Runtime Verification

Christian Colombo • Gordon J. Pace

Runtime Verification

A Hands-On Approach in Java

Springer

Christian Colombo
Department of Computer Science
Faculty of Information
and Communications Technology
University of Malta
Msida, Malta

Gordon J. Pace
Department of Computer Science
Faculty of Information
and Communications Technology
University of Malta
Msida, Malta

ISBN 978-3-031-09266-4 ISBN 978-3-031-09268-8 (eBook)
https://doi.org/10.1007/978-3-031-09268-8

© Springer Nature Switzerland AG 2022
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Foreword

In all my many years of software development, I have never seen runtime verification done as part of the testing process and now I cannot see how you could test a software product without it.

R. T. Brownrigg

The above quote is an unedited snippet from an e-mail by Richard Thomas Brownrigg. Richard is a distinguished software developer and project leader with numerous decades of experience. He wrote the e-mail to his wife, directly after participating in a one-day tutorial about Runtime Verification given by one of the authors of this book. I quote the message here, with Richard's permission, because it represents very well two truths about the topic at hand. First, Runtime Verification is not yet a mainstream technique in software development practice (at the time of writing this foreword). Second, Runtime Verification, in particular in the incarnation that Colombo and Pace put forward here, is very accessible and convincing to software developers once they are exposed to it, offering huge benefits on relatively little costs.

I am convinced that this book will have a strong impact on spreading the usage of Runtime Verification in software practice, thereby contributing greatly to the quality of the development of software systems, and indeed to the safe operation after deployment. The reason for my conviction is not simply the potential of Runtime Verification as a technology. That also. But it is very much the particular approach and style Colombo and Pace take in this book. It is not meant to barely inform the reader about the topic, but to put the reader in the position to actively use Runtime Verification in development and deployment of real software. The reader is part of a very practical journey, through exercise-centric chapters, which provide step-by-step insights into the usage of the discussed concepts and techniques.

When reading this book, what amazed me most, however, is the approach Colombo and Pace take to learning. The best way to truly understand any concept is to first invest into solving problems *without* it. Only then, after having experienced the targeted challenges, the new concept is introduced, whereafter we can *experience the empowerment* that the introduced concept provides us with. The authors follow this philosophy quite strictly and carefully. Manual monitoring is exercised before aspect-oriented programming is introduced, allowing for a better separation of concerns, and higher automation when realising monitored systems. This is, again, exercised, before guarded command specifications are introduced, allowing to specify properties that are easier to write, read, and maintain than aspects. Once more, this is put into practice before the introduction of automata allows us to, more concisely, match the (graphical) intuition of a system manoeuvring through different modes of operation. The book moves on in exactly this way to further topics: regular expressions, temporal logic, real-time monitoring, reactive monitoring, and offline verification. Time and again, the course of the text and the accompanying exercises assure that every new plateau of the toolbox is explored practically, before the next plateau offers new solutions to the problems we have already experienced.

With that, I wish the reader a great journey, empowering you with a new way of developing and deploying robust software solutions.

Gothenburg Wolfgang Ahrendt
May 2022

Preface

Runtime verification has been around for over 20 years, with a substantial body of literature and numerous tools available. In the couple of decades we have been working within the area of runtime verification, we have witnessed and contributed to the growth of the field: the theoretical underpinnings, new verification techniques, new instrumentation strategies, development of tools, and their application in industrial case studies.

The techniques refined over these past years feel ripe for adoption in industry and yet, while one can find various theoretical overviews and academic articles, a hands-on manual which guides the reader from zero knowledge to sufficient practical knowledge required to consider its use in industry is still sorely missing. Given how industry-friendly runtime verification is, lack of such a text is surprising and regretful.

This book aims to fill this void, starting with no assumptions on the knowledge of the reader and providing exercises throughout the book through which the reader builds their own runtime verification tool. All that is required are basic programming skills and a good working knowledge of the object-oriented paradigm, ideally Java.

The book does not aim to be an exhaustive overview of the area of runtime verification. Many important research questions are still the topic of active research and discoveries, but in this book we stick to the practical underlying principles and techniques. At the end of the book we touch upon a number of more specialised areas, such as the monitoring of distributed systems and real-time systems, but such advanced topics are beyond the scope of the book. Furthermore, although we cover the pragmatic side of runtime verification, we do not delve into the theoretical foundations. However, for each chapter we do provide a reading list in the appendix for the interested reader who would like to deepen their knowledge in a particular area.

Reading guide: The first six chapters of the book should be read in order and are prerequisites for all the others that follow. Chapters 7–10 can be read independently, since each look into the use of different specification language

requirements: whether automata, regular expressions, linear time temporal logic, or real-time properties. Similarly, Chapters 11–13 are largely independent, focusing on the practical implications of runtime verification: Chapters 11 and 12 are duals of each other on the spectrum of monitor intrusiveness, while Chapter 13 gives a brief overview of a number of more advanced topics, ranging from concerns of efficiency and persistence, to integration with testing and architectural considerations.

Christian Colombo
Gordon J. Pace
April 2022

Acknowledgements

The authors are indebted to the Runtime Verification community for many informal discussions about the contents of the book. We are also grateful for the opportunity to present early (and redacted) versions of the material herein as part of Escuela de Ciencias Informáticas (ECI) intensive courses organised by the University of Buenos Aires, and at the two organised schools as part of the ARVI COST Action (IC1402). We would also like to thank our runtime verification students over the years, too many to name, but whose input was crucial to improve the quality of the material.