Model-Based Testing for Embedded Systems
Computational Analysis, Synthesis, and Design of Dynamic Systems Series

Series Editor

Pieter J. Mosterman
MathWorks
Natick, Massachusetts

McGill University
Montréal, Québec

Discrete-Event Modeling and Simulation: A Practitioner’s Approach, Gabriel A. Wainer


Model-Based Design for Embedded Systems, edited by Gabriela Nicolescu and Pieter J. Mosterman


Multi-Agent Systems: Simulation & Applications, edited by Adelinde M. Uhrmacher and Danny Weyns

Forthcoming Titles:

Computation for Humanity: Information Technology to Advance Society, edited by Justyna Zander and Pieter J. Mosterman

Model-Based Testing for Embedded Systems

EDITED BY Justyna Zander, Ina Schieferdecker, and Pieter J. Mosterman
Contents

Preface ix
Editors xi
Contributors xv
Technical Review Committee xix
Book Introduction xxi

Part I Introduction

1 A Taxonomy of Model-Based Testing for Embedded Systems from Multiple Industry Domains 3
   Justyna Zander, Ina Schieferdecker, and Pieter J. Mosterman

2 Behavioral System Models versus Models of Testing Strategies in Functional Test Generation 23
   Antti Huima

3 Test Framework Architectures for Model-Based Embedded System Testing 49
   Stephen P. Masticola and Michael Gall

Part II Automatic Test Generation

4 Automatic Model-Based Test Generation from UML State Machines 77
   Stephan Weißleder and Holger Schlingloff

5 Automated Statistical Testing for Embedded Systems 111
   Jesse H. Poore, Lan Lin, Robert Eschbach, and Thomas Bauer

6 How to Design Extended Finite State Machine Test Models in Java 147
   Mark Utting

7 Automatic Testing of Lustre/SCADE Programs 171
   Virginia Papailiopoulou, Besnik Seljimi, and Ioannis Parissis

8 Test Generation Using Symbolic Animation of Models 195
   Frédéric Dadeau, Fabien Peureux, Bruno Legeard, Régis Tissot, Jacques Julliand, Pierre-Alain Masson, and Fabrice Bouquet
# Part III  Integration and Multilevel Testing

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Model-Based Integration Testing with Communication Sequence Graphs</td>
<td>Fevzi Belli, Axel Hollmann, and Sascha Padberg</td>
</tr>
<tr>
<td>11</td>
<td>Multilevel Testing for Embedded Systems</td>
<td>Abel Marrero Pérez and Stefan Kaiser</td>
</tr>
<tr>
<td>12</td>
<td>Model-Based X-in-the-Loop Testing</td>
<td>Jürgen Großmann, Philip Makedonski, Hans-Werner Wiesbrock, Jaroslav Svacina, Ina Schieferdecker, and Jens Grabowski</td>
</tr>
</tbody>
</table>

# Part IV  Specific Approaches

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>A Survey of Model-Based Software Product Lines Testing</td>
<td>Sebastian Oster, Andreas Wübbeke, Gregor Engels, and Andy Schürr</td>
</tr>
<tr>
<td>14</td>
<td>Model-Based Testing of Hybrid Systems</td>
<td>Thao Dang</td>
</tr>
<tr>
<td>15</td>
<td>Reactive Testing of Nondeterministic Systems by Test Purpose-Directed Tester</td>
<td>Jüri Vain, Andres Kull, Marko Käärmäe, Maili Markvardt, and Kullo Raiend</td>
</tr>
<tr>
<td>16</td>
<td>Model-Based Passive Testing of Safety-Critical Components</td>
<td>Stefan Gruner and Bruce Watson</td>
</tr>
</tbody>
</table>

# Part V  Testing in Industry

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Applying Model-Based Testing in the Telecommunication Domain</td>
<td>Fredrik Abbors, Veli-Matti Aho, Jani Koivulainen, Risto Teittinen, and Dragos Truscan</td>
</tr>
<tr>
<td>18</td>
<td>Model-Based GUI Testing of Smartphone Applications: Case S60™ and Linux®</td>
<td>Antti Jääskeläinen, Tommi Takala, and Mika Katara</td>
</tr>
<tr>
<td>19</td>
<td>Model-Based Testing in Embedded Automotive Systems</td>
<td>Paweł Skruch, Miroslaw Panek, and Bogdan Kowalczyk</td>
</tr>
</tbody>
</table>

# Part VI  Testing at the Lower Levels of Development

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Testing-Based Translation Validation of Generated Code</td>
<td>Mirko Conrad</td>
</tr>
</tbody>
</table>
The ever-growing pervasion of software-intensive systems into technical, business, and social areas not only consistently increases the number of requirements on system functionality and features but also puts forward ever-stricter demands on system quality and reliability. In order to successfully develop such software systems and to remain competitive on top of that, early and continuous consideration and assurance of system quality and reliability are becoming vitally important.

To achieve effective quality assurance, model-based testing has become an essential ingredient that covers a broad spectrum of concepts, including, for example, automatic test generation, test execution, test evaluation, test control, and test management. Model-based testing results in tests that can already be utilized in the early design stages and that contribute to high test coverage, thus providing great value by reducing cost and risk. These observations are a testimony to both the effectiveness and the efficiency of testing that can be derived from model-based approaches with opportunities for better integration of system and test development.

Model-based test activities comprise different methods that are best applied complementing one another in order to scale with respect to the size and conceptual complexity of industry systems. This book presents model-based testing from a number of different perspectives that combine various aspects of embedded systems, embedded software, their models, and their quality assurance. As system integration has become critical to dealing with the complexity of modern systems (and, indeed, systems of systems), with software as the universal integration glue, model-based testing has come to present a persuasive value proposition in system development. This holds, in particular, in the case of heterogeneity such as components and subsystems that are partially developed in software and partially in hardware or that are developed by different vendors with off-the-shelf components.

This book provides a collection of internationally renowned work on current technological achievements that assure the high-quality development of embedded systems. Each chapter contributes to the currently most advanced methods of model-based testing, not in the least because the respective authors excel in their expertise in system verification and validation. Their contributions deliver supreme improvements to current practice both in a qualitative as well as in a quantitative sense, by automation of the various test activities, exploitation of combined model-based testing aspects, integration into model-based design process, and focus on overall usability. We are thrilled and honored by the participation of this select group of experts. They made it a pleasure to compile and edit all of the material, and we sincerely hope that the reader will find the endeavor of intellectual excellence as enjoyable, gratifying, and valuable as we have.

In closing, we would like to express our genuine appreciation and gratitude for all the time and effort that each author has put into his or her chapter. We gladly recognize that the high quality of this book is solely thanks to their common effort, collaboration, and communication. In addition, we would like to acknowledge the volunteer services of those who joined the technical review committee and to extend our genuine appreciation for their involvement. Clearly, none of this would have been possible had it not been for the
continuous support of Nora Konopka and her wonderful team at Taylor & Francis. Many thanks to all of you! Finally, we would like to gratefully acknowledge support by the Alfried Krupp von Bohlen und Halbach Stiftung.

Justyna Zander
Ina Schieferdecker
Pieter J. Mosterman

MATLAB® is a registered trademark of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA
Tel: 508 647 7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com
Editors

Justyna Zander is a postdoctoral research scientist at Harvard University (Harvard Humanitarian Initiative) in Cambridge, Massachusetts (since 2009), and project manager at the Fraunhofer Institute for open communication systems in Berlin, Germany (since 2004).

She holds a PhD (2008) and an MSc (2005), both in the fields of computer science and electrical engineering from Technical University Berlin in Germany, a BSc (2004) in computer science, and a BSc in environmental protection and management from Gdansk University of Technology in Poland (2003).

She graduated from the Singularity University, Mountain View, California, as one of 40 participants selected from 1200 applications in 2009. For her scientific efforts, Dr. Zander received grants and scholarships from institutions such as the Polish Prime Ministry (1999–2000), the Polish Ministry of Education and Sport (2001–2004), which is awarded to 0.04% students in Poland, the German Academic Exchange Service (2002), the European Union (2003–2004), the Hertie Foundation (2004–2005), IFIP TC6 (2005), IEEE (2006), Siemens (2007), Metodos y Tecnologia (2008), Singularity University (2009), and Fraunhofer Gesellschaft (2009–2010). Her doctoral thesis on model-based testing was supported by the German National Academic Foundation with a grant awarded to 0.31% students in Germany (2005–2008).
Ina Schieferdecker studied mathematical computer science at Humboldt-University Berlin and earned her PhD in 1994 at Technical University Berlin on performance-extended specifications and analysis of quality-of-service characteristics. Since 1997, she has headed the Competence Center for Testing, Interoperability and Performance (TIP) at the Fraunhofer Institute on Open Communication Systems (FOKUS), Berlin, and now heads the Competence Center Modelling and Testing for System and Service Solutions (MOTION).

She has been a professor on engineering and testing of telecommunication systems at Technical university Berlin since 2003.

Professor Schieferdecker has worked since 1994 in the area of design, analysis, testing, and evaluation of communication systems using specification-based techniques such as unified modeling language, message sequence charts, and testing and test control notation (TTCN-3). Professor Schieferdecker has written many scientific publications in the area of system development and testing. She is involved as editorial board member with the International Journal on Software Tools for Technology Transfer. She is a cofounder of the Testing Technologies IST GmbH, Berlin, and a member of the German Testing Board. In 2004, she received the Alfried Krupp von Bohlen und Halbach Award for Young Professors, and she became a member of the German Academy of Technical Sciences in 2009. Her work on this book was partially supported by the Alfried Krupp von Bohlen und Halbach Stiftung.
Pieter J. Mosterman is a senior research scientist at MathWorks in Natick, Massachusetts, where he works on core Simulink® simulation and code generation technologies, and he is an adjunct professor at the School of Computer Science of McGill University. Previously, he was a research associate at the German Aerospace Center (DLR) in Oberpfaffenhofen. He has a PhD in electrical and computer engineering from Vanderbilt University in Nashville, Tennessee, and an MSc in electrical engineering from the University of Twente, the Netherlands. His primary research interests are in Computer Automated Multiparadigm Modeling (CAMPaM) with principal applications in design automation, training systems, and fault detection, isolation, and reconfiguration. He designed the Electronics Laboratory Simulator, nominated for the Computerworld Smithsonian Award by Microsoft Corporation in 1994. In 2003, he was awarded the IMechE Donald Julius Groen Prize for a paper on HYBRSIM, a hybrid bond graph modeling and simulation environment. Professor Mosterman received the Society for Modeling and Simulation International (SCS) Distinguished Service Award in 2009 for his services as editor-in-chief of SIMULATION: Transactions of SCS. He is or has been an associate editor of the International Journal of Critical Computer Based Systems, the Journal of Defense Modeling and Simulation, the International Journal of Control and Automation, Applied Intelligence, and IEEE Transactions on Control Systems Technology.
Contributors

Fredrik Abbors
Department of Information Technologies
Åbo Akademi University
Turku, Finland

Veli-Matti Aho
Process Excellence
Nokia Siemens Networks
Tampere, Finland

Lee Barford
Measurement Research Laboratory
Agilent Technologies
Reno, Nevada
and
Department of Computer
Science and Engineering
University of Nevada
Reno, Nevada

Thomas Bauer
Fraunhofer Institute for Experimental
Software Engineering (IESE)
Kaiserslautern, Germany

Fevzi Belli
Department of Electrical Engineering
and Information Technology
University of Paderborn
Paderborn, Germany

Fabrice Bouquet
Computer Science Department
University of Franche-Comté/INRIA
CASSIS Project
Besançon, France

Manfred Broy
Institute for Computer Science
Technische Universität München
Garching, Germany

Mirko Conrad
The MathWorks, Inc.
Natick, Massachusetts

Frédéric Dadeau
Computer Science Department
University of Franche-Comté/INRIA
CASSIS Project
Besançon, France

Thao Dang
VERIMAG CNRS (French National
Center for Scientific Research)
Gieres, France

Gregor Engels
Software Quality Lab—s-lab
University of Paderborn
Paderborn, Germany

Robert Eschbach
Fraunhofer Institute for Experimental
Software Engineering (IESE)
Kaiserslautern, Germany

Luca Ferro
TIMA Laboratory
University of Grenoble, CNRS
Grenoble, France

Michael Gall
Siemens Industry, Inc.
Building Technologies Division
Florham Park, New Jersey

Jens Grabowski
Institute for Computer Science
University of Goettingen
Goldschmidtstraße 7
Goettingen, Germany
Jürgen Großmann
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
Berlin, Germany

Stefan Gruner
Department of Computer Science
University of Pretoria
Pretoria, Republic of South Africa

Axel Hollmann
Department of Applied Data Technology
Institute of Electrical and Computer Engineering
University of Paderborn
Paderborn, Germany

Antti Huima
President and CEO
Conformiq-Automated Test Design
Saratoga, California

Antti Jääskeläinen
Department of Software Systems
Tampere University of Technology
Tampere, Finland

Jacques Julliand
Computer Science Department
University of Franche-Comté
Besançon, France

Marko Käärämies
Department of Computer Science
Tallinn University of Technology
Tallinn, Estonia

Stefan Kaiser
Fraunhofer Institute FOKUS
Berlin, Germany

Mika Katara
Department of Software Systems
Tampere University of Technology
Tampere, Finland

Jani Koivulainen
Conformiq Customer Success
Conformiq
Espoo, Finland

Bogdan Kowalczyk
Delphi Technical Center Kraków
ul. Podgórkí Tynieckie 2
Kraków, Poland

Andres Kull
ELVIOR
Tallinn, Estonia

Bruno Legeard
Research and Development
Smartesting/University of Franche-Comté
Besançon, France

Lan Lin
Department of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, Tennessee

Philip Makedonski
Institute for Computer Science
University of Goettingen
Goldschmidtstraße 7
Goettingen, Germany

Maili Markvardt
Department of Computer Science
Tallinn University of Technology
Tallinn, Estonia

Abel Marrero Pérez
Daimler Center for Automotive IT Innovations
Berlin Institute of Technology
Berlin, Germany

Pierre-Alain Masson
Computer Science Department
University of Franche-Comté
Besançon, France

Stephen P. Masticola
System Test Department
Siemens Fire Safety
Florham Park, New Jersey
Pieter J. Mosterman  
MathWorks, Inc.  
Natick, Massachusetts  
and  
McGill University  
School of Computer Science  
Montreal, Quebec, Canada

Sebastian Oster  
Real-Time Systems Lab  
Technische Universität Darmstadt  
Darmstadt, Germany

Sascha Padberg  
Department of Applied Data Technology  
Institute of Electrical and Computer Engineering  
University of Paderborn  
Paderborn, Germany

Mirosław Panek  
Delphi Technical Center Kraków  
ul. Podgórki Tynieckie 2  
Kraków, Poland

Virginia Papailiopoulou  
INRIA  
Rocquencourt, France

Ioannis Parissis  
Grenoble INP—Laboratoire de Conception et d’Intégration des Systèmes  
University of Grenoble  
Valence, France

Fabien Peureux  
Computer Science Department  
University of Franche-Comté  
Besançon, France

Laurence Pierre  
TIMA Laboratory  
University of Grenoble, CNRS  
Grenoble, France

Jesse H. Poore  
Ericsson-Harlan D. Mills Chair in Software Engineering  
Department of Electrical Engineering and Computer Science  
University of Tennessee  
Knoxville, Tennessee

Alexander Pretschner  
Karlsruhe Institute of Technology  
Karlsruhe, Germany

Kullo Raiend  
ELVIO  
Tallinn, Estonia

Ina Schieferdecker  
Fraunhofer Institute FOKUS  
Kaiserin-Augusta-Allee 31  
Berlin, Germany

Holger Schlingloff  
Fraunhofer Institute FIRST  
Kekulestraße  
Berlin, Germany

Andy Schürr  
Real-Time Systems Lab  
Technische Universität Darmstadt  
Darmstadt, Germany

Besnik Seljimi  
Faculty of Contemporary Sciences and Technologies  
South East European University  
Tetovo, Macedonia

Paweł Skruch  
Delphi Technical Center Kraków  
ul. Podgórki Tynieckie 2  
Kraków, Poland

Jaroslav Svacina  
Fraunhofer Institute FIRST  
Kekulestraße  
Berlin, Germany

Tommi Takala  
Department of Software Systems  
Tampere University of Technology  
Tampere, Finland
Contributors

**Risto Teittinen**  
Process Excellence  
Nokia Siemens Networks  
Espoo, Finland

**Bruce Watson**  
Department of Computer Science  
University of Pretoria  
Pretoria, Republic of South Africa

**Régis Tissot**  
Computer Science Department  
University of Franche-Comté  
Besançon, France

**Stephan Weißleder**  
Fraunhofer Institute FIRST  
Kekulestraße 7  
Berlin, Germany

**Dragos Truscan**  
Department of Information Technologies  
Åbo Akademi University  
Turku, Finland

**Hans-Werner Wiesbrock**  
IT Power Consultants  
Kolonnenstraße 26  
Berlin, Germany

**Mark Utting**  
Department of Computer Science  
University of Waikato  
Hamilton, New Zealand

**Andreas Wübbeke**  
Software Quality Lab—s-lab  
University of Paderborn  
Paderborn, Germany

**Jüri Vain**  
Department of Computer Science/Institute of Cybernetics  
Tallinn University of Technology  
Tallinn, Estonia

**Justyna Zander**  
Harvard University  
Cambridge, Massachusetts  
and  
Fraunhofer Institute FOKUS  
Kaiserin-Augusta-Allee 31  
Berlin, Germany
Technical Review Committee

Lee Barford
Fevzi Belli
Fabrice Bouquet
Mirko Conrad
Frédéric Dadeau
Thao Dang
Thomas Deiss
Vladimir Entin
Alain-Georges Vouffo Feudjio
Gordon Fraser
Ambar Gadkari
Michael Gall
Jeremy Gardiner
Juergen Grossmann
Stefan Gruner
Axel Hollmann
Mika Katara
Bogdan Kowalczyk
Yves Ledru
Pascale LeGall
Jenny Li
Levi Lucio
José Carlos Maldonado
Eda Marchetti
Steve Masticola
Swarup Mohalik
Pieter J. Mosterman
Sebastian Oster
Jan Peleska
Abel Marrero Pérez
Jesse H. Poore
Stacy Prowell
Holger Rendel
Axel Rennoch
Markus Roggenbach
Bernhard Rumpe
Ina Schieferdecker
Holger Schlingloff
Diana Serbanescu
Pawel Skruch
Paul Strooper
Mark Utting
Stefan van Baelen
Carsten Wegener
Stephan Weißleder
Martin Wirsing
Karsten Wolf
Justyna Zander
The purpose of this handbook is to provide a broad overview of the current state of model-based testing (MBT) for embedded systems, including the potential breakthroughs, the challenges, and the achievements observed from numerous perspectives. To attain this objective, the book offers a compilation of 22 high-quality contributions from world-renowned industrial and academic authors. The chapters are grouped into six parts.

- The first part comprises the contributions that focus on key test concepts for embedded systems. In particular, a taxonomy of MBT approaches is presented, an assessment of the merit and value of system models and test models is provided, and a selected test framework architecture is proposed.

- In the second part, different test automation algorithms are discussed for various types of embedded system representations.

- The third part contains contributions on the topic of integration and multilevel testing. Criteria for the derivation of integration entry tests are discussed, an approach for reusing test cases across different development levels is provided, and an X-in-the-Loop testing method and notation are proposed.

- The fourth part is composed of contributions that tackle selected challenges of MBT, such as testing software product lines, conformance validation for hybrid systems and nondeterministic systems, and understanding safety-critical components in the passive test context.

- The fifth part highlights testing in industry including application areas such as telecommunication networks, smartphones, and automotive systems.

- Finally, the sixth part presents solutions for lower-level tests and comprises an approach to validation of automatically generated code, contributions on testing analog components, and verification of SystemC models.

To scope the material in this handbook, an embedded system is considered to be a system that is designed to perform a dedicated function, typically with hard real-time constraints, limited resources and dimensions, and low-cost and low-power requirements. It is a combination of computer software and hardware, possibly including additional mechanical, optical, and other parts that are used in the specific role of actuators and sensors (Ganssle and Barr 2003). Embedded software is the software that is part of an embedded system. Embedded systems have become increasingly sophisticated and their software content has grown rapidly in the past decade. Applications now consist of hundreds of thousands or even millions of lines of code. Moreover, the requirements that must be fulfilled while developing embedded software are complex in comparison to standard software. In addition, embedded
systems are often produced in large volumes, and the software is difficult to update once the product is deployed. Embedded systems interact with the physical environment, which often requires models that embody both continuous-time and discrete-event behavior. In terms of software development, it is not just the increased product complexity that derives from all those characteristics, but it combines with shortened development cycles and higher customer expectations of quality to underscore the utmost importance of software testing (Schäuffele and Zurawka 2006).

MBT relates to a process of test generation from various kinds of models by application of a number of sophisticated methods. MBT is usually the automation of black-box testing (Utting and Legeard 2006). Several authors (Utting, Pretschner, and Legeard 2006; Kamga, Herrmann, and Joshi 2007) define MBT as testing in which test cases are derived in their entirety or in part from a model that describes some aspects of the system under test (SUT) based on selected criteria. In addition, authors highlight the need for having dedicated test models to make the most out of MBT (Baker et al. 2007; Schulz, Honkola, and Huima 2007).

MBT clearly inherits the complexity of the related domain models. It allows tests to be linked directly to the SUT requirements, makes readability, understandability, and maintainability of tests easier. It helps to ensure a repeatable and scientific basis for testing and has the potential for known coverage of the behaviors of the SUT (Utting 2005). Finally, it is a way to reduce the effort and cost for testing (Pretschner et al. 2005).

This book provides an extensive survey and overview of the benefits of MBT in the field of embedded systems. The selected contributions present successful test approaches where different algorithms, methodologies, tools, and techniques result in important cost reduction while assuring the proper quality of embedded systems.

**Organization**

This book is organized in the six following parts: (I) Introduction, (II) Automatic Test Generation, (III) Integration and Multilevel Testing, (IV) Specific Approaches, (V) Testing in Industry, and (VI) Testing at the Lower Levels of Development. An overview of each of the parts, along with a brief introduction of the contents of the individual chapters, is presented next. The following figure depicts the organization of the book.
Part I. Introduction

The chapter “A Taxonomy of Model-Based Testing for Embedded Systems from Multiple Industry Domains” provides a comprehensive overview of MBT techniques using different dimensions and categorization methods. Various kinds of test generation, test evaluation, and test execution methods are described, using examples that are presented throughout this book and in the related literature.

In the chapter “Behavioral System Models versus Models of Testing Strategies in Functional Test Generation,” the distinction between diverse types of models is discussed extensively. In particular, models that describe intended system behavior and models that describe testing strategies are considered from both practical as well as theoretical viewpoints. It shows the difficulty of converting the system model into a test model by applying the mental and explicit system model perspectives. Then, the notion of polynomial-time limit on test case generation is included in the reasoning about the creation of tests based on finite-state machines.

The chapter “Test Framework Architectures for Model-Based Embedded System Testing” provides reference architectures for building a test framework. The test framework is understood as a platform that runs the test scripts and performs other functions such as, for example, logging test results. It is usually a combination of commercial and purpose-built software. Its design and character are determined by the test execution process, common quality goals that control test harnesses, and testability antipatterns in the SUT that must be accounted for.

Part II. Automatic Test Generation

The chapter “Automatic Model-Based Test Generation from UML State Machines” presents several approaches for the generation of test suites from UML state machines based on different coverage criteria. The process of abstract path creation and concrete input value generation is extensively discussed using graph traversal algorithms and boundary value analysis. Then, these techniques are related to random testing, evolutionary testing, constraint solving, model checking, and static analysis.

The chapter “Automated Statistical Testing for Embedded Systems” applies statistics to solving problems posed by industrial software development. A method of modeling the population of uses is established to reason according to first principles of statistics. The Model Language and Java Usage Model Builder Library is employed for the analysis. Model validation and revision through estimates of long-run use statistics are introduced based on a medical device example while paying attention to test management and process certification.

In the chapter “How to Design Extended Finite State Machine Models in Java” extended finite-state machine (EFSM) test models that are represented in the Java programming language are applied to an SUT. ModelJUnit is used for generating the test cases by stochastic algorithms. Then, a methodology for building a MBT tool using Java reflection is proposed. Code coverage metrics are exploited to assess the results of the method, and an example referring to the GSM 11.11 protocol for mobile phones is presented.

The chapter “Automatic Testing of LUSTRE/SCADE Programs” addresses the automation of functional test generation using a LUSTRE-like language in the LUTESS V2 tool and refers to the assessment of the created test coverage. The testing methodology includes the definitions of the domain, environment dynamics, scenarios, and an analysis based on safety properties. A program control flow graph for SCADE models allows a family of coverage criteria to assess the effectiveness of the test methods and serves as an additional basis for the test generation algorithm. The proposed approaches are illustrated by a steam-boiler case study.
In the chapter “Test Generation Using Symbolic Animation of Models,” symbolic execution (i.e., animation) of B models based on set-theoretical constraint solvers is applied to generate the test cases. One of the proposed methods focuses on creation of tests that reach specific test targets to satisfy structural coverage, whereas the other is based on manually designed behavioral scenarios and aims at satisfying dynamic test selection criteria. A smartcard case study illustrates the complementarity of the two techniques.

Part III. Integration and Multilevel Testing

The chapter “Model-Based Integration Testing with Communication Sequence Graphs” introduces a notation for representing the communication between discrete-behavior software components on a meta-level. The models are directed graphs enriched with semantics for integration-level analysis that do not emphasize internal states of the components, but rather focus on events. In this context, test case generation algorithms for unit and integration testing are provided. Test coverage criteria, including mutation analysis, are defined and a robot-control application serves as an illustration.

In the chapter “A Model-Based View onto Testing: Criteria for the Derivation of Entry Tests for Integration Testing” components and their integration architecture are modeled early on in development to help structure the integration process. Fundamentals for testing complex systems are formalized. This formalization allows exploiting architecture models to establish criteria that help minimize the entry-level testing of components necessary for successful integration. The tests are derived from a simulation of the subsystems and reflect behaviors that usually are verified at integration time. Providing criteria to enable shifting effort from integration testing to component entry tests illustrates the value of the method.

In the chapter “Multilevel Testing for Embedded Systems,” the means for a smooth integration of multiple test levels artifacts based on a continuous reuse of test models and test cases are provided. The proposed methodology comprises the creation of an invariant test model core and a test-level specific test adapter model that represents a varying component. Numerous strategies to obtain the adapter model are introduced. The entire approach results in an increased optimization of the design effort across selected functional abstraction levels and allows for the easier traceability of the test constituents. A case study from the automotive domain (i.e., automated light control) illustrates the feasibility of the solution.

The chapter “Model-Based X-in-the-Loop Testing” provides a methodology for technology-independent specification and systematic reuse of testing artifacts for closed-loop testing across different development stages. Simulink®-based environmental models are coupled with a generic test specification designed in the notation called TTCN-3 embedded. It includes a dedicated means for specifying the stimulation of an SUT and assessing its reaction. The notions of time and sampling, streams, stream ports, and stream variables are paid specific attention as well as the definition of statements to model a control flow structure akin to hybrid automata. In addition, an overall test architecture for the approach is presented. Several examples from the automotive domain illustrate the vertical and horizontal reuse of test artifacts. The test quality is discussed as well.

Part IV. Specific Approaches

The chapter “A Survey of Model-Based Software Product Lines Testing” presents an overview of the testing that is necessary in software product line engineering methods. Such methods aim at improving reusability of software within a range of products sharing a common set of features. First, the requirements and a conception of MBT for software product lines are introduced. Then, the state of the art is provided and the solutions are compared to each other based on selected criteria. Finally, open research objectives are outlined and recommendations for the software industry are provided.
The chapter “Model-Based Testing of Hybrid Systems” describes a formal framework for conformance testing of hybrid automaton models and their adequate test generation algorithms. Methods from computer science and control theory are applied to reason about the quality of a system. An easily computable coverage measure is introduced that refers to testing properties such as safety and reachability based on the equal-distribution degree of a set of states over their state space. The distribution degree can be used to guide the test generation process, while the test creation is based on the rapidly exploring random tree algorithm (Lavalle 1998) that represents a probabilistic motion planning technique in robotics. The results are then explored in the domain of analog and mixed signal circuits.

The chapter “Reactive Testing of Nondeterministic Systems by Test Purpose Directed Tester” provides a model-based construction of an online tester for black-box testing. The notation of nondeterministic EFSM is applied to formalize the test model. The synthesis algorithm allows for selecting a suboptimal test path at run time by finding the shortest path to cover the test purpose. The rules enabling an implementation of online reactive planning are included. Coverage criteria are discussed as well, and the approach is compared with related algorithms. A feeder-box controller of a city lighting system illustrates the feasibility of the solution.

The chapter “Model-Based Passive Testing of Safety-Critical Components” provides a set of passive-testing techniques in a manner that is driven by multiple examples. First, general principles of the approach to passive quality assurance are discussed. Then, complex software systems, network security, and hardware systems are considered as the targeted domains. Next, a step-by-step illustrative example for applying the proposed analysis to a concurrent system designed in the form of a cellular automaton is introduced. As passive testing usually takes place after the deployment of a unit, the ability of a component to monitor and self-test in operation is discussed. The benefits and limitations of the presented approaches are described as well.

Part V. Testing in Industry

The chapter “Applying Model-Based Testing in the Telecommunication Domain” refers to testing practices at Nokia Siemens Networks at the industrial level and explains the state of MBT in the trenches. The presented methodology uses a behavioral system model designed in UML and SysML for generating the test cases. The applied process, model development, validation, and transformation aspects are extensively described. Technologies such as the MATERA framework (Abbors, Bäcklund, and Truscan 2010), UML to QML transformation, and OCL guideline checking are discussed. Also, test generation, test execution aspects (e.g., load testing, concurrency, and run-time executability), and the traceability of all artifacts are discussed. The case study illustrates testing the functionality of a Mobile Services Switching Center Server, a network element using offline testing.

The chapter “Model-Based GUI Testing of Smartphone Applications: Case S60™ and Linux®” discusses application of MBT along two case studies. The first one considers built-in applications in a smartphone model S60, and the second tackles the problem of a media player application in a variant of mobile Linux. Experiences in modeling and adapter development are provided and potential problems (e.g., expedient pace of product creation) are reported in industrial deployment of the technology for graphical user interface (GUI) testing of smartphone applications. In this context, the TEMA toolset (Jääskeläinen 2009) designed for test modeling, test generation, keyword execution, and test debugging is presented. The benefits and business aspects of the process adaptation are also briefly considered.

The chapter “Model-Based Testing in Embedded Automotive Systems” provides a broad overview of MBT techniques applied in the automotive domain based on experiences from Delphi Technical Center, Kraków (Poland). Key automotive domain concepts specific to
MBT are presented as well as everyday engineering issues related to MBT process deployment in the context of the system-level functional testing. Examples illustrate the applicability of the techniques for industrial-scale mainstream production projects. In addition, the limitations of the approaches are outlined.

Part VI. Testing at the Lower Levels of Development

The chapter “Testing-Based Translation Validation of Generated Code” provides an approach for model-to-code translation that is followed by a validation phase to verify the target code produced during this translation. Systematic model-level testing is supplemented by testing for numerical equivalence between models and generated code. The methodology follows the objectives and requirements of safety standards such as IEC 61508 and ISO 26262 and is illustrated using a Simulink-based code generation tool chain.

The chapter “Model-Based Testing of Analog Embedded Systems Components” addresses the problem of determining whether an analog system meets its specification as given either by a model of correct behavior (i.e., the system model) or of incorrect behavior (i.e., a fault model). The analog model-based test follows a two-phase process. First, a pretesting phase including system selection, fault model selection, excitation design, and simulation of fault models is presented. Next, an actual testing phase comprising measurement, system identification, behavioral simulation, and reasoning about the faults is extensively described. Examples are provided while benefits, limitations, and open questions in applying analog MBT are included.

The chapter “Dynamic Verification of SystemC Transactional Models” presents a solution for verifying logic and temporal properties of communication in transaction-level modeling designs from simulation. To this end, a brief overview on SystemC is provided. Issues related to globally asynchronous/locally synchronous, multicloked systems, and auxiliary variables are considered in the approach.

Target Audience

The objective of this book is to be accessible to engineers, analysts, and computer scientists involved in the analysis and development of embedded systems, software, and their quality assurance. It is intended for both industry-related professionals and academic experts, in particular those interested in verification, validation, and testing. The most important objectives of this book are to help the reader understand how to use Model-Based Testing and test harness to a maximum extent. Various perspectives serve to:

- Get an overview on MBT and its constituents;
- Understand the MBT concepts, methods, approaches, and tools;
- Know how to choose modeling approaches fitting the customers’ needs;
- Be able to select appropriate test generation strategies;
- Learn about successful applications of MBT;
- Get to know best practices of MBT; and
- See prospects of further developments in MBT.
References


