Lecture Notes in Computer Science 6100

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Holger Giese Gabor Karsai Edward Lee
Bernhard Rumpe Bernhard Schätz (Eds.)

Model-Based Engineering of Embedded Real-Time Systems

International Dagstuhl Workshop
Dagstuhl Castle, Germany, November 4-9, 2007
Revised Selected Papers

Springer

Volume Editors

Holger Giese
Hasso-Plattner-Institute
for Software Systems Engineering
Potsdam, Germany
E-mail: holger.giese@hpi.uni-potsdam.de

Gabor Karsai
Vanderbilt University
Nashville, TN, USA
E-mail: gabor.karsai@vanderbilt.edu

Edward Lee
University of California at Berkeley
Berkeley, USA
E-mail: eal@eecs.berkeley.edu

Bernhard Rumpe
RWTH Aachen University
Aachen, Germany
E-mail: rumpe@se-rwth.de

Bernhard Schätz
fortiss GmbH
Garching, Germany
E-mail: schaetz@fortiss.org

Library of Congress Control Number: 2010935675

CR Subject Classification (1998): D.2.9, D.2, D.3.3, C.3-4, F.3

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-642-16276-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-16276-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The topic of "Model-Based Engineering of Real-Time Embedded Systems" brings together a challenging problem domain (real-time embedded systems) and a solution domain (model-based engineering). It is also at the forefront of integrated software and systems engineering, as software in this problem domain is an essential tool for system implementation and integration. Today, real-time embedded software plays a crucial role in most advanced technical systems such as airplanes, mobile phones, and cars, and has become the main driver and facilitator for innovation. Development, evolution, verification, configuration, and maintenance of embedded and distributed software nowadays are often serious challenges as drastic increases in complexity can be observed in practice.

Model-based engineering in general, and model-based software development in particular, advocates the notion of using models throughout the development and life-cycle of an engineered system. Model-based software engineering reinforces this notion by promoting models not only as the tool of abstraction, but also as the tool for verification, implementation, testing, and maintenance. The application of such model-based engineering techniques to embedded real-time systems appears to be a good candidate to tackle some of the problems arising in the problem domain.

Model-based development strategies and model-driven automatic code generation are becoming established technologies on the functional level. However, they are mainly applied within a limited scope only. The use of analogous modeling strategies on the system, technical, and configuration levels remains challenging, especially with the increasing shift to networks of systems, tight coupling between the control-engineering oriented and reactive parts of a system, and the growing number of variants introduced by product lines. Specific domain constraints such as real-time requirements, resource limitations, and hardware-specific dependencies often impede the acceptance of standard high-level modeling techniques and their application. Much effort in industry and academia therefore goes into the adaptation and improvement of object-oriented and component-based methods and model-based engineering that promise to facilitate the development, deployment, and reuse of software components embedded in real-time environments. The model-based development approach for embedded systems and their software proposes application-specific modeling techniques using domain specific concepts (e.g., time-triggered execution or synchronous data flow) to abstract "away" the details of the implementation, such as interrupts or method calls. Furthermore, analytical techniques (e.g., the verification of the completeness of function deployment and consistency of dynamic interface descriptions) and generative techniques (e.g., automatic schedule generation, default behavior generation) can then be applied to the resulting more abstract models to enable the efficient development of high-quality software.

Our Dagstuhl seminar brought together researchers and practitioners from the field of model-based engineering of embedded real-time systems. The topics covered included: frameworks and methods, validation, model-based integration technology, formal modeling of semantics, fault management, concurrency models and models of computation, requirements modeling, formal derivation of designs from requirements, test modeling and model-based test generation, quality assurance, design management, abstractions and extensions, and development techniques and problems of application domains. The broad spectrum of presentations clearly illustrate the prevalence of model-based techniques in the embedded systems area, as well as progress in the field.

This volume is a collection of long and short papers that survey the state of the art in model-based development of real-time embedded systems. It is composed of longer chapters that cover broad areas and short papers that discuss specific tools. The chapters are organized into sections as follows:

- **Foundations:** The chapters in this section survey general models of reactive systems, techniques, and approaches for model-based integration, and modeling and simulation of real-time applications.
- **Language Engineering:** The chapters here review metamodeling as a fundamental tool, the methods for specifying the semantics of models for dynamic behavior, and the requirements for modeling languages for real-time embedded systems.
- **Domain-Specific Issues:** Relevant issues of real-time embedded systems are discussed in this section, including the use of model-based techniques for safety-critical software, and analysis and development approaches to dependable systems.
- **Life-Cycle Issues:** These chapters discuss requirements modeling techniques for embedded systems, and the technology for model evolution and management

The short papers provide a state-of-the-art survey of existing tools that are being used in the model-based engineering of embedded real-time systems.

Finally, we would like to thank all authors and contributors to the project without whom such a large and complex project could not have been completed. It has been made possible by several researchers who supported the organizers and kept things going. In particular, we have to thank Claas Pinkernell, Markus Look, and Sven Bürger for their assistance in compiling this book. Thanks also goes to the Dagstuhl organization staff members who always make our meetings there a unique event.

Holger Giese
Gabor Karsai
Edward Lee
Bernhard Rumpe
Bernhard Schätz

Table of Contents

Part III: Modeling

Part IV: Model Analysis