

NETWORKS & TELECOMMUNICATIONS SERIES

NEW GENERATION NETWORKS SET



Volume 6

Enhancing Stochastic Petri Nets with Reconfigurability

*Modeling, Analysis
and Performance Evaluation*

**Samir Tigane, Laid Kahloul
and Abdelhamid Mellouk**

ISTE

WILEY

Enhancing Stochastic Petri Nets with Reconfigurability

*To my beloved family, my rock,
and to my beloved late mother, my lighthouse,
who lives in me and whom I miss very much*

Abdelhamid Mellouk

*To my beloved family. To everyone who has
contributed in any way – directly or indirectly*

Samir Tigane

To my family, with all my love and gratitude

Laid Kahloul

New Generation Networks Set

coordinated by
Abdelhamid Mellouk

Volume 6

Enhancing Stochastic Petri Nets with Reconfigurability

*Modeling, Analysis and
Performance Evaluation*

Samir Tigane
Laid Kahloul
Abdelhamid Mellouk

ISTE

WILEY

First published 2025 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

© ISTE Ltd 2025

The rights of Samir Tigane, Laid Kahloul and Abdelhamid Mellouk to be identified as the authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s), contributor(s) or editor(s) and do not necessarily reflect the views of ISTE Group.

Library of Congress Control Number: 2025935041

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-83669-062-7

Contents

Preface	ix
General Introduction	xi
Part 1. State of the Art	1
Chapter 1. From Petri Nets to Stochastic Petri Nets	3
1.1. Introduction	3
1.2. Modeling with Petri nets	4
1.3. Petri net structure	6
1.4. Dynamic behavior of Petri nets	7
1.5. Petri net analysis	11
1.5.1. Petri net properties	11
1.5.2. Temporal logic	14
1.5.3. Analysis methods	18
1.6. Stochastic Petri nets	19
1.6.1. Stochastic process	21
1.6.2. Markov process	21
1.6.3. Stochastic Petri nets having exponential law	22
1.6.4. Quantitative properties	26
1.7. Generalized stochastic Petri nets	27
1.7.1. Embedded Markov Chain	29
1.8. Conclusion	32
Chapter 2. Reconfiguration Aspects in Petri Nets	33
2.1. Introduction	33
2.2. Graph Transformation Systems	34

2.3. Double-pushout approach for Petri nets	35
2.4. Net rewriting systems	38
2.5. Self-modifying nets	41
2.6. Reconfigurable Petri nets	43
2.7. Improved net rewriting systems	47
2.8. Other extensions	48
2.9. Trade-off between expressiveness and calculability in PN-based reconfigurable formalisms	49
2.10. Conclusion	51

Part 2. Orientation 1 53

Chapter 3. Rewritable Topology in Generalized Stochastic Petri Nets . . . 55

3.1. Introduction	55
3.2. GSPNs with rewritable topology	56
3.2.1. Formal definition	58
3.3. Proofs	60
3.4. Illustrative example	61
3.5. Stochastic reward nets	67
3.6. Configuration-dependent stochastic reward nets	68
3.7. Transformation of CD-SRNs into basic SRNs	69
3.8. Proofs	72
3.9. Illustrative example	73
3.10. Conclusion	79

Chapter 4. Generalized Stochastic Petri Nets with Dynamic Structure . . . 81

4.1. Introduction	81
4.2. Dynamic GSPNs	83
4.2.1. Formal definition	83
4.3. D-GSPN transformation towards GSPNs	85
4.4. Qualitative/quantitative analysis of D-GSPNs	89
4.5. Proofs	90
4.6. Illustrative example	92
4.7. Generalized stochastic Petri nets with inhibitor and reset arcs	98
4.8. Improved D-GSPNs under infinite-server semantics	99
4.9. Unfolding ID-GSPNs into GSPNs	105
4.10. Running examples	110
4.11. Conclusion	119

Part 3. Orientation 2	121
 Chapter 5. Reconfigurable Generalized Stochastic Petri Nets	123
5.1. Introduction	123
5.2. Reconfigurable generalized stochastic Petri nets	125
5.2.1. Definition of RecGSPNs	125
5.2.2. Properties preserving nets	130
5.3. Preservation of properties in RecGSPNs	131
5.3.1. Preservation of LBR, home state and deadlock-free	131
5.3.2. Preservation of linear temporal properties	136
5.4. Quantitative analysis	141
5.5. Using RecGSPNs in practice	144
5.6. Conclusion	149
 Part 4. Evaluation, Discussion and Conclusion	151
 Chapter 6. Evaluation and Discussion	153
6.1. Introduction	153
6.2. Qualitative aspects	153
6.3. Quantitative aspects	156
6.3.1. Factor 1: model size	157
6.3.2. Factor 2: Markov chain spatial complexity	159
6.3.3. Factor 3: Markov chain time complexity	161
6.4. Conclusion	163
 Conclusion	165
 References	169
 Index	177

Preface

The evolution of computing systems has brought forth a new era of complexity, where adaptability and resilience are crucial to ensuring reliability and efficiency. Traditional modeling techniques often struggle to capture the dynamic nature of modern discrete-event systems, which require structural flexibility/reconfigurability at runtime. This book addresses these challenges by introducing reconfigurable stochastic Petri nets (RSPNs) – a powerful extended formalism of classical Petri nets designed to model and analyze dynamic-structure systems.

Our motivation for this work stems from the increasing need for formal verification techniques that can accommodate reconfigurability without compromising analytical rigor. Thus, by integrating graph transformation systems with generalized stochastic Petri nets (GSPNs), we provide a comprehensive framework that supports both *qualitative* as well as *quantitative* analysis. The proposed approach in this book enables system designers to model and verify critical properties such as *liveness* and *boundedness* and to evaluate *performance metrics* for system reconfigurations.

This book is structured to provide both theoretical foundations and practical applications. The first part presents an in-depth discussion on Petri nets, their extensions, and reconfigurabilities aspects. The second part explores new formalisms such as dynamic GSPNs (D-GSPNs) and reconfigurable GSPNs (RecGSPNs), offering methodologies for modeling as well as verifying reconfigurable systems. The final sections focus on comparative evaluations, demonstrating the efficiency and effectiveness of these approaches in various domains, including network security, manufacturing systems, and distributed computing.

We hope that this book serves as a valuable resource for researchers, practitioners, and students seeking to advance the field of formal methods for dynamic-structure systems. By bridging the gap between theoretical advancements and real-world applications, we aspire to contribute to the ongoing development of robust and adaptive computing models.

The completion of this book would not have been possible without the support and contributions of many individuals and institutions. First and foremost, we would like to express our deepest gratitude to all our colleagues and collaborators, whose insightful discussions and valuable feedback have greatly enriched the content of this work. Their expertise and dedication have been instrumental in shaping the ideas presented in this book.

We are also profoundly thankful to all researchers of LINFI Laboratory in Biskra University as well as to Dr. Soheyb Baarir in LIP6 Laboratory in France, for the deep discussions and important guidance and encouragement, which played a crucial role in this academic journey.

A special note of appreciation goes to our families and friends, whose patience and understanding provided us with the motivation and balance needed to complete this project. Their continuous encouragement has been a source of strength throughout this endeavor.

To everyone who has contributed in any way – directly or indirectly – thank you. Your support has been invaluable.

March 2025

General Introduction

In this introduction, we start by presenting the background of this book, namely the formal modeling and verification based on dynamic-structure stochastic Petri nets. Then, we focus on the motivations of this work, specify the problem and the objectives and highlight our contributions. Finally, we end with the description of the manuscript organization.

I.1. Background

The major advancements in computing power, connectivity, sensors, storage capacity and software development have motivated companies as well as individuals to adopt and integrate IT solutions into their daily tasks (from a small device at the house to a gigantic infrastructure). As the list of advantages and benefits resulting from this orientation continues to grow, so do the risks of failure and malfunctioning that may threaten companies as well as individuals. Therefore, it is absolutely mandatory to ensure the proper functioning of computer systems according to customers' and designers' expectations. This concern had been identified since the late 1960s that marked the birth of software engineering. One of the main goals of the latter is enabling developers to implement complex systems that work properly. In this regard, several approaches have emerged to meet this crucial requirement in the various stages of the software life cycle (Woodcock et al. 2009).

Approaches in which the syntax, semantics and manipulation rules of specification language are explicitly defined by mathematics are called *formal approaches*. These approaches include Petri nets (Murata 1989), sequential process communication (SPC) (Brookes et al. 1984), LOTOS (Bolognesi and Brinksma 1987), B-method (Abrial 2005), etc. Actually, formal approaches allow a complete verification of the whole system behavior and prove the presence of certain desired properties for all possible inputs. By formal methods, we can write a formal specification of a system on which different properties can be proved, and thereafter we can mathematically

prove that a system implementation meets this specification (Craig et al. 1993; Haxthausen 2010; Garavel and Graf 2013; Rodhe and Karresand 2015). However, the use of formal methods does not guarantee a priori the accuracy of developed systems. Indeed, their use enhances our understanding of a system under construction while revealing its shortcomings, inconsistencies and ambiguities that might otherwise go unnoticed (Clarke and Wing 1996).

Among the most widespread formalisms, we find Petri nets (PNs) (Peterson 1977; Murata 1989). They are characterized by three major advantages (Peterson 1981; Kamath and Viswanadham 1986):

1) **Modeling level:** they have a powerful mathematical foundation, as well as an intuitive graphical representation. The graphical representation gives a flat view to PN models, making it possible to have simple and very explicit models. Also, their graphic modeling enables easy visualization of complex systems.

2) **Verification level:** their mathematical foundation is at the origin of all the analysis techniques that were proposed in order to verify the modeled systems. Indeed, they dispose of a well-developed qualitative/quantitative analysis panoply.

3) **Coupling modeling and verification:** they offer a careful balance of modeling and decision power. In fact, Petri nets have been used in the modeling of a wide variety of systems. As for their decision power, the reachability problem is decidable in Petri nets (note that most problems can be converted into reachability problems).

The model, in its origin proposed by “Adam Petri” in Petri (1962), that was initially concerned with describing the causal relationships between events that can occur in a computer system, has undergone significant evolution and adaptation to meet several requirements imposed by the appearance of new complicated systems. The most notable extensions can be found in four main categories:

- Colored PNs (Jensen 2013): each token becomes rather a distinguishable value from other tokens. The weights on the arcs are no longer constants, but rather mathematical functions that can be complicated. This model makes it possible to have models of reasonable size for complicated systems.

- Temporal PNs (Ramchandani 1973): the time introduced in PNs allows us to put explicit constraints on the dynamics of the model which reflect the real temporal constraints imposed on the system.

- Stochastic PNs (Marsan et al. 1994): stochastic PNs are a response to another missing realistic aspect in previous models, which is the aspect of hazard, randomness and non-absolute events. The random events in their arrivals will be explicitly considered and modeled, allowing the model to get closer to the real system and thus to have a good representation of the studied system.

– Reconfigurable PNs (Llorens and Oliver 2004b; Ehrig and Padberg 2004; Padberg and Kahloul 2018): this last category groups formalisms allowing the modeling of structure flexibility.

The purpose of this last variant is to provide a formal model for dynamic-structure systems, for example, flexible manufacturing systems (FMSs) (Buzacott and Shanthikumar 1980), reconfigurable manufacturing systems (RMSs) (Koren et al. 1999), production in cloud (Xu 2012), Industry 4.0 (Lasi et al. 2014), etc. In fact, many discrete event systems (DESs) are becoming increasingly complex, structurally dynamic and variably interconnected. These systems are designed to be able to change their structure and/or topology, at run time, by adding/removing interconnections, objects or even subsystems, to accommodate new circumstances/requirements. Ongoing studies on this class of systems focus on their key feature, namely, the reconfigurability (Brettel et al. 2014) that must occur at run time (i.e. dynamic reconfigurability) (Jackson et al. 2016). Dynamic reconfigurability is a critical activity that influences the performance, security and cost of such systems. To overcome the previous challenges, the designer must dispose of a rigorous approach and a set of appropriate tools.

The use of PNs in the study of such systems attracts many researchers (Silva and Valette 1990; Marsan et al. 1994; Recalde et al. 2004; Chen et al. 2017; Long et al. 2017; Latorre-Biel et al. 2018; Liu et al. 2018; You et al. 2018). In the literature, many classes of PNs have been proposed and applied to specify/verify reconfigurable systems. The chosen PN class is often motivated by the aspects to be specified and the properties to be verified. In fact, we can distinguish three classes of work: that uses basic PNs, that uses temporal or stochastic PNs and, finally, that applies reconfigurable PNs. Stochastic PNs (SPNs) and generalized SPNs (GSPNs) represent an extension of PNs (Marsan et al. 1994) used to model and evaluate stochastic systems. These formalisms allow the analysis of performance metrics such as productivity, energy consumption and machine utilization. Marsan et al. (1994) strongly emphasize the importance of GSPNs and SPNs as versatile design tools that fit well with the behavior of DESs at different stages of development (Long et al. 2015; Čapkovič 2017; Simon et al. 2018; Latorre-Biel et al. 2018). Although PNs (low or high) are a powerful and expressive tool, they are unable to specify/verify, in a natural way, advanced dynamic-structure systems (Capra and Camilli 2018). Systems supporting volatile environments, continuous variations and reconfigurable structures are expected to be extremely complex (Chryssolouris et al. 2013). The design of such systems is an increasingly complex and omnipresent challenge. Therefore, designers must dispose of the necessary approaches, models and tools to handle this complexity (Möller 2016). To overcome this issue, researchers introduce dynamic structures into PNs, thus expanding the standard formalism (Padberg and Kahloul 2018).

Rule-based graph transformations (Ehrig and Padberg 2004) offer a mathematically based graphical framework for modeling the reconfigurations in

PN structures. Nevertheless, increasing the modeling power of a formalism decreases its decision power. Therefore, extensions proposed in the literature introducing reconfigurability to PNs try to find a compromise between the modeling and the verification levels. From this perspective, we can distinguish three main directions. On the one hand, researchers develop pre-processing techniques that encode, unfold or compile graphs and transformation rules into existing formalisms in order to exploit the panoply of their tools (Llorens and Oliver 2004b; Capra and Camilli 2018; Camilli et al. 2018). Although they can naturally model the reconfigurations, these approaches did not increase the modeling power compared to existing ones, since they depend on target formalisms expressiveness and in particular do not allow modeling infinite graphs (Rensink et al. 2004). For instance, classical model-checkers (Baier and Katoen 2008) use a fixed number of propositions, which prevents the modeling of infinite-structure systems (Rensink 2008). On the other hand, some techniques execute graph transformation systems and compute the reachability graph; nevertheless, an upper artificial threshold is still needed (Kastenberg and Rensink 2006). To mitigate this issue, some approaches compute either under-approximations of a system's behavior so that any property that holds in an under-approximated model is satisfied in its original system, or over-approximations including all system behaviors, and possibly more (da Costa and Ribeiro 2012). Nevertheless, a property that does not hold in an under-estimated model may hold in its original system and a property that holds in an over-estimated model may not be satisfied in its original system (Baldan et al. 2008). A promising approach described in Li et al. (2009b), called improved net rewriting system (INRS), preserves particular properties, namely, liveness, boundedness and reversibility of PNs after each reconfiguration. These properties are therefore decidable regardless of the number of obtained configurations. However, INRSs are limited to (i) ordinary, live, bounded and reversible PNs and (ii) particular forms of reconfiguration.

I.2. Motivation and Objectives

Exception for few work (Meer and Dusterhoft 1997) and (Capra 2017), reconfigurability in either SPNs or GSPNs has not received much attention. Existing approaches in the literature often focus on the performance evaluation of reconfigurable systems using SPNs (which are not reconfigurable formalisms) or on reconfigurability simulation and verification using reconfigurable PNs (which do not consider stochastic aspects). Our goal in this book is to introduce reconfigurability into the well-known GSPNs formalism using graph transformation systems. The objective of this book is doubled:

- Integration of the dynamic aspect in GSPNs: this requires a formal definition of a new formalism combining the stochastic and the dynamic aspects in a single formalism.

– Study of this hybridization consequences on GSPNs analysis: introducing reconfigurability into GSPNs requires adapting the classical algorithms towards the new formalisms and/or proposing new analysis techniques.

I.3. Contributions

The present book falls within the field of reconfigurable system modeling and verification based on GSPNs, hence the need for a GSPNs-based formalism supporting dynamic structures. However, increasing the modeling power of a formalism involves straightforwardly increasing the verification complexity. Faced with these challenges, we have defined five approaches – incrementally developed – for the modeling and the verification of dynamic-structure GSPNs, each of which has its advantages, limits and orientation.

I.3.1. Orientation 1: Transforming dynamic GSPNs into GSPNs

In the first place, we were interested in the modeling and the verification of reconfigurable systems using (non-stochastic) reconfigurable Petri nets. With this in mind, our first intention was to propose an extension for one of these reconfiguration approaches to the stochastic ones. Our primary contributions in this direction were reconfigurable SPNs (R-SPNs) (Tigane et al. 2017b), GSPNs with rewritable topology (Tigane et al. 2017a) (GSPNs-RT) and configuration-dependent stochastic reward nets (Tigane et al. 2021) (CD-SRNs), extending formalism presented in Llorens and Oliver (2004b), to cope with reconfigurability in SPNs, GSPNs and stochastic reward nets (SRNs), respectively. However, these three proposed formalisms allow only dynamic topology, i.e. sets of places and transitions cannot be changed. By fixing both sets of places and transitions, we can transform SPNs, GSPNs and SRNs having dynamic topologies into basic SPNs, GSPNs and SRNs, respectively, by encompassing all topologies in one model. In the latter, the switching between configurations and appearing/disappearing/reappearing of topologies are modeled via the token game. The transformation into basic SPNs, GSPNs or SRNs straightforwardly allows exploiting the methods and techniques proposed in the literature for SPNs, GSPNs and/or SRNs to verify the properties of dynamic ones. However, allowing only dynamic topologies severely limits the modeling power of these formalisms.

To overcome this shortcoming, we propose a new formalism, called dynamic generalized stochastic Petri nets (D-GSPNs), that allows us to model dynamic sets of places and transitions, as well as keeping the possibility to transform D-GSPNs into GSPNs for verification purposes. The obtained GSPNs preserve the stochastic behaviors of dynamic GSPNs, allowing the use of the panoply of verification methods and tools proposed for GSPNs in D-GSPNs analysis. Moreover, arcs in D-GSPNs are basic ones, i.e. there is no read, reset, or inhibitor arcs (Dufourd et al. 1998),

which increases their decision power. However, D-GSPNs only allow the modeling and verification of nets whose transitions follow single-server semantics. Thus, tasks that can provide n services at a time cannot be modeled. To overcome D-GSPNs limits, we extend D-GSPNs to a new formalism called improved D-GSPNs (ID-GSPNs), in order to model and verify dynamic nets whose transitions follow single-, infinite-, or multiple-server semantics. A new algorithm that transforms ID-GSPNs into basic equivalent GSPNs preserving their stochastic behaviors is also presented. Moreover, the proposed algorithm yields more compact models than those obtained by the unfolding algorithm proposed in D-GSPNs.

1.3.2. Orientation 2: Preserving properties in reconfigurable generalized stochastic Petri nets (RecGSPNs)

Although D-GSPNs allow natural modeling and verification of dynamic structure, they do not increase the modeling power of GSPNs. In fact, the dynamic structure of any D-GSPN must be finite, otherwise transforming D-GSPNs towards GSPNs may become infinite. To consider infinite structures, we extended INRSs (Li et al. 2009b) to model reconfiguration in GSPNs and could publish two papers (Tigane et al. 2017c, 2016). In these two extensions, each reconfiguration is expressed by a rule having left- and right-hand sides. The application of a rule implies the substitution of its left-hand-side image, in a given GSPN to be reconfigured, by its right-hand side. These two sides must belong to particular sets in order to allow developers to reconfigure a live, bounded and reversible GSPN while preserving these three essential properties in the resulting model. However, these formalisms suffer from three major drawbacks: (i) system states are not considered (reconfigurations are done in an off-line mode), (ii) only three qualitative properties, namely liveness, boundedness and reversibility are decidable and finally (iii) the quantitative aspect of GSPNs is not studied.

To remedy these problems, we have proposed a new formalism, called INRSs-GSPNs (Tigane et al. 2018), which takes into account, inter alia, system states in the reconfiguration application and provides an algorithm for both qualitative and quantitative verifications. At the modeling level, designers will be able to model a reconfigurable system and its dynamic structure using GSPNs and rewriting rules that are controlled by the system state. Unlike our extensions described in Tigane et al. (2016, 2017c), which limit rule application to an initial marking, we associate each reconfiguration rule with a marking controlling its activation, i.e. if the net has not yet reached a marking, then the rule is not yet applicable. As for the verification level, an algorithm computing from the dynamic model, the Markov chain, describing the stochastic behavior of the system is proposed. As a result, the designers can evaluate the system performance. Nevertheless, the reconfiguration remains too limited in INRSs-GSPNs formalism. On the one hand, only live, bounded and reversible GSPNs are concerned, which limits its application field. On the other

hand, left- and right-hand sides of rules must belong imperatively to a particular set which can only further limit the formalism applicability. The need to (i) relax the constraints imposed by INRSs-GSPNs formalism, (ii) address all types of GSPNs (not only live, bounded and reversible GSPNs) and (iii) enrich the set of nets used in reconfiguration led us to propose reconfigurable generalized stochastic Petri nets (RecGSPNs) (Tigane et al. 2019). Actually, RecGSPNs formalism allows designers to model a wider range of structural changes where both sides of any rule are no longer defined by their structure. Instead, they are defined by their behaviors. The use of RecGSPNs-based reconfiguration allows preserving five important properties, namely, liveness, boundedness, reversibility, deadlock-freedom and home state. Moreover, many properties expressed by linear time logic (Baier and Katoen 2008) can be preserved after a system reconfiguration. Thus, these properties are decidable regardless of the number of obtained configurations that can be infinite. This practice enjoys double advantages:

1) Temporal and spatial complexity are reduced since these properties are verified only at the first configuration, hence no need to compute and explore the whole set of reachable states of all reachable configurations.

2) Often, applying rules to graphs leads to structurally infinite models, and hence properties are not decidable based on classical verification techniques. In the RecGSPNs approach, several properties are still decidable since applying any rule preserves them.

I.4. Book Organization

This chapter has introduced the book topic, stated research motivation and objectives and given an overview of our contributions. The remainder of this book is organized as follows:

Chapter 1 presents the background theory of Petri nets, model checking, Markov chains and generalized stochastic Petri nets. In order to introduce generalized stochastic Petri nets, we describe Petri nets in their basic form, and provide basic definitions for both model checking and Markov chains. Finally, the extension of Petri nets toward GSPNs is presented.

Chapter 2 introduces the graph transformation field and its use in PNs context. Initially, graph transformation systems are outlined. Then, we focus on graph transformation applications to PNs in the literature. As for verification aspects, we discuss some proposed verification algorithms obtained by either developing new ones or updating and transferring existing ones to graph transformation systems. Finally, we conclude by showing the advantages/disadvantages of today's formalisms and their verification techniques proposed for dynamic structure PNs.

Chapter 3 describes two of our primary contributions, namely GSPNs-RT (Tigane et al. 2017a) and CD-SRNs (Tigane et al. 2021), in which we present trivial approaches that introduce dynamic topologies into GSPNs and SRNs, as well as transform GSPNs and SRNs with dynamic topologies into their respective equivalent underlying formalisms. The obtained equivalent models are exploited in the verification of dynamic nets using classical analysis approaches.

Chapter 4 develops extensions of GSPNs-RT, namely D-GSPNs and ID-GSPNs, in which the sets of places and transitions are dynamic and can be transformed using transformation rules formalized by DPO approach. Also, both formalisms are equipped by algorithms that transform D-GSPNs and ID-GSPNs into equivalent GSPNs.

Chapter 5 consists of our major contribution to this book. It describes a new approach, called reconfigurable GSPNs (Tigane et al. 2019). Similarly to graph transformation systems, it formalizes system configuration as GSPNs, and it describes structure evolution as transformation rules. The chapter concludes with a discussion about the decidability of some properties of dynamic systems even if they can be structurally infinite, i.e. the reachable configuration set is infinite.

In **Chapter 6**, we compare the proposed approaches with the current state-of-the-art. We start showing new qualitative modeling aspects provided by RecGSPNs and D-GSPNs formalisms. Then, a quantitative comparison that shows how the proposed approaches optimize time and memory consumption in the verification phase is provided.

Finally, in the **Conclusion** of this book, we summarize this book and discuss possible directions in future work.