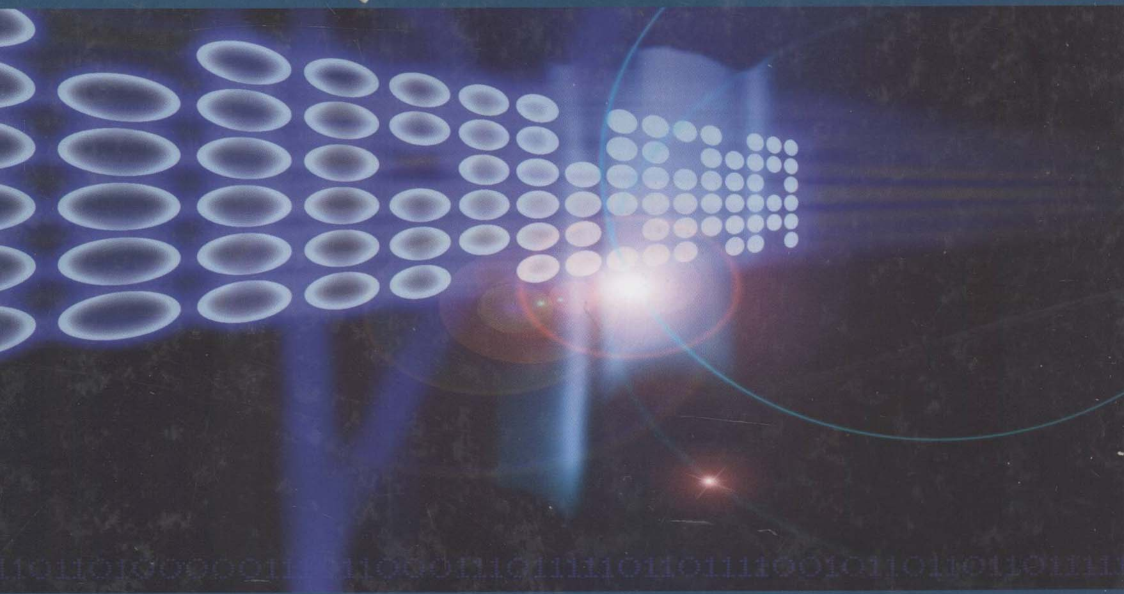

Modeling and Verification of Real-Time Systems

Edited by
Stephan Merz and Nicolas Navet

ISTE

WILEY

Modeling and Verification of Real-Time Systems

Modeling and Verification of Real-Time Systems

Formalisms and Software Tools

Edited by
Stephan Merz
Nicolas Navet

ISTE

WILEY

First published in Great Britain and the United States in 2008 by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
6 Fitzroy Square
London W1T 5DX
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Cataloging-in-Publication Data

Modeling and verification of real-time systems : formalisms and software tools / edited by Nicolas Navet, Stephan Merz.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-1-84821-013-4
1. Real-time data processing. 2. Computer software--Verification. 3. Formal methods (Computer science) I. Navet, Nicolas. II. Merz, Stephan.
QA76.54.M635 2008
004.01'51--dc22

2007045063

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN: 978-1-84821-013-4

Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire.

Contents

Preface

The study of real-time systems has been recognized over the past 30 years as a discipline of its own whose research community is firmly established in academia as well as in industry. This book aims at presenting some fundamental problems, methods, and techniques of this domain, as well as questions open for research.

The field is mainly concerned with the control and analysis of dynamically evolving systems for which requirements of timeliness are paramount. Typical examples include systems for the production or transport of goods, materials, energy or information. Frequently, controllers for these systems are "embedded" in the sense that they are physically implemented within the environment with which they interact, such as a computerized controller in a plane or a car. This characteristic imposes strong constraints on space, cost, and energy consumption, which limits the computational power and the available memory for these devices, in contrast with traditional applications of computer science where resources usually grow exponentially according to Moore's law. The design of real-time systems relies on techniques that originate in several disciplines, including control theory, operations research, software engineering, stochastic process analysis and others.

Software supporting real-time systems needs not only to compute the correct value of a given function, but it must also deliver these values at the right moment in order to ensure the safety and the required performance level of the overall system. Usually, this is implemented by imposing constraints (or *deadlines*) on the termination of certain activities. The verification techniques presented in this volume can help to ensure that deadlines are respected.

Chapter written by Stephan MERZ and Nicolas NAVET.

The chapters of this book present basic concepts and established techniques for modeling real-time systems and for verifying their properties. They concentrate on functional and timing requirements; the analysis of non-functional properties such as schedulability and Quality of Service guarantees would be a useful complement, but would require a separate volume. Formal methods of system design are based on mathematical principles and abstractions; they are a cornerstone for a "zero-default" discipline. However, their use for the development of real-world systems requires the use of efficient support tools. The chapters therefore emphasize the presentation of verification techniques and tools associated with the different specification methods, as well as the presentation of case studies that illustrate the application of the formalisms and the tools. The focus lies on model checking approaches, which attempt to provide a "push-button" approach to verification and integrate well into standard development processes. The main obstacle for the use of model checking in industrial-sized developments is the state-explosion problem, and several chapters describe techniques based on abstraction, reduction or compression that stretch the limits of the size of systems that can be handled.

Before verification can be applied, the system must be modeled in a formal description language such as (timed) Petri nets, timed automata or process algebra. The properties expected of a system are typically expressed in temporal logic or using automata as observers. Two main classes of properties are *safety* properties that, intuitively, express that nothing bad ever happens, and *liveness* properties that assert that something good eventually happens. The third step is the application of the verification algorithm itself to decide whether the properties hold over the model of the system or not; in the latter case, model checking generates a counter-example exhibiting a run of the system that violates the property.

Beyond *verification*, which compares two formal objects, the model should also be *validated* to ensure that it faithfully represents the system under development. One approach to validation is to decide healthiness properties of the model (for example, ensure that each component action can occur in a system run), and model checking is again useful here. In general, it is helpful to narrow the gap between the system description and its formal model, for example by writing a model in a high-level executable language or in a notation familiar to designers such as UML. The chapters of this book, written by researchers active in the fields, present different possible approaches to the problems of modeling, verification and validation, as well as open research questions.

Chapter 1, written by Bernard Berthomieu, Florent Peres and François Vernadat, explains the analysis of real-time systems based on timed Petri nets. It illustrates the high expressiveness of that formalism and the different, complementary verification techniques that are implemented in the Tina tool.