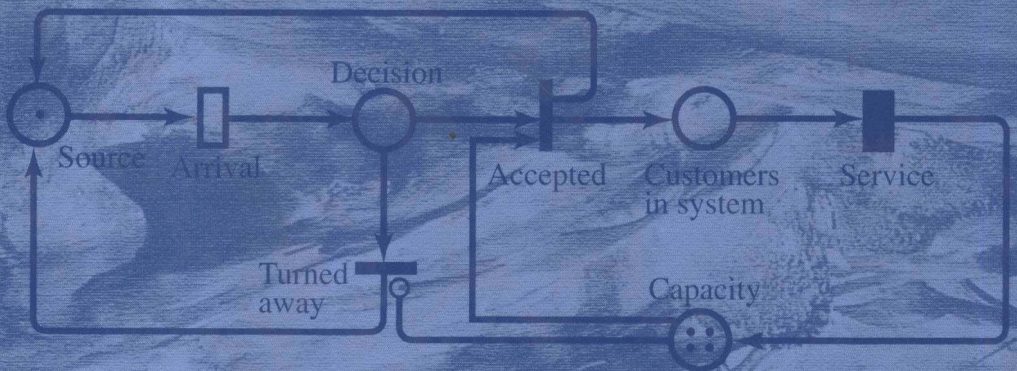


Performance Modelling with Deterministic and Stochastic Petri Nets

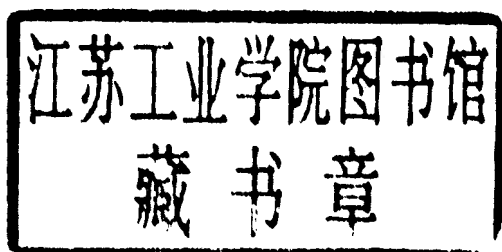


CHRISTOPH LINDEMANN

Performance Modelling with Deterministic and Stochastic Petri Nets

Christoph Lindemann

GMD Research Institute FIRST, Berlin, Germany



JOHN WILEY AND SONS

Chichester · New York · Brisbane · Toronto · Singapore

Copyright © 1998 by John Wiley & Sons Ltd,
Baffins Lane, Chichester,
West Sussex PO19 1UD, England

National 01243 779777
International (+44) 1243 779777

e-mail (for orders and customer service enquiries): cs-books@wiley.co.uk.
Visit our Home Page on <http://www.wiley.co.uk>
or <http://www.wiley.com>

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London, UK W1P 9HE, without the permission in writing of the publisher.

Neither the author nor John Wiley & Sons Ltd accept any responsibility or liability for the loss or damaged occasioned to any person or property through using the materials, instructions, methods or ideas contained herein, or acting or refraining from acting as a result of such use. The authors and Publisher expressly disclaim all implied warranties, including merchantability or fitness for any particular purpose. There will be no duty on the author or Publisher to correct any errors or defects in the software.

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

WILEY-VCH Verlag GmbH, Pappelallee 3,
D-69469 Weinheim, Germany

Jacaranda Wiley Ltd, 33 Park Road, Milton,
Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01,
Jin Xing Distripark, Singapore 129809

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0 471 97646 6

Produced from camera-ready copy supplied by the author

Printed and bound in Great Britain by Bookcraft (Bath) Ltd

This book is printed on acid-free paper responsibly manufactured from sustainable forestation, for which at least two trees are planted for each one used for paper production.

Performance Modelling with Deterministic and Stochastic Petri Nets

Preface

This text provides an up-to-date treatment of the fundamental techniques and algorithms for numerical analysis of deterministic and stochastic Petri nets, a particular stochastic modelling formalism, and the application of this modelling formalism to performance analysis of parallel computer architectures. The material grew from work conducted in the last six years at the Technische Universität Berlin, at the GMD Institute for Computer Architecture and Software Technology (GMD FIRST) in Berlin, and at the IBM Almaden Research Center in San Jose, CA.

Performance evaluation is an applied discipline of computer science which has its roots in queuing theory and teletraffic engineering. A discussion on the evolution of the discipline performance evaluation between 1965 and 1990 appeared in 1991 as an editorial of the journal *Performance Evaluation* [Rei91]. The following historical review continues in this light with emphasis on newly proposed modelling formalisms and the evolution of their numerical solution techniques in the last fifteen years.

Methodological research work in the 1970s emphasised numerical algorithms for steady-state analysis of queuing networks with product-form solution. Typically, such algorithms are based on simple iterative schemes. The first class of such schemes successively considers individual queues of the queuing network, i.e., they exploit the property that the solution has a product form. The second class of such simple iterative schemes considers the entire network starting with a single customer and iterates on the number of customers until the desired customer population is reached. At that time computer equipment was a mainframe connected to terminals or a punchcard reader. Since main memory was quite limited in these computer systems, the amenability of numerical solution methods was restricted to the class of queuing networks with product-form solution. Major results in this decade include the BCMP theorem named after its authors, Baskett, Chandy, Muntz and Palacios [BCM+75], Norton's theorem for queuing networks [CHW75], and the Mean Value Analysis algorithm [RL80].

In the 1980s, theoretically known numerical methods for transient and stationary analysis of Markov chains were extended and refined in order to put them to work.

Markov chains have more modelling power than product-form queuing networks, although they allow only the representation of exponentially distributed delays. Algorithms for steady-state analysis are based on direct or iterative methods for the solution of a sparse linear system of equations whereas algorithms for transient analysis employ iterative techniques for solving a system of ordinary differential equations. Computer equipment available at that time consisted of early personal computers and first-generation workstations. Such workstations could compute transient solutions of Markov chains up to a few thousands of states and the steady-state solution of Markov chains up to several tens of thousands of states. A major result in this decade was the introduction of stochastic Petri nets [Mol82] and generalised stochastic Petri nets as high-level representations of Markov chains [ABC84]. Moreover, Ajmone Marsan and Chiola proposed deterministic and stochastic Petri nets [AC87] which include exponentially distributed and deterministic delays. Further important results in that decade constituted the practical discussion of the randomisation technique [GM84] in conjunction with the development of a stable numerical method for computing Poisson probabilities [FG88]. Later, it was shown that depending on structural properties, the stochastic process underlying a deterministic and stochastic Petri net is either a Markov regenerative process [CKT94] or a generalised semi-Markov process [LS96].

Methodological research work in the 1990s is focusing on time- and memory-efficient numerical algorithms for transient and steady-state analysis of Markov regenerative processes and more general stochastic processes. Algorithms for the analysis of these stochastic processes do not only involve numerical solution of one linear system of equations. Depending on the stochastic process, their analysis also involves numerical solution of a system of ordinary differential equations and/or solution of a system of stochastic integro-differential equations. At present, powerful personal computers and very powerful workstations with several RISC processors and hundreds of Mbytes of main memory are widely available. Numerical methods have been implemented that can compute steady-state solutions of a restricted class of Markov regenerative processes with up to several hundred thousand states and moderate stiffness in the delay parameters on a modern workstation in one hour of CPU time (see Chapter 4). At present, there exists no large-scale implementations of a numerical method for more general stochastic processes. However, methodological results presented in Chapter 5 show great promise that analysis of a quite general stochastic process (i.e., the generalised semi-Markov process with exponential and deterministic events) can be conducted in parallel on a cluster of four to six modern workstations in a similar amount of time.

So far, two textbooks have been devoted to analysis and applications of generalised stochastic Petri nets: [ABC86] and [ABC+95]. Furthermore, a few recent textbooks contain a chapter dealing with some classes of stochastic Petri nets: [BDM+94], [Kan92], and [VN92]. To the best of the author's knowledge, no

monograph on the analysis and application of deterministic and stochastic Petri nets has been published so far. This text tries to fill this gap. It covers methodological results on the numerical analysis of deterministic and stochastic Petri nets and their application to performance modelling of parallel computer architectures. The author considers himself as an engineer who has been fortunate to work with gifted mathematicians. As a consequence, the exposition of the text is engineering-oriented, i.e., intuitive explanations for mathematical results are provided instead of rigorous mathematical proofs. The intended audience are computer scientists, applied mathematicians, and electrical engineers who are interested in performance analysis of computer systems and communication networks. The text is accompanied by a CD-ROM containing the object code of the software package DSPNexpress for several hardware platforms and specification files of a variety of deterministic and stochastic Petri net models. Readers, who are interested in further information on DSPNexpress, are welcome to visit the DSPNexpress Web page mentioned above and to contact the author.

ACKNOWLEDGEMENTS

Several people contributed in one or more ways to this text. First, I would like to express my sincere gratitude to Professor Wolfgang Giloi for his continuous support of my career and his impact on my work. The participation in research projects he initiated since 1990 and the collaboration with him and his co-workers introduced me to the challenging and rapidly emerging field of computer architecture. I also owe a significant debt to Professor Kishor Trivedi for his interest in my research work and our joint work on numerical methods for acyclic Markov chains.

The preparation of this text has also benefited from joint work and technical discussions with a number of people. In particular, I would like to mention Heinz Beilner, Gianfranco Ciardo, Reinhard German, Peter Glynn, Gerd Heber, Holger Hermanns, Manish Malhotra, Samuel Pletner, Friedrich Schön, and Gerald Shedler. Moreover, it is my pleasure to acknowledge the contributions and help of numerous (former) students of the Technische Universität Berlin, who worked with me in the last six years. In particular, I am very grateful to Tobias Bading, Enrik Baumann, Stefan Busse, Christian Lühe, Martin Müller, and Armin Zimmermann. The software package DSPNexpress would not have become reality without their excellent programming capabilities and their dedication to this project. Furthermore, I would like to thank Michael Bäuning and Matthias Weiss for developing programs for the two examples presented in Chapter 5.

Last, but by no means least, the financial support received from the German Research Council (DFG), from the GMD Research Institute FIRST, and from the IBM Almaden Research Center is gratefully acknowledged.

*This text is dedicated to
Professor Wolfgang K. Giloi
– an outstanding researcher
with whom I was fortunate to work.*

Ch. L.

Contents

Preface	xi
Part I Introduction to Performance Modelling	1
Chapter 1 Performance Evaluation of Computer Systems	3
1.1 Why Performance Modelling?	3
1.2 Performance Evaluation Techniques	5
1.2.1 The Art of Performance Evaluation	5
1.2.2 Deterministic Evaluation versus Stochastic Modelling	6
1.2.3 Numerical Solution Methods versus Discrete-event Simulation	9
1.3 The Process of Performance Modelling	12
1.4 Important Professional Organisations	14
1.5 Related Technical Journals and Major Conferences	15
1.6 Scope and Organisation of this Text	19
Chapter 2 Stochastic Modelling Formalisms	23
2.1 Desired Features and Modelling Power	23
2.2 Queuing Networks	27
2.2.1 Separable Queuing Networks	27
2.2.2 Extended Queuing Networks	29
2.3 Generalised Stochastic Petri Nets	31
2.4 Stochastic Process Algebras	34
2.4.1 Introduction to the Stochastic Process Algebra TIPP	34
2.4.2 Stochastic Process Algebra Description of the Multiprocessor Model	39
2.5 Comparative Evaluation of the Considered Modelling Formalisms	42
2.5.1 Queuing Networks versus Stochastic Petri Nets	42
2.5.2 Stochastic Petri Nets versus Stochastic Process Algebras	44
2.6 Summary	46
Bibliography for Part I	47
Part II Deterministic and Stochastic Petri Nets	53
Chapter 3 Structural Properties and Reachability Analysis	55
3.1 Fundamental Definitions and Structural Properties	55

3.2 Marking-dependent Definitions for Building Blocks of Deterministic and Stochastic Petri Nets Deterministic and Stochastic Petri Nets	69
3.2.1 The Extended Definition	69
3.2.2 An Application Example	74
3.3 Algorithmic Generation of the Tangible Reachability Graph	78
3.3.1 Description of the Algorithm	78
3.3.2 Comparison of Computational Cost and Memory Requirements	85
Chapter 4 Numerical Analysis of DSPN without Concurrent Deterministic Transitions	91
4.1 The Approach Based on an Embedded Markov Chain	91
4.1.1 The Marking Process	91
4.1.2 Description of the Numerical Solution Method	94
4.1.3 An Example: A DSPN of an $E_r/D/1/K$ Queuing System	101
4.2 Efficient Computation of the Probability Matrix of the Embedded Markov Chain	106
4.2.1 Transient Analysis of Continuous-time Markov Chains with the Randomisation Technique	106
4.2.2 Efficient Computational Formulas for the Transient Quantities of Subordinated Markov Chains	110
4.2.3 Special Structures for Subordinated Markov Chains	
4.2.4 Isomorphisms between Connected Components of Subordinated Markov Chains	116
4.2.5 Numerical Experiments	120
4.3 Marking-dependent Firing Delays for Deterministic Transitions	126
4.3.1 Motivation	126
4.3.2 Derivation of the Modified Computational Formulas	128
4.3.3 An Example: A DSPN of an $M/D/1/K$ Queuing System with Degradable Service Time	131
4.4 Summary	134
Chapter 5 Numerical Analysis of DSPN with Concurrent Deterministic Transitions	137
5.1 The Approach Based on a General State Space Markov Chain	138
5.1.1 Representation of the Marking Process as a Generalised Semi-Markov Process	138
5.1.2 The Embedded General State Space Markov Chain	139
5.1.3 Algorithmic Description	142
5.2 Numerical Computation of the Transition Kernel	143
5.2.1 Determination of the Step Size	143
5.2.2 The Transition Kernel of the GSSMC underlying a DSPN without Concurrent Deterministic Transitions	147
5.2.3 Computation of Kernel Elements Using Subordinated Markov Chains	149
5.2.4 First Example: A DSPN of an $M/D/1/K$ Queuing System	151

5.2.5 The Transition Kernel of the GSSMC underlying a DSPN with Concurrent Deterministic Transitions	155
5.2.6 Second Example: A DSPN of an M/D/2/K Queuing System	158
5.2.7 Numerical Computation of Derivatives of Kernel Elements	165
5.3 The System of Stationary Equations.....	167
5.3.1 Stationary Equations for a DSPN without Concurrent Deterministic Transitions	167
5.3.2 Stationary Equations for a DSPN with Concurrent Deterministic Transitions	168
5.3.3 Numerical Solution of the Volterra Equations	176
5.4 Numerical Examples	179
5.5 Summary.....	182
Bibliography for Part II.....	185
Part III Performance Analysis of Multiprocessor Systems	191
Chapter 6 Organisation of Multiprocessor Systems.....	193
6.1 Classification of Computer Architectures.....	193
6.2 Multiprocessor Systems with Distributed Shared Memory.....	198
6.3 Multithreaded Architectures.....	202
Chapter 7 Modelling Node Architectures.....	205
7.1 Introduction	205
7.1.1 Node Architectures for MIMD Multiprocessor Systems.....	205
7.1.2 The DSPN Modelling Approach for Node Architectures.....	206
7.2 The Basic Node Architecture and its DSPN Model.....	208
7.3 The MANNA System	210
7.3.1 The Node Architecture of MANNA	210
7.3.2 The DSPN Model.....	212
7.4 The META System	214
7.4.1 The Node Architecture of META	214
7.4.2 The DSPN Model.....	216
7.5 Performance Results.....	219
7.5.1 Discussion of the Employed Approximate Numerical Method.....	219
7.5.2 Performance Curves.....	220
7.6 Summary.....	224
Chapter 8 Modelling Memory Consistency Protocols.....	225
8.1 Introduction	225
8.1.1 Consistency Models and Consistency Protocols for Multiprocessors with Distributed Shared Memory	225
8.1.2 The DSPN Modelling Approach for Memory Consistency Protocols.....	226

8.2 The Sequential Memory Consistency Model	230
8.2.1 Definition and Sufficient Conditions of Sequential Consistency	230
8.2.2 The DSPN of the Sequential Consistency Protocol.....	231
8.3 The Release Memory Consistency Model.....	235
8.3.1 Definition of Release Consistency for Software Implementations of Distributed Shared Memory	235
8.3.2 The Adaptive Consistency Protocol	236
8.3.3 The Eager Release Consistency Protocol.....	240
8.3.4 The Delayed Release Consistency Protocol.....	244
8.3.5 The Lazy Release Consistency Protocol	248
8.4 Performance Results	251
8.5 Summary	264
Bibliography for Part III.....	265
Part IV The Software Package DSPNexpress.....	269
Chapter 9 Overview of the Software Package DSPNexpress.....	271
9.1 Motivation for Developing DSPNexpress	271
9.2 Main Innovative Features of DSPNexpress	274
9.3 Directory Organisation of the Package	278
9.4 The Software Architecture of the Numerical Solution Component.....	280
9.5 Lessons Learned from the Development of DSPNexpress	283
Chapter 10 The Graphical Interface of the Package DSPNexpress	287
10.1 Overall Organisation.....	287
10.2 Organisation and Usage of the Command Line	289
10.2.1 The Add Command	289
10.2.2 The Move Command	292
10.2.3 The Delete Command	292
10.2.4 The Change Command.....	292
10.2.5 The Show Command.....	295
10.2.6 Modification and Duplication of DSPN Subnets.....	298
10.3 Organisation and Usage of the Buttons.....	300
10.3.1 The File Button	300
10.3.2 The Edit Button	302
10.3.3 The Measures Button	303
10.3.4 The Print Button	304
10.3.5 The Validate Button	305
10.3.6 The Solve Button.....	307
10.3.7 The Options Button.....	310
Chapter 11 DSPNexpress Intrinsic.....	312
11.1 Specification Languages for Marking-dependent Parameters of the Building Blocks of DSPN.....	312

11.2 The Profile .DSPNexpress	318
11.3 The Specification Files <Model>.NET and <Model>.DEF	319
Bibliography for Part IV	323
Appendix Primer to Applied Probability	327
Appendix A Fundamentals of Probability Theory	329
A.1 Random Variables	329
A.2 Stochastic Processes	336
Appendix B Common Probability Distributions	355
B.1 Discrete Distributions	355
B.2 Continuous Distributions	370
Bibliography for Appendix	397
Index	399

Part I

Introduction to Performance Modelling

This introductory part states fundamental concepts of computer systems performance evaluation in order to help the reader to get an appreciation for the results presented in Parts II and III of this text. Chapter 1 describes different methods for conducting performance studies of computer systems and states their pros and cons. In particular, we strongly argue for the employment of stochastic modelling to support early design stages of computer systems. Furthermore, we emphasise the extended applicability of numerical methods for the stationary analysis of stochastic models due to the rapid progress in computer hardware technology and the availability of efficient numerical algorithms. In Chapter 2, we illustrate how to employ queuing networks, stochastic Petri nets, and stochastic process algebras, three particular stochastic modelling formalisms, for analysing the performance of a small-scale multiprocessor system.

Part I Introduction to Performance Modelling 1

Chapter 1 Performance Evaluation of Computer Systems..... 3

1.1 Why Performance Modelling?..... 3

1.2 Performance Evaluation Techniques..... 5

1.3 The Process of Performance Modelling..... 12

1.4 Important Professional Organisations..... 14

1.5 Related Technical Journals and Major Conferences..... 15

1.6 Scope and Organisation of this Text..... 19

Chapter 2 Stochastic Modelling Formalisms 23

2.1 Desired Features and Modelling Power 23

2.2 Queuing Networks..... 27

2.3 Generalised Stochastic Petri Nets 31

2.4 Stochastic Process Algebras 34

2.5 Comparative Evaluation of the Considered Modelling Formalisms 42

2.6 Summary..... 46

Bibliography for Part I 47

Performance Evaluation of Computer Systems

1.1 WHY PERFORMANCE MODELLING?

Today's computer systems are rapidly evolving and include typically complex system features. Recent progress in semiconductor technology will soon allow the production of chips for superscalar processors with peak performances comparable to that of the bus-structured multiprocessor systems of ten years ago. As a consequence of this rapid progress, thorough understanding and evaluation of the behaviour of such complex systems is necessary to meet cost and performance requirements through the life cycle of the system. In particular, during the design and implementation phase of a computer system the evaluation of different design alternatives is essential in order to achieve optimal performance for given cost.

Figure 1.1 shows a straightforward design cycle for a computer system using a top-down approach. Starting from a high-level design, e.g., as written in a research proposal, individual system components are designed, and subsequently a detailed design of a novel computer system is derived. Without the support of performance modelling, computer architects have to rely on their experience and intuition for the development of this detailed design. Subsequently, based on the detailed design a software prototype is implemented which mimics the behaviour of the novel computer system and estimates for performance measures are derived using *trace-driven* or *execution-driven simulation*. After the hardware development for a novel computer system has been completed and the system has been installed, *measurements* can be conducted for evaluating system performance. Measurements

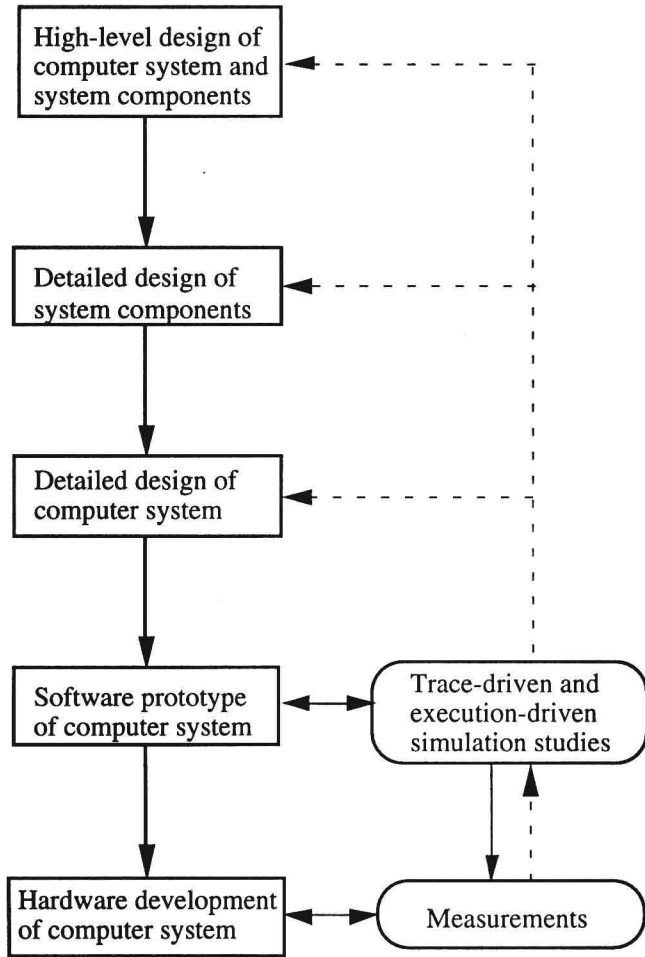


Figure 1.1. Computer system design without support of performance modelling

as well as trace-driven and execution driven simulation belong to the class of deterministic performance evaluation techniques and are briefly discussed in Section 1.2. The dashed lines in Figure 1.1 indicate the possibility of going back to early design stages and redesigning system components due to performance bottlenecks detected by trace-driven and execution-driven simulation studies on the prototype implementation.

If performance is to be considered in the early design stages of a computer system, stochastic modelling must be used because the system is not yet operational, and therefore measurements cannot be conducted. Loosely speaking, a *stochastic model* is an attempt to derive, from the mass of details of the entire system, exactly those which have significant impact on the performance measures of interest. Since a model avoids unnecessary details, it is typically easier to develop and more