

**MODELLING
WITH
GENERALIZED
STOCHASTIC
PETRI NETS**

**M. Ajmone Marsan, G. Balbo, G. Conte,
S. Donatelli and G. Franceschinis**



WILEY

SERIES IN PARALLEL COMPUTING

Modelling with Generalized Stochastic Petri Nets

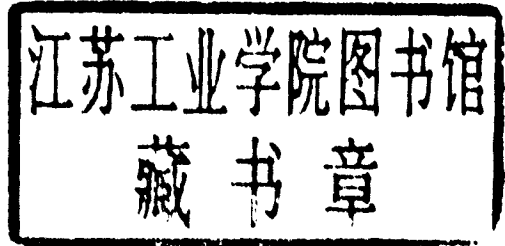
M. Ajmone Marsan, *Politecnico di Torino, Italy*

G. Balbo, *Università di Torino, Italy*

G. Conte, *Università di Parma, Italy*

S. Donatelli, *Università di Torino, Italy*

G. Franceschinis, *Università di Torino, Italy*



JOHN WILEY & SONS

Chichester · New York · Brisbane · Toronto · Singapore

Copyright © 1995 by John Wiley & Sons Ltd.
Baffins Lane, Chichester
West Sussex PO19 1UD, England

National Chichester (01243) 779777
International (+44) 1243 779777

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, xerographic, photographic, recorded, or otherwise, without the prior written permission of the publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system for exclusive use by the purchaser of the publication.

Other Wiley Editorial Offices

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

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

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

John Wiley & Sons (SEA) Pte Ltd, 37 Jalan Pemimpin #05-04,
Block B, Union Industrial Building, Singapore 2057

Library of Congress Cataloging-in-Publication Data

Modelling with generalized stochastic Petri nets / M. Ajmone Marsan
...[et al.].

p. cm. — (Wiley series in parallel computing)

Includes bibliographical references and index.

ISBN 0 471 93059 8

1. Electronic data processing—Distributed processing. 2. Petri
nets. I. Ajmone Marsan, M. II. Series.

QA76.9.D5M64 1995

004:36—dc20

94-20916

CIP

British Library Cataloguing in Publication Data

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

ISBN 0 471 93059 8

Produced from camera-ready copy supplied by the authors using LaTeX
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.

Modelling with Generalized Stochastic Petri Nets



SERIES EDITORS:

R.G. Babb II, *Oregon Graduate Institute, USA*

J.W. de Bakker, *Centrum voor Wiskunde en Informatica, The Netherlands*

M. Hennessy, *University of Sussex, UK*

R. Oldehoeft, *Colorado State University, USA*

D. Simpson, *Brighton University, UK*

Carey (ed.): Parallel Supercomputing: Methods, Algorithms and Applications

de Bakker (ed.): Languages for Parallel Architectures: Design, Semantics, Implementation Models

Axford: Concurrent Programming: Fundamental Techniques for Real-Time and Parallel Software Design

Gelenbe: Multiprocessor Performance

Treleaven (ed.): Parallel Computers: Object-oriented, Functional, Logic

Williams: Programming Models for Parallel Systems

Raynal and Helary: Synchronization and Control of Distributed Systems and Programs

Takeuchi: Parallel Logic Programming

Eliens: DLP — A Language for Distributed Logic Programming: Design, Semantics and Implementation

Kacsuk and Wise (eds): Implementations of Distributed Prolog

Pitas (ed.): Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Networks

Kishinevsky et al: Concurrent Hardware: The Theory and Practice of Self-timed Design

Marsan et al: Modelling with Generalized Stochastic Petri Nets

Preface

The goal of this book is to introduce readers to the use of Generalized Stochastic Petri Nets (GSPNs) for the performance analysis of distributed systems of different natures. We hope that a unified presentation of GSPN theory, together with a set of illustrative examples from different application fields, will further extend the success that GSPNs have obtained in recent years as a modelling paradigm for the quantitative analysis of distributed systems.

The book is addressed to beginners, rather than experts in the field of Petri Net applications to performance analysis, and aims to be a useful tool for a first approach to the GSPN field, and a support for a graduate course on modelling and performance evaluation of computer and communication systems.

For this reason, the book is written in an informal style, trying to always explain concepts in plain language before giving a formal definition (often even omitting formal definitions that are not strictly necessary), and including as many examples as possible, to visualize the concepts being explained.

Before a short outlook on the book's content, some historical remarks and acknowledgments are necessary.

The origin of GSPNs goes back to the early 80s, and is due to the cooperation of three of the authors, who participated in the research group that was organized in the city of Torino to join a national research project on the design, implementation and performance evaluation of multiprocessor systems.

Gianni Conte was the initiator of the group: he was the one with the most experience in multiprocessor architectures, and he had known about the possibility of adopting the Petri net paradigm for the description and analysis of such systems from the papers of L. Dadda. Gianfranco Balbo was the one with the most experience in the performance analysis field: he had just obtained his PhD in Computer Science from Purdue University with a thesis on queuing networks and their applications to the performance analysis of computer systems. Marco Ajmone Marsan had recently returned from UCLA, where he had met Mike Molloy, and had learnt about stochastic Petri nets (SPNs), and their usefulness. Actually, the idea of introducing immediate

transitions into the SPN paradigm was discussed several times along the corridors of Boelter Hall at UCLA between the two.

The original description of GSPNs was submitted to the 1982 European Workshop on Petri Net Applications and Theory in Varenna, but was rejected with the longest list of negative comments that we ever saw! In fact, the original version of the theory was not quite clean, but after a number of changes, a revised GSPN presentation was submitted to SIGMETRICS'83, and selected as one of the two best papers of the conference, deserving publication in the ACM Transactions on Computer Systems. The official birth date for GSPNs is thus the May 1984 issue of the ACM TOCS.

The first students to write a thesis on GSPN theory and applications were Giovanni Chiola at the Electronics Department of Politecnico di Torino and Gianfranco Ciardo at the Computer Science Department of the University of Torino. They were also the first to implement the software for a GSPN analysis tool. G. Ciardo, after his graduation, went to the USA for a PhD at Duke University with Professor Kishor S. Trivedi. G. Chiola, instead, remained in Torino, first at Politecnico, then at the University, where he made some very significant contributions to the development of GSPN theory, he applied GSPNs to the performance analysis of many different types of systems, and he developed the GreatSPN software tool for the analysis of GSPN models. We all deeply regret the choice of Giovanni to participate only in the early discussions about the contents of this book. Many of the ideas and results presented in the chapters on GSPN theory are his own, and he surely is today one of the most active and creative researchers in the field of GSPNs.

The presence of G. Ciardo at Duke helped make the GSPN paradigm known, and one of the first groups in the USA to adopt GSPNs was indeed the one led by Professor Trivedi. In particular, when the idea arose of organizing a first workshop to bring together the researchers active in the development and application of GSPNs and similar modelling paradigms, the support of Professor Trivedi was the key factor to give us sufficient confidence to organize the International Workshop on Timed Petri Nets in July 1985 in Torino. The name of the workshop was originally meant to be the International Workshop on Stochastic Petri Nets, but the fear of characterizing the topic too narrowly suggested the actual name. However, given the success of the Workshop in Torino, the subsequent editions adopted a different name: International Workshops on Petri Nets and Performance Models were held in Madison, Wisconsin, in 1987; in Kyoto, Japan, in 1989; in Melbourne, Australia, in 1991; in Toulouse, France, in 1993; and the next one will be hosted by Duke University in Durham, North Carolina, in October 1995.

Susanna Donatelli and Giuliana Franceschinis were the first students to choose GSPN theory and applications as a subject for their doctoral theses at the Computer Science Department of the University of Torino. S. Donatelli studied techniques for

the analysis of large GSPNs. G. Franceschinis developed coloured extensions of the GSPN paradigm, to exploit symmetries in the GSPN model definition and analysis. After the completion of their doctoral studies, both Susanna and Giuliana continued their research on GSPNs at the Computer Science Department of the University of Torino.

The continuing success of GSPNs, and the growing number of researchers interested in them, suggested the collection of the main results of more than ten years of research in this book, with the intent of providing newcomers to the GSPN field with a useful tool for their first approach with the formalism.

The book is divided into two parts. Part 1 comprises Chapters 1–6, and summarizes the main results in GSPN theory. Part 2 is made up of Chapters 7–11, which present examples of application of the GSPN methodology.

Chapter 1 contains an informal introduction to Petri nets, mainly intended for those readers who have had no previous experience with them. This chapter, however, is also useful for a first presentation of some Petri net models that will be used in later chapters.

Chapter 2 provides a formal definition of Petri nets, Petri net systems, and Petri net models, and contains a summary of some of the definitions and results of classical Petri net theory that are most useful for the development of the theory in the rest of the book.

Chapter 3 contains a first discussion about the introduction of temporal concepts into Petri nets, and provides an intuitive justification for the importance of priority in timed Petri nets. This discussion naturally leads to the problem of priority in Petri nets, which is addressed in Chapter 4, where formal definitions and theoretical results for Petri nets with priority are presented, extending the results of Chapter 2 to this environment.

After these four introductory chapters, the reader is ready to enter the GSPN arena. Chapter 5 provides the definitions of SPN (stochastic Petri net) and GSPN, as well as a description of the dynamic behaviour of these models. Chapter 6 illustrates the techniques for the analysis of SPN and GSPN models. Both chapters contain illustrative examples, to help the reader understand the concepts that are explained in the text.

The second part of the book is entirely devoted to application examples, to show how the GSPN methodology can be used in different fields.

Chapter 7 shows how GSPN models can account for activities with generally distributed durations, using as an example the classical central server model. In particular, the chapter shows how it is possible to develop a central server model in which the execution times are characterized with an Erlang-2 distribution, rather than with the usual negative exponential distribution adopted in SPNs and GSPNs.

The possibility of generalizing the approach to any phase-type distribution is also discussed.

Chapter 8 provides examples of the application of the GSPN approach to the performance analysis of flexible manufacturing systems. A model of the classical “Kanban” system, and two models of a push production system fed by either a continuous transport system or an automated guided vehicle are presented.

Chapter 9 illustrates the construction of GSPN models of polling systems, which find a variety of applications in the fields of telecommunications, manufacturing, and computer engineering. Rather than looking at the classical cyclic polling systems, we study *random* polling systems, since this choice allows the discussion of the approaches that can often be followed for the construction of *compact* GSPN models, with the advantage of a significant reduction in the cost of the model solution. An example of the reduction process that transforms a GSPN model into an SPN model is also contained in this chapter.

Chapter 10 provides an example of how GSPNs can be used to validate and evaluate concurrent programs. An automatic translation from the software to a GSPN is illustrated, and the techniques for the identification of faults in the software are outlined.

Chapter 11 provides examples of the application of the GSPN approach to the analysis of massively parallel computer architectures.

All the numerical results presented in the examples are obtained with the GreatSPN software package [Chi91a].

The book is completed by two appendices. Appendix A collects some of the results of the theory of stochastic processes that are necessary in the development of GSPN theory. Appendix B contains a glossary with most of the notation used throughout the book.

It should be noted that prerequisites for the study of GSPNs are the knowledge of some simple results of the theory of stochastic processes, and of the theory of Petri nets. Whereas the latter are presented in Chapter 2, the former are discussed in Appendix A. The reason for giving more space to the theory of Petri nets in Chapter 2, and for extracting the presentation of the basic results of the theory of stochastic processes from the main body of the book, lies in the fact that we imagine that a majority of the readers may already be familiar with the material in Appendix A, and in the wide variety of good books devoted to the theory of stochastic processes.

Before leaving the reader to the technical discussions, we wish to acknowledge the support and patience of Gaynor Redvers-Mutton, our Editor at John Wiley in England, and of her staff, and to thank all of our colleagues who read the early versions of the manuscript and provided useful comments and suggestion; in particular we wish to thank Giovanni Chiola and Gianfranco Ciardo for their very constructive remarks.

We also wish to thank L. Brino for providing some of the figures. We acknowledge the financial support of several agencies, which over the years have sponsored our research in the field of GSPNs. In particular, we are grateful to the Italian National Research Council, to the Italian Ministry for University and Research, and to the European Economic Community for their support.

Torino, Italy

June 1995

M.A.M.

G.B.

G.C.

S.D.

G.F.

Contents

Preface	xvii
1 INTRODUCTION	1
1.1 Shared Resources	4
1.2 Fork and Join	8
1.3 Kanban Process	10
1.4 Token Ring LAN	13
2 PETRI NETS AND THEIR PROPERTIES	15
2.1 Petri Net Models	15
2.2 Models, Systems, and Nets	18
2.3 System Dynamics	19
2.3.1 Enabling and firing rules	20
2.3.2 Reachability set and reachability graph	22
2.3.3 Modelling issues	24
2.3.4 Typical situations in the PN system evolution	26
2.4 Properties of Petri Nets	32
2.5 Structural Analysis Techniques	35
2.5.1 Linear algebraic techniques	35
2.5.2 Graph-based structural techniques	42
2.6 State Space Analysis Techniques	46
3 TIME IN PETRI NETS	49
3.1 The Motivations for Timing	49
3.2 Timed Transitions	51
3.2.1 Immediate transitions	53
3.2.2 Two examples	54
3.3 Parallelism and Conflict	56
3.4 Memory	59
3.4.1 An example with the enabling memory policy	61

3.4.2	An example with the age memory policy	62
3.5	Multiple Enabling	63
3.6	A Timed PN Model Example	66
3.7	Performance Measures	68
4	PETRI NETS WITH PRIORITY	71
4.1	Definition and Dynamics	71
4.2	Conflicts, Confusion and Priority	75
4.3	Properties of Priority PN Models	79
4.4	Structural Analysis Techniques	82
4.4.1	Linear algebraic techniques	82
4.4.2	Graph based analysis	83
4.5	Reachability Analysis Techniques	85
5	GSPN BASICS	91
5.1	Exponential Distributions for Transition Delays	91
5.2	Mixing Exponentially Distributed and Null Delays	95
5.3	Some Extensions	98
5.4	The Definition of a GSPN Model	100
5.5	An Example of a GSPN Model	101
5.6	Some Fine Points	104
6	ANALYSIS OF GSPN MODELS	107
6.1	The Stochastic Process Associated with a SPN	107
6.1.1	SPN performance indices	109
6.1.2	An example SPN system	110
6.2	The Stochastic Process Associated with a GSPN	115
6.2.1	Marking dependency	120
6.2.2	An example GSPN system	122
6.3	Numerical Solution of GSPN Systems	125
6.3.1	Evaluation of the steady-state probability distribution for the example GSPN system	133
6.3.2	Performance analysis of the example GSPN	136
6.4	Reducing GSPNs to SPNs	136
6.5	Simulation of GSPN Systems	140
6.5.1	Simulation of the example GSPN system	146
7	GSPN REPRESENTATION OF PHASE-TYPE DISTRIBUTIONS	149
7.1	The Central Server Model Example	149

7.1.1	CPU interruptions	152
7.1.2	CPU memory policies	156
7.2	Phase-Type Distributions	157
7.2.1	Erlang distributions	158
7.2.2	Numerical evaluation	163
7.3	General Phase-Type Distributions	168
8	MODELLING FLEXIBLE MANUFACTURING SYSTEMS	171
8.1	Flexible Manufacturing Systems	172
8.2	A Kanban System	173
8.2.1	Structural analysis of the GSPN model of the Kanban system	176
8.2.2	Performance analysis of the Kanban system	177
8.3	Push Production Systems	180
8.3.1	Structural analysis of the push production systems	187
8.3.2	Performance analysis of the system with AGV transport	190
9	COMPACT MODELS OF RANDOM POLLING SYSTEMS	199
9.1	Polling Systems	199
9.2	Modelling Random Polling the Easy Way	203
9.2.1	The first complete random polling model	205
9.3	Beginning to Exploit Symmetries	208
9.3.1	Walking before choosing?	209
9.3.2	Number of states and limitations of the model	211
9.4	Abstracting from the Queue Identifiers	212
9.4.1	The abstract GSPN model	213
9.4.2	Walking before choosing!	215
9.4.3	Number of states and limitations of the model	216
9.5	From GSPNs to SPNs	217
10	MODELLING AND ANALYSIS OF CONCURRENT PROGRAMS	225
10.1	Introduction to the Static Analysis of Concurrent Programs	225
10.2	The Automatic Translation from Programs to GSPNs	227
10.2.1	Languages of interest	227
10.2.2	The translation procedure	229
10.3	An Example of Translation	233
10.4	Modelling Control Variables to Reduce Nonfaults	239
10.5	Stochastic Analysis	241
11	MODELS OF CONCURRENT ARCHITECTURES	245

11.1	Concurrent Architectures	245
11.2	Single Processor Model	246
11.3	Single Processor Model with Multitasking	249
11.4	Models of Mesh Structures	252
11.4.1	Mesh models without multitasking	252
11.4.2	Mesh models with multitasking	254
Appendix A STOCHASTIC PROCESS FUNDAMENTALS		259
A.1	Basic Definitions	259
A.2	Markov Processes	260
A.3	Discrete-Time Markov Chains	263
A.3.1	Steady-state distribution	264
A.3.2	Matrix formulation	266
A.3.3	Example	267
A.4	Continuous-Time Markov Chains	269
A.4.1	Steady-state distribution	270
A.4.2	Matrix formulation	271
A.4.3	Sojourn and recurrence times	272
A.4.4	The embedded Markov chain	272
A.4.5	Example	274
A.5	Aggregation of States in Markov Chains	277
A.5.1	Aggregation in discrete-time Markov chains	277
A.5.2	Aggregation in continuous-time Markov chains	280
A.6	Semi-Markov Processes	281
A.7	Semi-Markov Reward Processes	283
Appendix B GLOSSARY		285
References		289
Index		295

List of Figures

1	PN description of a switch	3
2	PN description of the user (a), and of the resource (b)	5
3	PN description of the user access to a resource	6
4	PN description of two users accessing a shared resource	7
5	Reduced PN description of two users accessing a shared resource	8
6	PN description of a system with N users and M resources	9
7	PN description of a fork and join behaviour	9
8	PN description of a parallel system	11
9	PN description of a Kanban system	12
10	PN description of a token ring LAN with three users	14
11	A Petri net model	17
12	The enabling and firing rules	21
13	The reachability set of the readers & writers PN system obtained from the PN model in Fig. 11 by letting $K = 2$	23
14	The reachability graph of the readers & writers PN system obtained from the PN model in Fig. 11 by letting $K = 2$	24
15	Implementation of tests	26
16	Implementation of a test	27
17	Implementation of an or	27
18	Modelling conflict	28
19	Two cases of asymmetric conflict	29
20	Modelling concurrency	30
21	Confusion	31
22	Petri nets with equal incidence matrix	36
23	Visualization of the two minimal P-semiflows of the readers & writers example	40
24	Visualization of the two T-semiflows of the readers & writers example	42
25	An example of HME relation: $t_l HME t_m$	45
26	An example of MME relation: $t_l MME t_m$	45

27	A timed transition	51
28	A timed PN system	52
29	Pictorial representation of a timed execution	53
30	Timed (a) and immediate (b) transitions	54
31	Timed version of the PN system in Fig. 5	55
32	Subnet modelling the migration of tokens from p_a to p_b when a synchronizing event takes place	56
33	Non-conflicting concurrent transitions	57
34	Conflicting transitions	57
35	Using immediate transitions to separate conflict resolution from transition timing specification	58
36	Interrupting an activity with an immediate transition	59
37	Timed PN system with an example of enabling memory policy	61
38	Timed PN system with age memory policy	63
39	Timed transition with enabling degree 3	64
40	Firing epochs corresponding to the three different timing semantics	65
41	Queueing network model of a timesharing system	67
42	Timed PN model of a timesharing system	67
43	Timed PN model of the readers & writers system	68
44	A modified readers & writers model	73
45	An example of indirect conflict	77
46	An example of boundedness due to priority	80
47	An example of boundedness due to inhibitor arcs	81
48	Influence of the priority on the maximum enabling degree of transitions	81
49	An example of modified P-invariant due to priority	83
50	An example of <i>IME</i> relation	85
51	Reachability set of the readers & writers PN system with priority	86
52	Reachability graph of the readers & writers PN system with priority	87
53	1-reduced reachability graph of the readers & writers PN system with priority	88
54	0-reduced reachability graph of the readers & writers PN system with priority	89
55	SPN description of a simple parallel system	94
56	GSPN description of a simple parallel system	99
57	The example GSPN model: a modification of the readers & writers model	102
58	A conflict set comprising a timed and an immediate transition	105

59	The SPN description of shared memory system	111
60	The reachability graph of the SPN system in Fig. 59	113
61	The state transition rate diagram of the Markov chain associated with the SPN in Fig. 59	113
62	The Infinitesimal Generator of the Markov Chain in Fig. 61	114
63	An example of ECS	117
64	Comparison of two simple confused GSPN systems	119
65	Examples of complex marking dependency situations	122
66	The GSPN description of a parallel system	123
67	A GSPN system with multiple paths between tangible markings . . .	128
68	A GSPN system with several multiple paths between tangible markings	130
69	Removing immediate transitions: the free-choice case	137
70	Removing immediate transitions: the non-free-choice case	139
71	First reduction step for the GSPN of Fig. 66	141
72	Second reduction step for the GSPN of Fig. 66	142
73	Third reduction step for the GSPN of Fig. 66	143
74	The GSPN system representing a Central Server Model	150
75	The GSPN representation of a CPU interruption cycle	153
76	The GSPN system representing a Central Server Model with CPU interruptions	154
77	Alternative GSPN representation of a Central Server Model with CPU interruptions	157
78	The GSPN representation of an Erlang-3 server	159
79	The GSPN model of a conflict between Erlang-3 and exponential servers	160
80	The interruptible GSPN model of an Erlang-3 server	160
81	Age memory conflict	161
82	Enabling memory conflict	162
83	The GSPN representation of a Central Server Model with CPU interruptions and with an Erlang-3 service time distribution of regular customers with Age Memory policy	164
84	The GSPN representation of a Central Server Model with CPU interruptions and with an Erlang-3 service time distribution of regular customers with Enabling Memory policy	165
85	A Kanban cell and the parts and cards that flow into and out of it . .	174
86	GSPN model of a single Kanban cell	174
87	GSPN model of an n -cell sequential Kanban system	175