of electronic embedded systems, along with new design methodologies, such as the virtual platform. The final section presents validation and verification techniques relating to safety issues. Providing domain-specific solutions to various technical challenges, this handbook serves as a reliable, complete, and well-documented source of information on automotive embedded systems.

SysML Distilled

This book constitutes the refereed proceedings of the 10th International Conference on System Analysis and Modeling, SAM 2018, held in Copenhagen Denmark, in October 2018. The 12 full papers and 2 short papers presented were carefully reviewed and selected from 24 submissions. The papers describe innovations, trends, and experiences in modeling and analysis of complex systems using ITU-T's Specification and Description Language (SDL-2010) and Message Sequence Chart (MSC) notations, as well as related system design languages — including UML, ASN.1, TTCN, SysML and the User Requirements Notation (URN). This year's edition of SAM will be under the theme “Languages, Methods, and Tools for Systems Engineering”, including languages and methods standardized by the ITU-T, and domain-specific languages. Also included are software engineering technologies, such as for requirements engineering, software verification and validation, and automated code generation.

Wearable Electronics Sensors

Scilab and its Scicos block diagram graphical editor, with a special emphasis on modeling and simulation tools. The first part is a detailed Scilab tutorial, and the second is dedicated to modeling and simulation of dynamical systems in Scicos. The concepts are illustrated through numerous examples, and all code used in the book is available to the reader.

Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica

This book constitutes the proceedings of the 8th International Workshop on Design, Modeling, and Evaluation of Cyber Physical Systems, CyPhy 2018 and 14th International Workshop on Embedded and Cyber-Physical Systems Education, WESE 2018, held in conjunction with ESWeek 2018, in Torino, Italy, in October 2018. The 13 full papers presented together with 1 short paper in this volume were carefully reviewed and selected from 18 submissions. The conference presents a wide range of domains including Modeling, simulation, verification, design, cyber-physical systems, embedded systems, real-time systems, safety, and reliability.

Model-based System and Architecture Engineering with the Arcadia Method

Handbook of Knowledge Representation describes the essential foundations of Knowledge Representation, which lies at the core of Artificial Intelligence (AI). The book provides an up-to-date review of twenty-five key topics in knowledge representation, written by the leaders of each field. It includes a tutorial background and cutting-edge developments, as well as applications of Knowledge Representation in a variety of AI systems. This handbook is organized into three parts. Part I deals with general methods in Knowledge Representation and reasoning and covers such topics as classical logic in Knowledge Representation; satisfiability solvers; description logics; constraint programming; conceptual graphs; nonmonotonic reasoning; model-based problem solving; and Bayesian networks. Part II focuses on classes of knowledge and specialized representations, with chapters on temporal representation and reasoning; spatial and physical reasoning; reasoning about knowledge and belief; temporal action logics; and nonmonotonic causal logic. Part III discusses Knowledge Representation in applications such as question answering; the semantic web; automated planning; cognitive robotics; multi-agent systems; and knowledge engineering. This book is an essential resource for graduate students, researchers, and practitioners in knowledge representation and AI. * Make your computer smarter * Handle qualitative and uncertain information * Improve computational tractability to solve your problems easily

Model-Based Engineering of Embedded Systems

Ontologies are formal knowledge models that describe concepts and relationships and enable data integration, information search, and reasoning. Ontology Design Patterns (ODPs) are reusable solutions intended to simplify ontology development and support the use of semantic technologies by ontology engineers. ODPs document and package good modelling practices for reuse, ideally enabling inexperienced ontologists to construct high-quality ontologies. Although ODPs are already used for development, there are still remaining challenges that have not been addressed in the literature. These research gaps include a lack of knowledge about (1) which ODP features are important for ontology engineering, (2) less experienced developers' preferences and barriers for employing ODP tooling, and (3) the suitability of the eXtreme Design (XD) ODP usage methodology in non-academic contexts. This dissertation aims to close these gaps by combining quantitative and qualitative methods, primarily based on five ontology engineering projects involving inexperienced ontologists. A series of ontology engineering workshops and surveys provided data about developer preferences regarding ODP features, ODP usage methodology, and ODP tooling needs. Other data sources are ontologies and ODPs published on the web, which have been studied in detail. To evaluate tooling improvements, experimental approaches provide data from comparison of new tools and techniques against established alternatives. The analysis of the gathered data resulted in a set of measurable quality indicators that cover aspects of ODP documentation, formal representation or axiomatisation, and usage by ontologists. These indicators highlight quality trade-offs: for instance, between ODP Learnability and Reusability, or between Functional Suitability and Performance Efficiency. Furthermore, the results demonstrate a need for ODP tools that support three novel property specialisation strategies, and highlight the preference of inexperienced developers for

template-based ODP instantiation---neither of which are supported in prior tooling. The studies also resulted in improvements to ODP search engines based on ODP-specific attributes. Finally, the analysis shows that XD should include guidance for the developer roles and responsibilities in ontology engineering projects, suggestions on how to reuse existing ontology resources, and approaches for adapting XD to project-specific contexts.

Simulation and Modeling Methodologies, Technologies and Applications

The first complete guide to SAE AADL: written by the standard's author, completely authoritative, and promoted by both SAE and SEI * *Thoroughly explains the new SAE AADL architecture notation for model-based analysis and validation of mission/safety-critical software-reliant systems. *Presents many real-world examples: ideal for self-learning, instruction, and as a working reference. *Addresses a key standard pioneered by Boeing, Lockheed Martin, Rockwell Collins, DOD, FAA, NASA, ESA, JAXA, and many top universities. Embedded, software-reliant systems are increasingly critical in many industries. In response, 30+ organizations have joined SAE (formerly, the Society of Automobile Engineers) to define the Architecture Analysis and Design Language (AADL). This international industry standard will help streamline and improve systems development through state-of-the-art architecture modeling, analysis, and validation. Ideal for both self-learning and classroom instruction, and an excellent reference for implementers, Model-Based Engineering with AADL is the first book on this crucial new standard. It introduces the reader to all aspects of AADL notation as part of an architecture-centric, model-based engineering approach to discover embedded software systems problems earlier in the lifecycle, and thereby solve them more cost-effectively. Co-authored by Peter Feiler, the standard's author and technical lead, this introductory reference and tutorial is packed with real-world examples. Throughout, the authors compare AADL to other modeling notations and approaches, while presenting the language via a complete case study: the development and analysis of a realistic example system through repeated refinement and analysis.

Introduction to Physical Modeling with Modelica

This volume constitutes the refereed post-conference proceedings of the 11th International Conference on Simulation Tools and Techniques, SIMUTools 2019, held in Chengdu, China, in August 2019. The 97 revised full papers were carefully selected from 156 submissions. The papers focus on simulation methods, simulation techniques, simulation software, simulation performance, modeling formalisms, simulation verification and widely used frameworks.

Smart Connected Buildings Design Automation

This book is an illustrative guide for the understanding and implementation of model-based systems and architecture

engineering with the Arcadia method, using Capella, a new open-source solution. More than just another systems modeling tool, Capella is a comprehensive and extensible Eclipse application that has been successfully deployed in a wide variety of industrial contexts. Based on a graphical modeling workbench, it provides systems architects with rich methodological guidance using the Arcadia method and modeling language. Intuitive model editing and advanced viewing capabilities improve modeling quality and productivity, and help engineers focus on the design of the system and its architecture. This book is the first to help readers discover the richness of the Capella solution. Describes the toolset widely deployed on operational projects in all Thales domains worldwide (defense, aerospace, transportation, etc.) Emphasizes the author's pedagogical experience on the methods and the tools gained through conducting more than 80 training sessions for a thousand engineers at Thales University Examines the emergence of an ecosystem of organizations, including industries that would drive the Capella roadmap according to operational needs, service and technology suppliers who would develop their business around the solution, and academics who would pave the future of the engineering ecosystem

Topological UML Modeling

Topological UML Modeling: An Improved Approach for Domain Modeling and Software Development presents a specification for Topological UML® that combines the formalism of the Topological Functioning Model (TFM) mathematical topology with a specified software analysis and design method. The analysis of problem domain and design of desired solutions within software development processes has a major impact on the achieved result – developed software. While there are many tools and different techniques to create detailed specifications of the solution, the proper analysis of problem domain functioning is ignored or covered insufficiently. The design of object-oriented software has been led for many years by the Unified Modeling Language (UML®), an approved industry standard modeling notation for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system, and this comprehensive book shines new light on the many advances in the field. Presents an approach to formally define, analyze, and verify functionality of existing processes and desired processes to track incomplete or incorrect functional requirements Describes the path from functional and nonfunctional requirements specification to software design with step-by-step creation and transformation of diagrams and models with very early capturing of security requirements for software systems. Defines all modeling constructs as extensions to UML®, thus creating a new UML® profile which can be implemented in existing UML® modeling tools and toolsets

Model-based Engineering with AADL

Model transformations are the glue that tie modelling activities together. If you've used modelling in anger then, whether

you know it or not, you've used model transformations. They come in all shapes and sizes from moving models between different tools to generating implementations. Model transformations have humble beginnings—at one point, not long ago, it was said by many 'in the know' that the way forward in model transformations was to use XSLT. That this idea now raises a wry smile shows how far the model transformation community has come in a short time. Where once model transformations were hacked together in a variety of unsuitable languages, we now have a number of powerful, dedicated languages and theories at our disposal. Since 2008, the ICMT conference series has played a huge part in advancing the subject, and this third edition was no different. The theories and languages presented at ICMT have allowed principled model transformations to play an ever greater part in real systems. Of course there is still much more to do: we need our model transformations, languages, and theories to scale further, allow greater expressivity, be more flexible, and aid reusability; and we lack empirically backed studies of model transformations in use. Doubtless you can think of other gaps. Yet, though some real-world challenges lie just beyond our reach, each year sees once-daunting problems conquered. Much of that progress is now driven by ICMT, and this year's edition showed how model transformations are increasingly being used in previously unfamiliar areas.

Simulation, Modeling, and Programming for Autonomous Robots

Component-based software development is the next step after object-oriented programming that promises to reduce complexity and improve reusability. These advantages have also been identified by the industry, and consequently, over the past years, a large number of component-based techniques and processes have been adopted in many of these organizations. A visible result of this is the number of component models that have been developed and standardized. These models define how individual software components interact with each other and simplify the design process of software systems by allowing developers to choose from previously existing components. The development of component models is a first step in the right direction, but there are many challenges that cannot be solved by the development of a new component model alone. Such challenges are the adaptation of components, and their development and verification. Software Composition is the premiere workshop to advance the research in component-based software engineering and its related fields. SC 2005 was the fourth workshop in this series. As in previous years, SC 2005 was organized as an event co-located with the ETAPS conference. This year's program consisted of a keynote on the revival of dynamic languages given by Prof. Oscar Nierstrasz and 13 technical paper presentations (9 full and 4 short papers). The technical papers were carefully selected from a total of 41 submitted papers. Each paper was thoroughly peer reviewed by at least three members of the program committee and consensus on acceptance was achieved by means of an electronic PC discussion. This LNCS volume contains the revised versions of the papers presented at SC 2005.

Simulation Tools and Techniques

This book contains selected papers from the 7th International Conference on Information Science and Applications (ICISA 2016) and provides a snapshot of the latest issues encountered in technical convergence and convergences of security technology. It explores how information science is core to most current research, industrial and commercial activities and consists of contributions covering topics including Ubiquitous Computing, Networks and Information Systems, Multimedia and Visualization, Middleware and Operating Systems, Security and Privacy, Data Mining and Artificial Intelligence, Software Engineering, and Web Technology. The contributions describe the most recent developments in information technology and ideas, applications and problems related to technology convergence, illustrated through case studies, and reviews converging existing security techniques. Through this volume, readers will gain an understanding of the current state-of-the-art information strategies and technologies of convergence security. The intended readers are researchers in academia, industry and other research institutes focusing on information science and technology.

System Analysis and Modeling. Languages, Methods, and Tools for Systems Engineering

This edited book contains invited papers from renowned experts working in the field of Wearable Electronics Sensors. It includes 14 chapters describing recent advancements in the area of Wearable Sensors, Wireless Sensors and Sensor Networks, Protocols, Topologies, Instrumentation architectures, Measurement techniques, Energy harvesting and scavenging, Signal processing, Design and Prototyping. The book will be useful for engineers, scientist and post-graduate students as a reference book for their research on wearable sensors, devices and technologies which is experiencing a period of rapid growth driven by new applications such as heart rate monitors, smart watches, tracking devices and smart glasses.

Simulation and Modeling Methodologies, Technologies and Applications

Provides an introduction to modern object-oriented design principles and applications for the fast-growing area of modeling and simulation Covers the topic of multi-domain system modeling and design with applications that have components from several areas Serves as a reference for the Modelica language as well as a comprehensive overview of application model libraries for a number of application domains

Theory and Practice of Model Transformations

"A very rich book sprinkled with real-life examples as well as battle-tested advice." —Pierre Haren, VP ILOG, IBM "James does a thorough job of explaining Decision Management Systems as enablers of a formidable business transformation."
—Deepak Advani, Vice President, Business Analytics Products and SPSS, IBM Build Systems That Work Actively to Help You

Maximize Growth and Profits Most companies rely on operational systems that are largely passive. But what if you could make your systems active participants in optimizing your business? What if your systems could act intelligently on their own? Learn, not just report? Empower users to take action instead of simply escalating their problems? Evolve without massive IT investments? Decision Management Systems can do all that and more. In this book, the field's leading expert demonstrates how to use them to drive unprecedented levels of business value. James Taylor shows how to integrate operational and analytic technologies to create systems that are more agile, more analytic, and more adaptive. Through actual case studies, you'll learn how to combine technologies such as predictive analytics, optimization, and business rules—improving customer service, reducing fraud, managing risk, increasing agility, and driving growth. Both a practical how-to guide and a framework for planning, Decision Management Systems focuses on mainstream business challenges. Coverage includes Understanding how Decision Management Systems can transform your business Planning your systems “with the decision in mind” Identifying, modeling, and prioritizing the decisions you need to optimize Designing and implementing robust decision services Monitoring your ongoing decision-making and learning how to improve it Proven enablers of effective Decision Management Systems: people, process, and technology Identifying and overcoming obstacles that can derail your Decision Management Systems initiative

Handbook of Research on Embedded Systems Design

This book explains the modelling and simulation of thermal power plants, and introduces readers to the equations needed to model a wide range of industrial energy processes. Also featuring a wealth of illustrative, real-world examples, it covers all types of power plants, including nuclear, fossil-fuel, solar and biomass. The book is based on the authors' expertise and experience in the theory of power plant modelling and simulation, developed over many years of service with EDF. In more than forty examples, they demonstrate the component elements involved in a broad range of energy production systems, with detailed test cases for each chemical, thermodynamic and thermo-hydraulic model. Each of the test cases includes the following information: • component description and parameterization data; • modelling hypotheses and simulation results; • fundamental equations and correlations, with their validity domains; • model validation, and in some cases, experimental validation; and • single-phase flow and two-phase flow modelling equations, which cover all water and steam phases. A practical volume that is intended for a broad readership, from students and researchers, to professional engineers, this book offers the ideal handbook for the modelling and simulation of thermal power plants. It is also a valuable aid in understanding the physical and chemical phenomena that govern the operation of power plants and energy processes.

Collaborative Design for Embedded Systems

As real-time and integrated systems become increasingly sophisticated, issues related to development life cycles, non-

recurring engineering costs, and poor synergy between development teams will arise. The Handbook of Research on Embedded Systems Design provides insights from the computer science community on integrated systems research projects taking place in the European region. This premier references work takes a look at the diverse range of design principles covered by these projects, from specification at high abstraction levels using standards such as UML and related profiles to intermediate design phases. This work will be invaluable to designers of embedded software, academicians, students, practitioners, professionals, and researchers working in the computer science industry.

Decision Management Systems

Equation-based object-oriented (EEO) modeling languages such as Modelica provide a convenient, declarative method for describing models of cyber-physical systems. Because of the ease of use of EEO languages, large and complex models can be built with limited effort. However, current state-of-the-art tools do not provide the user with enough information when errors appear or simulation results are wrong. It is of paramount importance that such tools should give the user enough information to correct errors or understand where the problems that lead to wrong simulation results are located. However, understanding the model translation process of an EEO compiler is a daunting task that not only requires knowledge of the numerical algorithms that the tool executes during simulation, but also the complex symbolic transformations being performed. As part of this work, methods have been developed and explored where the EEO tool, an enhanced Modelica compiler, records the transformations during the translation process in order to provide better diagnostics, explanations, and analysis. This information is used to generate better error-messages during translation. It is also used to provide better debugging for a simulation that produces unexpected results or where numerical methods fail. Meeting deadlines is particularly important for real-time applications. It is usually essential to identify possible bottlenecks and either simplify the model or give hints to the compiler that enable it to generate faster code. When profiling and measuring execution times of parts of the model the recorded information can also be used to find out why a particular system model executes slowly. Combined with debugging information, it is possible to find out why this system of equations is slow to solve, which helps understanding what can be done to simplify the model. A tool with a graphical user interface has been developed to make debugging and performance profiling easier. Both debugging and profiling have been combined into a single view so that performance metrics are mapped to equations, which are mapped to debugging information. The algorithmic part of Modelica was extended with meta-modeling constructs (MetaModelica) for language modeling. In this context a quite general approach to debugging and compilation from (extended) Modelica to C code was developed. That makes it possible to use the same executable format for simulation executables as for compiler bootstrapping when the compiler written in MetaModelica compiles itself. Finally, a method and tool prototype suitable for speeding up simulations has been developed. It works by partitioning the model at appropriate places and compiling a simulation executable for a suitable parallel platform.

The MATHEMATICA ® Book, Version 4

Master modeling and simulation using Modelica, the new powerful, highly versatile object-based modeling language. Modelica, the new object-based software/hardware modeling language that is quickly gaining popularity around the world, offers an almost universal approach to high-level computational modeling and simulation. It handles a broad range of application domains, for example mechanics, electrical systems, control, and thermodynamics, and facilitates general notation as well as powerful abstractions and efficient implementations. Using the versatile Modelica language and its associated technology, this text presents an object-oriented, component-based approach that makes it possible for readers to quickly master the basics of computer-supported equation-based object-oriented (EEO) mathematical modeling and simulation. Throughout the text, Modelica is used to illustrate the various aspects of modeling and simulation. At the same time, a number of key concepts underlying the Modelica language are explained with the use of modeling and simulation examples. This book:

- Examines basic concepts such as systems, models, and simulations
- Guides readers through the Modelica language with the aid of several step-by-step examples
- Introduces the Modelica class concept and its use in graphical and textual modeling
- Explores modeling methodology for continuous, discrete, and hybrid systems
- Presents an overview of the Modelica Standard Library and key Modelica model libraries

Readers will find plenty of examples of models that simulated distinct application domains as well as examples that combine several domains. All the examples and exercises in the text are available via DrModelica. This electronic self-teaching program, freely available on the text's companion website, guides readers from simple, introductory examples and exercises to more advanced ones. Written by the Director of the Open Source Modelica Consortium, *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica* is recommended for engineers and students interested in computer-aided design, modeling, simulation, and analysis of technical and natural systems. By building on basic concepts, the text is ideal for students who want to learn modeling, simulation, and object orientation.

Modeling and Simulation of Thermal Power Plants with ThermoSysPro

The complex and multidisciplinary nature of environmental problems requires that they are dealt with in an integrated manner. Modeling and software have become key instruments used to promote sustainability and improve environmental decision processes, especially through systematic integration of various knowledge and data and their ability to foster learning and help make predictions. This book presents the current state-of-the-art in environmental modeling and software and identifies the future challenges in the field. State-of-the-art in environmental modeling and software theory and practice for integrated assessment and management serves as a starting point for researchers. Identifies the areas of research and practice required for advancing the requisite knowledge base and tools, and their wider usage. Best practices of environmental modeling enables the reader to select appropriate software and gives the reader tools to integrate natural

system dynamics with human dimensions

Compiling Natural Semantics

Fritzson covers the Modelica language in impressive depth from the basic concepts such as cyber-physical, equation-based, object-oriented, system, model, and simulation, while also incorporating over a hundred exercises and their solutions for a tutorial, easy-to-read experience. The only book with complete Modelica 3.3 coverage Over one hundred exercises and solutions Examines basic concepts such as cyber-physical, equation-based, object-oriented, system, model, and simulation

Complex Systems Design & Management

An in-depth overview of research activities in the field of smart buildings modeling, design automation and control. It will be of interest to industry professionals and researchers who work in the area of smart buildings, smart cities, and smart grid.

Principles of Object-Oriented Modeling and Simulation with Modelica 2.1

Literate programming is a programming methodology that combines a programming language with a documentation language, making programs more easily maintained than programs written only in a high-level language. A literate programmer is an essayist who writes programs for humans to understand. When programs are written in the recommended style they can be transformed into documents by a document compiler and into efficient code by an algebraic compiler. This anthology of essays includes Knuth's early papers on related topics such as structured programming as well as the Computer Journal article that launched literate programming. Many examples are given, including excerpts from the programs for TeX and METAFONT. The final essay is an example of CWEB, a system for literate programming in C and related languages. Index included.

Systems Architecture Modeling with the Arcadia Method

This book constitutes thoroughly revised and selected papers from the Third International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2015, held in Angers, France, in February 2015. The 25 thoroughly revised and extended papers presented in this volume were carefully reviewed and selected from 94 submissions. They are organized in topical sections named: invited papers; modeling languages, tools and architectures; methodologies, processes and platforms; applications and software development.

IEC 61131-3: Programming Industrial Automation Systems

This book is a definitive introduction to models of computation for the design of complex, heterogeneous systems. It has a particular focus on cyber-physical systems, which integrate computing, networking, and physical dynamics. The book captures more than twenty years of experience in the Ptolemy Project at UC Berkeley, which pioneered many design, modeling, and simulation techniques that are now in widespread use. All of the methods covered in the book are realized in the open source Ptolemy II modeling framework and are available for experimentation through links provided in the book. The book is suitable for engineers, scientists, researchers, and managers who wish to understand the rich possibilities offered by modern modeling techniques. The goal of the book is to equip the reader with a breadth of experience that will help in understanding the role that such techniques can play in design.

Handbook of Knowledge Representation

This book constitutes the refereed proceedings of the 4th International Conference on Simulation, Modeling, and Programming for Autonomous Robots, SIMPAR 2014, held in Bergamo, Italy, in October 2014. The 49 revised full papers presented were carefully reviewed and selected from 62 submissions. The papers are organized in topical sections on simulation, modeling, programming, architectures, methods and tools, and systems and applications.

The Definition of Standard ML

Embedded systems have long become essential in application areas in which human control is impossible or infeasible. The development of modern embedded systems is becoming increasingly difficult and challenging because of their overall system complexity, their tighter and cross-functional integration, the increasing requirements concerning safety and real-time behavior, and the need to reduce development and operation costs. This book provides a comprehensive overview of the Software Platform Embedded Systems (SPES) modeling framework and demonstrates its applicability in embedded system development in various industry domains such as automation, automotive, avionics, energy, and healthcare. In SPES 2020, twenty-one partners from academia and industry have joined forces in order to develop and evaluate in different industrial domains a modeling framework that reflects the current state of the art in embedded systems engineering. The content of this book is structured in four parts. Part I “Starting Point” discusses the status quo of embedded systems development and model-based engineering, and summarizes the key requirements faced when developing embedded systems in different application domains. Part II “The SPES Modeling Framework” describes the SPES modeling framework. Part III “Application and Evaluation of the SPES Modeling Framework” reports on the validation steps taken to ensure that the framework met the requirements discussed in Part I. Finally, Part IV “Impact of the SPES Modeling

Framework” summarizes the results achieved and provides an outlook on future work. The book is mainly aimed at professionals and practitioners who deal with the development of embedded systems on a daily basis. Researchers in academia and industry may use it as a compendium for the requirements and state-of-the-art solution concepts for embedded systems development.

Literate Programming

Why are the many highly capable autonomous robots that have been promised for novel applications driven by society, industry, and research not available - day despite the tremendous progress in robotics science and systems achieved during the last decades? Unfortunately, steady improvements in specific robot abilities and robot hardware have not been matched by corresponding robot performance in real world environments. This is mainly due to the lack of - vancements in robot software that master the development of robotic systems of ever increasing complexity. In addition, fundamental open problems are still awaiting sound answers while the development of new robotics applications s- fers from the lack of widely used tools, libraries, and algorithms that are designed in a modular and performant manner with standardized interfaces. Simulation environments are playing a major role not only in reducing development time and cost, e. g. , by systematic software- or hardware-in-the-loop testing of robot performance, but also in exploring new types of robots and applications. H- ever, their use may still be regarded with skepticism. Seamless migration of code using robot simulators to real-world systems is still a rare circumstance, due to the complexity of robot, world, sensor, and actuator modeling. These challenges drive the quest for the next generation of methodologies and tools for robot development. The objective of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN) is to o- er a unique forum for these topics and to bring together researchers from academia and industry to identify and solve the key issues necessary to ease the development of increasingly complex robot software.

[ROMANCE](#) [ACTION & ADVENTURE](#) [MYSTERY & THRILLER](#) [BIOGRAPHIES & HISTORY](#) [CHILDREN'S](#) [YOUNG ADULT](#) [FANTASY](#)
[HISTORICAL FICTION](#) [HORROR](#) [LITERARY FICTION](#) [NON-FICTION](#) [SCIENCE FICTION](#)