Behavioural Models

Matthias Kunze · Mathias Weske

Behavioural Models

From Modelling Finite Automata to Analysing Business Processes

Springer

Matthias Kunze
Zalando SE
Berlin
Germany

Mathias Weske
Hasso Plattner Institute (HPI)
University of Potsdam
Potsdam
Germany

ISBN 978-3-319-44958-6 ISBN 978-3-319-44960-9 (eBook)
DOI 10.1007/978-3-319-44960-9

Library of Congress Control Number: 2016949114

Printed on acid-free paper

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

for M. A. J.
M.K.

for Daniela, Jonathan, Emilia, and Theresa
M.W.

Foreword

Bridges and buildings are here to stay and thus mainly need static descriptions for engineering them. Software is designed and built to perform actions in various forms. Designing complex behaviour is a difficult and error-prone task that enforces solid intellectual capabilities to understand the temporal and often very dynamic behaviour of a software system. Behaviour is generally more difficult to design than the pretty static data and architectural structures of software systems.

In the age of digitalisation of virtually every domain, designing appropriate software systems is one of the big challenges of our time. Their scope, their complexity, and their interactions with users can only be understood with a solid theory that is at the same time practically applicable. Designing and understanding behaviour of systems becomes relevant not only for the software itself, but for intelligent and smart technical solutions that usually embody several interacting subsystems, often running in a cloud environment. Intelligent products will allow, but also enforce us to interact with them in complex behavioural patterns that need to be designed well and thoroughly understood and analysed when composing even more complex systems, services and business processes. Interactions between intelligent components that autonomously control our traffic, electric power and water supply system also need to be established very precisely in order to avoid errors that might have severe consequences.

Whereas most engineering disciplines use mathematical calculus for describing continuous behaviour, computer science has developed a solid theory for digital transitions between concretely definable set of states. This theory has been defined and explored in various variants of finite automata, state machines, temporal logics, Petri nets, and business processes. They all have a shared understanding of state and discrete state changes using events and actions.

The book written by Matthias Kunze und Mathias Weske covers the core of this theory and incrementally adds and discusses the various extensions that have been created over the last few decades. This includes traditional sequential

behaviour and concurrent behaviour, the latter of which is particularly useful to model systems with independent subsystems. The book very nicely shows how these different variants of the digital theory are related to each other. It also discusses how concurrent models, such as Petri nets and interacting systems, relate to sequential state transition systems. The book gives deeper insights on how to model interactions between software systems and their users in form of business process models and behavioural patterns to describe them.

Modelling behaviour is without doubt an important aspect of systems engineering. But an equally important question is how to constructively derive an implementation from the model and how to analyse and understand the complex models that have been built. As implementing those models is relatively straightforward the book concentrates very much on the analysis of the modelled behaviour. It discusses the concept of state spaces as underlying semantic concept that allows us to compare model behaviour not only by the model structure but by the behavioural semantics of models. It describes several relations between behavioural models based on model checking. It also discusses several forms of behavioural extension, refinement and inheritance. These techniques are very helpful to reuse common parts of business processes as well as standards and to individualize them to the specific needs of software development projects. The concepts introduced in this book are the core for all forms of software development that build on reusable, high quality assets.

The reader thus gets a thorough understanding of the various behavioural modelling techniques and how they relate to each other. This will help the reader to use behavioural modelling techniques when designing their own system and trying to understand early whether the implementation will behave properly.

This text reads very well without giving up the claim of providing a solid formally grounded content and technical presentation. The book by Kunze and Weske provides a valuable in-depth overview of modelling in the digital era. It will thus be helpful for practitioners to understand the core assets as well as for students who get an easy accessible, yet formally grounded overview of behavioural modelling techniques and their analysis forms.

Prof. Dr. Bernhard Rumpe
RWTH Aachen University, June 2016

Preface

Engineers are experiencing increasing requirements regarding the functionality and reliability of systems to be built. To cope with the resulting complexity, they use models to specify and analyse systems. In general, systems can be specified from several different perspectives, including structure and behaviour.

The structure of a system refers to its architecture and the data that is stored and manipulated. System behaviour covers the dynamic aspects of systems. It describes essentially how systems operate. From its beginning more than half a century ago, the computer science community has developed methods and techniques to capture the behaviour of systems. Today, engineers and system designers use standardised methods and techniques to describe systems in terms of both, their structural and their behavioural aspects.

With the increasing complexity of systems, the demand for modelling and analysis of systems is also growing. As a result, a thorough understanding of the principles and concepts of behavioural modelling and its application in today's standard modelling techniques is essential for any software engineer and any system designer.

We wrote this book as a textbook for students of computer science and software engineering, and also for programmers and system analysts who are interested in the question "How can we represent the behaviour of the system we are developing and how can we prove its essential properties?"

To answer this question, this book takes readers on a journey from the fundamentals of behavioural modelling to advanced techniques for modelling and analysing sequential and concurrent systems. Undergraduate students will want to focus on Parts I and II of this book, while graduate students and researchers might find Part III especially valuable.

This book is based on lectures that the second author gave to students of IT systems engineering at the University of Potsdam. The core aspects of behavioural modelling were covered in a lecture on system modelling for first-year students. Advanced concepts were covered in lectures related to business process management.

This book consists of three parts. The foundations of behavioural modelling are addressed in Part I. Chapter 2 introduces discrete dynamic systems by demarcating them from continuous systems. Transition systems are introduced as a basic formalism to represent the behaviour of discrete dynamic systems and causality is discussed, which is a fundamental concept for modelling and reasoning about behaviour. Part II forms the centre of gravity of the book. It is devoted to models of behaviour. Chapter 3 introduces sequential systems by investigating different types of automata, ranging from simple finite automata to automata with output and extended automata with variables and assignments. The chapter concludes with state machines, which are a realisation of extended automata in the UML industry standard. Concurrent systems are at the centre of Chapter 4. State machines with orthogonal states are introduced, since these can represent concurrent behaviour. Concurrency can also be represented by asynchronous interactions of several independent, sequential systems. The chapter concludes by introducing several different types of Petri nets, which provide important concepts for modelling and analysing concurrent behaviour of systems. Business process models are covered in Chapter 5. Based on a set of workflow patterns, whose semantics is represented by Coloured Petri nets, this chapter looks at the Business Process Model and Notation (BPMN), which is the industry standard for modelling business processes. The semantics of BPMN process diagrams and collaboration diagrams are defined by a mapping to Petri nets. Part III investigates how the behaviour of systems can be analysed. The basis of this part is formed by Chapter 6, which introduces the concept of state spaces. State spaces can be regarded as unified models of behaviour, which, can therefore serve as a basis for behavioural analysis. The state space of sequential systems described by different types of automata and the state space of concurrent systems are investigated. The comparison of behaviour is covered in Chapter 7 by looking at behavioural equivalence, inheritance, and similarity. The formal analysis of behavioural models is covered in the verification Chapter 8, which looks at temporal logic and model checking before investigating the behavioural properties of systems and, ultimately, business process compliance.

Many people have a share in the success of a book project. The authors thank the members of the Business Process Technology research group at the Hasso Plattner Institute for their support and commitment. In particular, we thank Marcin Hewelt, Luise Pufahl, and Kimon Batoulis for proofreading of book chapters. We also thank our students in the bachelor's and master's programmes for their involvement and their critical suggestions, which – through the lectures – found their way into this book as well.

Berlin and Potsdam, June 2016 *M.K.* and *M.W.*

Contents