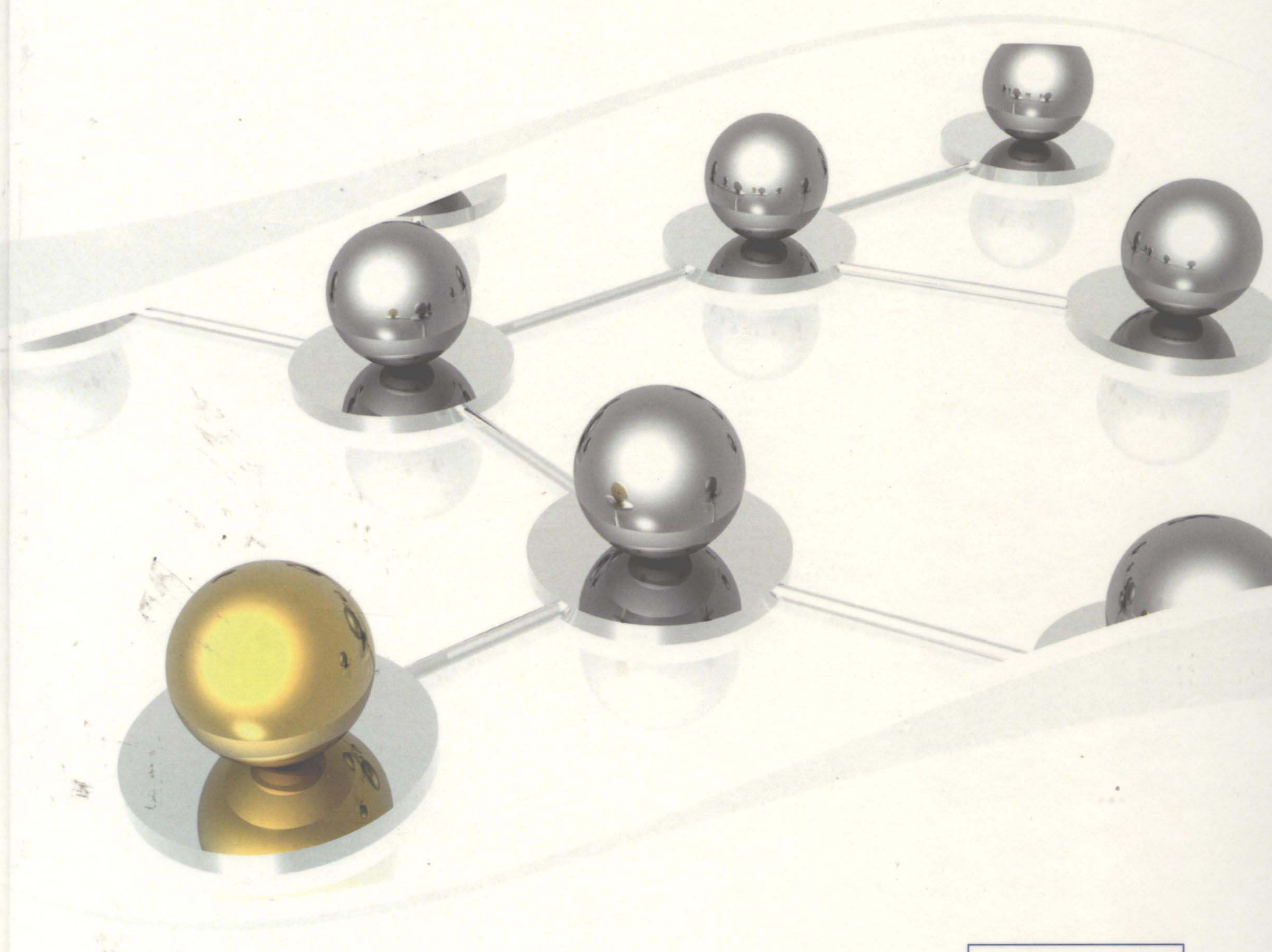
Premier Reference Source

Theory and Application of Multi-Formalism Modeling

Part of the Advances in Systems Analysis, Software Engineering, and High Performance Computing Series



IGI GLOBAL
DISSEMINATOR OF KNOWLEDGE

Marco Gribaudo and Mauro Iacono

Theory and Application of Multi-Formalism Modeling

Marco Gribaudo
Politecnico di Milano, Italy

Mauro Iacono
Seconda Università degli Studi di Napoli, Italy

A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

Information Science REFERENCE
An Imprint of IGI Global

Managing Director: Lindsay Johnston
Production Manager: Jennifer Yoder
Publishing Systems Analyst: Adrienne Freeland
Development Editor: Christine Smith
Acquisitions Editor: Kayla Wolfe
Typesetter: Christina Barkanic
Cover Design: Jason Mull

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Theory and application of multi-formalism modeling / Marco Gribaudo and Mauro Iacono, editors.
pages cm
Includes bibliographical references and index.
ISBN 978-1-4666-4659-9 (hardcover) -- ISBN 978-1-4666-4660-5 (ebook) -- ISBN 978-1-4666-4661-2 (print & perpetual access) 1. Mathematical models. I. Gribaudo, Marco, 1972- II. Iacono, Mauro, 1975-
TA342.T478 2014
003--dc23

2013026788

This book is published in the IGI Global book series Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 2327-3453; eISSN: 2327-3461)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

ISSN: 2327-3453
EISSN: 2327-3461

Mission

The theory and practice of computing applications and distributed systems has emerged as one of the key areas of research driving innovations in business, engineering, and science. The fields of software engineering, systems analysis, and high performance computing offer a wide range of applications and solutions in solving computational problems for any modern organization.

The **Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series** brings together research in the areas of distributed computing, systems and software engineering, high performance computing, and service science. This collection of publications is useful for academics, researchers, and practitioners seeking the latest practices and knowledge in this field.

Coverage

- Computer Graphics
- Computer Networking
- Computer System Analysis
- Distributed Cloud Computing
- Enterprise Information Systems
- Metadata and Semantic Web
- Parallel Architectures
- Performance Modeling
- Software Engineering
- Virtual Data Systems

IGI Global is currently accepting manuscripts for publication within this series. To submit a proposal for a volume in this series, please contact our Acquisition Editors at Acquisitions@igi-global.com or visit: http://www.igi-global.com/publish/.

The Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series (ISSN 2327-3453) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-systems-analysis-software-engineering/73689. Postmaster: Send all address changes to above address.

Titles in this Series

For a list of additional titles in this series, please visit: www.igi-global.com

Enabling the New Era of Cloud Computing Data Security, Transfer, and Management
Yushi Shen (Microsoft, USA) Yale Li (Microsoft, USA) Ling Wu (EMC, USA) Shaofeng Liu (Microsoft, USA) and Qian Wen (Endronic Corp, USA)
Information Science Reference • copyright 2014 • 336pp • H/C (ISBN: 9781466648012) • US $195.00 (our price)

Theory and Application of Multi-Formalism Modeling
Marco Gribaudo (Politecnico di Milano, Italy) and Mauro Iacono (Seconda Università degli Studi di Napoli, Italy)
Information Science Reference • copyright 2014 • 319pp • H/C (ISBN: 9781466646599) • US $195.00 (our price)

Communication Infrastructures for Cloud Computing
Hussein T. Mouftah (University of Ottawa, Canada) and Burak Kantarci (University of Ottawa, Canada)
Information Science Reference • copyright 2014 • 583pp • H/C (ISBN: 9781466645226) • US $195.00 (our price)

Organizational, Legal, and Technological Dimensions of Information System Administration
Irene Marie Portela (Polytechnic Institute of Cávado and Ave, Portugal) and Fernando Almeida (Polytechnic Institute of Gaya, Portugal)
Information Science Reference • copyright 2014 • 321pp • H/C (ISBN: 9781466645264) • US $195.00 (our price)

Advances and Applications in Model-Driven Engineering
Vicente García Díaz (University of Oviedo, Spain) Juan Manuel Cueva Lovelle (University of Oviedo, Spain) B. Cristina Pelayo García-Bustelo (University of Oviedo, Spain) and Oscar Sanjuán Martinez (University of Carlos III, Spain)
Information Science Reference • copyright 2014 • 424pp • H/C (ISBN: 9781466644946) • US $195.00 (our price)

Service-Driven Approaches to Architecture and Enterprise Integration
Raja Ramanathan (Independent Researcher, USA) and Kirtana Raja (Independent Researcher, USA)
Information Science Reference • copyright 2013 • 411pp • H/C (ISBN: 9781466641938) • US $195.00 (our price)

Progressions and Innovations in Model-Driven Software Engineering
Vicente García Díaz (University of Oviedo, Spain) Juan Manuel Cueva Lovelle (University of Oviedo, Spain) B. Cristina Pelayo García-Bustelo (University of Oviedo, Spain) and Oscar Sanjuán Martínez (University of Oviedo, Spain)
Engineering Science Reference • copyright 2013 • 388pp • H/C (ISBN: 9781466642171) • US $195.00 (our price)

Editorial Advisory Board

List of Reviewers

Preface

The term "multiformalism models" describes a complex research area that has become a reality through the years. "Models" refers to one of the most important tools used by scientists in many fields, such as physics, system theory, economics, and computer science. As defined by E.D. Lazoswka (1984) in his *Quantitative System Performance* book, "A model is an abstraction of a system: an attempt to distill, from the mass of details that is the system itself, exactly those aspects that are essential to the system's behavior." "Formalism" includes the word "formal"; it describes the importance of having a set of rules, precisely specified and unambiguous, which must be followed to fulfill a task. Although this might seem to be somehow limiting, this "formality" is what allows automatic techniques to be employed. Since the features that can be used are limited and fixed, algorithms can interpret them with the need of any type of intelligence. In our context, formalism refers to the rules that can be used to describe a model.

"Multi" is a Latin word including the concept of multiplicity. This somehow opens new horizons previously shrunk by the formalism; we must adhere to a set of fixed and unambiguous rules, but we can use several different sets of them. Depending on the particular modeling task we are facing, this gives us the opportunity to choose the most suitable tool to obtain the expected result either more precisely or more efficiently.

The focus of this book concerns *multiformalism modeling*, that is tools, techniques, and practical examples of models exploiting several different formalisms to achieve a specific goal, which would not be possible, or simply not convenient, if pursued with a single modeling paradigm.

Before providing a better specification of multiformalism, it is helpful to present the evolution that led to the definition of this paradigm. As soon as new formalisms were studied, new techniques to automatically perform some kind of analysis on the models described according to the proposed formalism were introduced. The next step was to develop a tool to support the use of both the formalism and the proposed analysis techniques.

Tool development is always a long, demanding, and not a rewarding task. A tool, to be successful, requires a good user interface and must support the user in the task of writing a model. It must perform the required analysis with the least hassle, and it must present the results in the form in which they are easiest to read and to interpret. This, however, requires very long programs to be written: they must predict all the possible actions taken by the users and must guide the actions to be coherent with the selected formalism. Worst of all, a tool developer is hardly rewarded for his job; modeling tools are rarely sold (since in most of the cases they are just used by a research center for experimentations and not for production), and even when they are commercially successful, the budget that can be planned for further development and support is usually very limited. From a scientific point of view, the quantity of time spent on tool development can be an order of magnitude greater than the one required to formulate a hypothesis, verify it, and describe it in a paper for a top-class journal. Usually a tool can lead to a minor contribution, such as a poster or a two page description in a track of a specialized conference.

However, tools are important for modeling: they are what can lead the scientific community to choose one particular formalism over another. For example, in the mid 1980s, several different formalisms were being studied to add timing to Petri Nets models. In particular, there were two main lines being followed: adding time to places, or to transitions. For the latter, a tool with one of the first Graphical User Interfaces (GUI), based on X-Windows and Motif, was developed. It was called the *GreatSPN* tool, and it was developed by G. Chiola and others. The tool allowed drawing Generalized Stochastic Petri Nets models (GSPNs) by graphically inserting primitives in a window. The tool had enormous success and gave a big contribution to the adoption of the GSPN formalism where time was associated with Petri Nets Transitions.

Let us return to multiformalism models. Why did tools play an important role in the development of multiformalism? Well, whoever has worked on a tool, despite all the problems previously pointed out, has usually enjoyed the experience, and most likely repeated with another one! Of course, this has deeper reasons than the willingness to pursue difficult results and to sacrifice themselves for the community. In most of the cases, the developers working on tools also work on formalism specification. As the formalism under study evolves, the tools follow accordingly. Sometimes formalisms evolve in something completely different; this leads to the requirement of completely new tools.

When developing a second, third, or n-th tool, one usually finds a very large similarity with the one previously created. Such similarity can reside in the user interface, in the solution algorithms, in the support library created to analyze the models, or in the modules exporting the results. When there is a need to write a tool from scratch, it is easy to realize that creating it to support a larger number of formalisms is usually only a fraction of work harder than developing it formalism-locked on a single specific modeling paradigm.

The earliest examples of multiformalism tools were thus multiformalism in the sense that they could analyze models described in different languages but just one at a time. Although being a good starting point, since the users would have had to learn just one common interface to be able to experiment with different modeling formalisms, their features were still very limited. Once the modeling language was chosen, the user was stuck with it. If a model could be split in several different parts, where each of them could be better expressed using a different language, the modeler would have to choose one and try to use it to model the components of the system where that particular formalism was sub-optimal.

In parallel to tool development, theoretical researchers were trying to understand the expressive power of the various formalisms that were introduced. This was achieved by formal proofs, where models defined in a specific formalism were transformed in equivalent models of another formalism whose properties and expressiveness were already known. Formalism transformations led to a first type of multiformalism modeling: one could create a model using a paradigm, transform it in an equivalent model for a different formalism, and then use it in the target language.

Both the tools and the language transformations approaches led to the idea that allowing a modeler to use more formalisms at the same time can provide large benefits. In particular, it allows one to use the language that is most suited for each particular component of the system, leading to models that are clearer to read, easier to understand, and more efficient to analyze.

Of course, this is not always possible. Sometimes formalisms are so different they simply cannot be integrated in any meaningful way. In other cases, what can be described with a simple connection of elements coming from multiple specification languages hides a large set of details that should be specified to make the model "formal" enough to be analyzed by a tool. These difficulties led the research in multiformalism modeling recently.

Several different approaches for the multiformalism have been introduced; each one has its own target, its own goal, and its specific set of applications where it can work at its best. Some of them are well defined and well known to the whole modeling community. Among them, we must emphasize the *Möbius* tool, by W. H. Sanders, which currently is the state-of-the-art of multiformalism tools. In particular, it allows one to define (sub)-models in different formalisms and to compose them in a larger multiformalism model. Some other approaches are lesser known but interesting for the way in which the problem is approached, solved, and presented.

A concept tightly coupled with the multiformalism model and how multiformalism models can be analyzed is *multisolution*. The term multisolution refers to the ability of integrating different tools to perform different analyses on the considered multiformalism models. It has two main advantages: it allows using the most mature and well performing tools already developed for a particular formalism, and it reduces the time required to develop multiformalism analysis algorithms, since the task usually reduces to the orchestration of existing software components. However, multisolution has several drawbacks. The main one is that it can be applied only to a few very specific (but important) interpretations of multiformalism modeling. The second practical concern is the standardization of results and interchange formats, which must be present to allow the interaction of different tools.

In this book, we provide a picture of the current state of multiformalism modeling and multisolution, continuing a work that was started with the organization of the WRUMMM ("Workshop on Research and Use of Multiformalism Modeling Methods") workshop during the QEST 2012 conference in London, UK. The workshop was held in London on September the 17th 2012, with the aim of gathering researchers, practitioners, and users of whatever is related to multiformalism modeling methods and techniques. It was guided by the idea that multiformalism modeling methods may constitute a powerful tool to improve modeling and model evaluation experiences in every application field. Within the goals of WRUMMM there was the creation and coordination of a community of modelers and theoreticians to develop and spread multiformalism approaches and to allow an easy information and collaboration exchange inside academia and with industrial partners. The original aim of the workshop was to collect original papers related to theoretical and methodological issues as well as case studies and automated tool support in the research areas related to multiformalism modeling, which were identified in:

- Multiformalism theoretical foundations.
- Methodological approaches.
- Modeling techniques.
- Modeling experiences.
- Multisolution approaches.
- Heterogeneous models composition.
- Case studies from different disciplines.
- Tools integration.
- Results presentation and analysis.
- Standard interchange modeling language definitions.

Many of the contributions of this book come from the participants of that conference but many others are from other researchers who decided to describe their vision of the subject specifically for this work. Contributions has been chosen to focus on what we believe to be the emerging directions of multiformalism modeling, leaving the more consolidated approaches to the references that will presented at the end of each chapter.

The book is organized as follows. It is divided in three sections: the first focuses on the theoretical aspects, the second the practical concerns, while the last one presents successful applications of the considered techniques.

The first section, called "Methodology and Problems," presents contributions about the main themes around multiformalism modeling. Its purpose is to introduce the theoretical aspects of the topic and to make a deep review of the related literature. It is composed of six chapters, written by some of the most active researchers in the field.

The 1st chapter is an introduction to multiformalism modeling. The chapter introduces the main themes concerning multiformalism modeling, presenting their advantages and disadvantages, research experiences and directions, existing solutions, and open questions. After an introduction to the basic ideas behind multiformalism modeling, the chapter proposes some informal definitions and classifications. This allows building a common discussion background as well as supporting the analysis and the comparison of the main existing solutions. The chapter also gives a brief review of literature about the most widely spread frameworks, supporting the interested reader in further readings.

The 2nd chapter studies techniques to combine heterogeneity, compositionality, and automatic generation in formal modeling. It mainly deals with the application of model transformation-based approaches to multiformalism modeling. The chapter examines the advantages and the principles of this technique to design consistent modeling frameworks capable of conjugating the benefit model composition, multiformalism, and high-level abstract representations. Effectiveness of the proposal is demonstrated by a case study.

The 3rd chapter traces the directions towards a multiformalism multisolution framework for model-driven performance engineering. It deals with the problem of enabling multisolution of high-level representation-based models. The chapter, exploiting the conceptual tools typical of the Model-Driven Engineering field, proposes a transformation-based approach founded on the automatic generation of models, written in different performance-oriented formalisms, starting from a Performance Model Interchange Format description of a system. The approach is demonstrated by showing how it is possible to generate some example transformations, which are applied in a case study.

Multi-modeling, meta-modeling, and workflow languages are considered in chapter 4. Multi-modeling techniques and an application of workflow languages to the solution of multiformalism models are presented. The chapter proposes a modeling methodology, built on a semantic background, aiming to evaluate metrics over scenarios developed in a given domain. The proposed methodology guides the modeler from an ontological description of a given domain to a dedicated workflow language that implements the analysis. The approach is demonstrated by a case study that considers an application to drug interdiction and intelligence.

The SIMTHESys framework, working on multiformalism modeling and compositionality, is described in chapter 5. The problem of compositionality in multiformalism models is considered in depth by first analyzing the main results of research activities in this field and then by comparing the most well-known frameworks and tools, with special reference to the solutions implemented by the SIMTHESys framework, which is used as an example. For each of the examined approaches, the mechanisms to obtain the interactions between heterogeneous sub-models are reported.

The issues of result representations are considered in the 6^{th} chapter, which focuses on a meta-model based approach to the definition of the analysis results of Petri net models. The chapter shows how a proper application of meta-modeling techniques can enable the creation of a common method to report and reuse the (intermediate) results in a solution process that involves different cooperating solvers. The approach is applied to Petri Net-based models.

The second section, called "Exploiting Multiformalism: Tool Design Experiences," presents design experiences of multiformalism tools. Its aim is to give an idea of the challenges that must be faced when creating a multiformalism tool. This is accomplished by presenting three examples in three different chapters.

Chapter 7 describes a Petri Net-based tool for the analysis of Generalized Continuous Time Bayesian Networks. In particular, it presents a tool that exploits Petri nets to analyze models based on the Generalized Continuous Time Bayesian Network Formalism. The chapter describes the architecture of the tool, together with the considerations that guided its design and the transformation rules applied to obtain the formalism translation. This chapter can serve as a guideline that can be generalized to deal with multiple formalisms. The tool is applied to a case study to show the solution process.

Chapter 8 proposes a multiformalism, multisolution approach to efficient analysis of large-scale population models. This is accomplished by a multiformalism-based tool capable of facing models with large-scale populations. The architecture of the tool is discussed showing how different solvers are integrated. The tool applies a multisolution approach that includes a mean field analysis solution engine. The approach is validated by a case study of a chemical process.

Chapter 9 studies a symbolic approach to the analysis of multiformalism Markov reward models. The chapter deals with the problem of representing a high number of states in state-space-based solvers. The authors show how it is possible to apply an efficient state representation technique to support the solution processes for models of large systems. The proposed method can be used for the implementation of multiformalism-oriented solvers. The technique can produce a solver based on Markov reward models that is capable of supporting composed models in which sub-models interact by shared state variables or synchronization.

Chapter 10 proposes the experiences developed within the SIMTHESys project with several user-defined formalisms, chosen to demonstrate how complex features can be embedded in various design languages to face specific problems. The chapter briefly introduces the framework to offer a complete modeling example and a number of formalisms through which the reader is guided to inspire and stimulate new custom solutions.

The final section of the book presents two interesting case studies. In particular, it focuses on modeling experiences coming from industrial applications.

Chapter 11 considers a unified modeling and operational framework for fault detection, identification, and recovery in autonomous spacecraft. The chapter presents a dependability application in the aerospace sector that exploits multiformalism to provide probabilistic diagnosis and prognosis of the state of an autonomous spacecraft, capable of self-repair. The integration of the chosen formalisms, oriented to assess different characteristics of the system, allows obtaining a unified modeling framework for all the relevant aspects.

Finally, chapter 12 presents a Model-Driven Methodology to Evaluate Performability of Metro Systems. It considers an application of multiformalism modeling in Model-Driven Engineering to the assessment of metropolitan train systems. The chapter proposes a framework to support performability evaluation during design and in-service phases, which exploits automatic generation of the required models.

We hope that this book will help interested readers in having a broader picture of multiformalism modeling. Our goal is to build a community to continue to study this fascinating and open problem and to apply this technique to more and more case studies. We think that the advantages that can be obtained in many fields by using such techniques can reward the difficulties that this topic poses, especially at the beginning of its study.

Marco Gribaudo
Politecnico di Milano, Italy

Mauro Iacono
Seconda Università degli Studi di Napoli, Italy

REFERENCES

Lazowska, E. D., Zahorjan, J., Graham, G., & Sevcik, K. (1984). *Quantative system performance: Computer system analysis using queueing network models*. Upper Saddle River, NJ: Prentice Hall..

Section 1
Methodology and Problems

This section presents contributions about the main themes around multiformalism modeling.

Table of Contents