Texts in Theoretical Computer Science
An EATCS Series

Wan Fokkink

Modelling Distributed Systems

With 18 Figures and 7 Tables

Springer

Author

Wan Fokkink
Vrije Universiteit Amsterdam
Department of Computer Science
Section Theoretical Computer Science
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
wanf@cs.vu.nl

Series Editors

Prof. Dr. Wilfried Brauer
Institut für Informatik der TUM
Boltzmannstr. 3
85748 Garching, Germany
brauer@informatik.tu-muenchen.de

Prof. Dr. Juraj Hromkovič
ETH Zentrum
Department of Computer Science
Swiss Federal Institute of Technology
8092 Zürich, Switzerland
juraj.hromkovic@inf.ethz.ch

Prof. Dr. Grzegorz Rozenberg
Leiden Institute of Advanced
Computer Science
University of Leiden
Niels Bohrweg 1
2333 CA Leiden, The Netherlands
rozenber@liacs.nl

Prof. Dr. Arto Salomaa
Turku Centre of
Computer Science
Lemminkäisenkatu 14 A
20520 Turku, Finland
asalomaa@utu.fi

Library of Congress Control Number: 2007933179

ACM Computing Classification (2007): F.3.1, D.2.4, C.2, I.6

ISSN 1862-4499

ISBN 978-3-540-73937-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Cover Design: KünkelLopka, Heidelberg
Typesetting: by the author
Production: Integra Software Services Pvt. Ltd., India

Printed on acid-free paper 45/3100/Integra 5 4 3 2 1 0

Preface

A distributed system is driven by its separate concurrent components, which are being executed in parallel. In today's world of wireless and mobile networking, distributed algorithms and network protocols tend to constitute an important aspect of system design. Verifying the correctness of such algorithms and protocols tends to be a formidable task, as even simple behaviours become wildly complicated when they are executed in parallel.

Much effort is being spent on the development of novel techniques for the formal description and analysis of distributed systems. However, the majority of these techniques have up to now not been used widely, due to the sharp learning curve required to adopt them. Such verification techniques often have non-trivial theoretical underpinnings, and, as a result, according to practitioners, it requires in-depth knowledge and sophisticated mathematical skills to apply them.

The main aim of this book is to provide a gentle guide to some of the most prominent formal verication techniques for distributed systems. For a start, the reader is acquainted with the algebraic specification of distributed systems. The μCRL toolset is used as a vehicle to teach students how to specify and analyse real-life distributed algorithms and network protocols with the support of specialised tools. μCRL consists of a specification language and verification toolset based on process algebra and abstract data types. Such formal system specifications can be verified at two different levels: either by reasoning about such a specification on a symbolic level, or by generating its state space explicitly. State-of-the-art methods are presented for these two different verification approaches.

Case studies have a valuable role to play both in promoting and demonstrating particular verification techniques, and by providing practical examples of their application. At the same time, case studies help in pushing forward the boundaries of verification techniques. Therefore, formal specifications of several network protocols from the literature are studied in detail, to illustrate how the framework can be applied.

This book was developed from a set of lecture notes for an MSc course on 'Protocol Validation', which I have been lecturing at the Vrije Universiteit Amsterdam since 2001. For prospective lecturers there is a set of slides available on the Web, which can be used to present a course based on this book. Also lab exercises and example specifications are available. I strongly recommend that lecturers include one substantial and open-ended practical exercise, in which the students should (in teams of two or three) specify and verify a real-life distributed system. The book offers one such case study, in the form of a trolley bed on which a patient can lie inside a medical scanning machine for magnetic resonance imaging.

My earlier book *Introduction to Process Algebra*, which appeared in the same series in 2000, can in principle be used as a companion. In that book, the theoretical foundations of process algebra are explained in full detail. Here, we take a more pragmatic view, in that the basics of process algebra and abstract data types are only explained up to a level that suffices for using them in the specification and verification of distributed algorithms and network protocols. The mathematical proofs underlying the verification techniques are largely omitted.

I would like to thank the assistants and students who took part in the course 'Protocol Validation' for their constructive comments and suggestions regarding the lecture notes. For the structure of Chaps. 2 and 3, I benefited from reading the chapter on *Algebraic Process Verification* in the *Handbook of Process Algebra*, by Jan Friso Groote and Michel Reniers, who also provided useful feedback on earlier versions of the book. Moreover, Jan Friso Groote provided the system description of the patient support system.

Utrecht, *Wan Fokkink*
March 2007

Contents