SLAP$_N$: A Tool for Slicing Algebraic Petri Nets

Yasir Imtiaz Khan and Nicolas Guelfi

University of Luxembourg, Laboratory of Advanced Software Systems
6, rue R. Coudenhove-Kalergi, Luxembourg
{yasir.khan,nicolas.guelfi}@uni.lu

Abstract. Algebraic Petri nets is a well suited formalism to represent the behavior of concurrent and distributed systems by handling complex data. For the analysis of systems modelled in Algebraic Petri nets, model checking and testing are used commonly. Petri nets slicing is getting an attention recently to improve the analysis of systems modelled in Petri nets or Algebraic Petri nets. This work is oriented to define Algebraic Petri nets slicing and implement it in a verification tool.

1 Introduction

Among several dedicated analysis techniques for Petri nets (PNs) and Algebraic Petri nets (APNs) (i.e., an evolution to PNs), model checking and testing are used more commonly. A typical drawback of model checking is its limits with respect to the state space explosion problem. Similarly testing suffers with the problems such as large input amount of test data, test case selection etc.

Petri nets slicing is a technique that aims to improve the verification of systems modelled in Petri nets. *PNs slicing* is used to syntactically reduce Petri net model based on the given *criteria*. A criteria is a property for which Petri net model is analysed. The sliced part constitutes only that part of the PN model that may affect the criteria. Roughly, we can divide *PN Slicing* into two major classes, which are

Static Slicing: If the initial markings of places are not considered for generating sliced net.

Dynamic Slicing: If the initial markings of places are considered for generating sliced net.

One characteristic of APNs that makes them complex to slice is the use of multisets of algebraic terms over the arcs. In principle, algebraic terms may contain variables. Even though, we want to reach a syntactically reduced net (to be semantically valid), its reduction by slicing, needs to determine the possible ground substitutions of these algebraic terms. We use partial unfolding proposed in [1] to determine ground substitutions of the algebraic terms over the arcs of an APN. In the first column of Table1, our proposed APN slicing algorithms are shown [2, 3]. The second column represents properties that are preserved by algorithm whereas in the last column slicing type is mentioned.

Table 1. Different APNs Slicing Algorithms

Algorithm	*Preserved Prop*	*Type Slicing*
Abstract Slicing	CTL^*_{-X}	*Static*
APN Slicing	LTL_{-X}	*Static*
Liveness Slicing	*Livenss*	*Static*
Concerned Slicing	*Particular*	*Dynamic*

1.1 SLAPN$_N$ Overview

First of all, an APN is partially unfolded and from the temporal description of properties places are extracted (shown in Fig.1). Different slicing algorithms such as *abstract slicing, concerned slicing, APN slicing, safety slicing, liveness slicing* can be used to generate the slice (can be observed in the meta model of SLAP$_N$ (shown in Fig.2)). Following steps an APN slice can be generate by the tool.

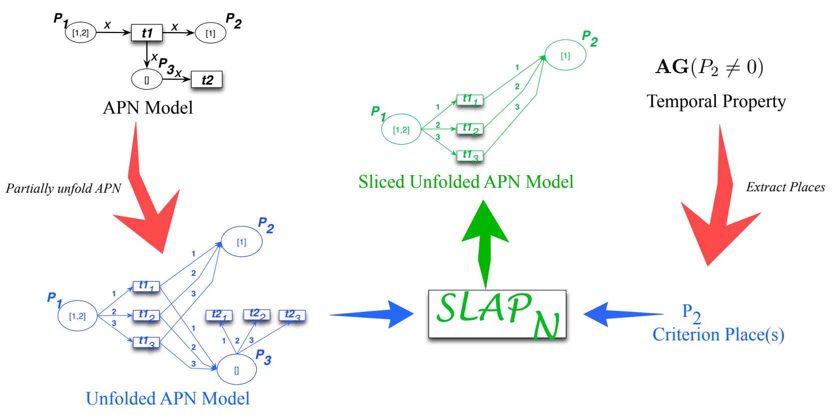

Fig. 1. SLAP$_N$ Overview

Step 0: Create an APN model and interesting property (i.e., writing a property in the form of temporal formula).

Step 1: Choose a slicing algorithm.

Step 3: Sliced APN model is generated.

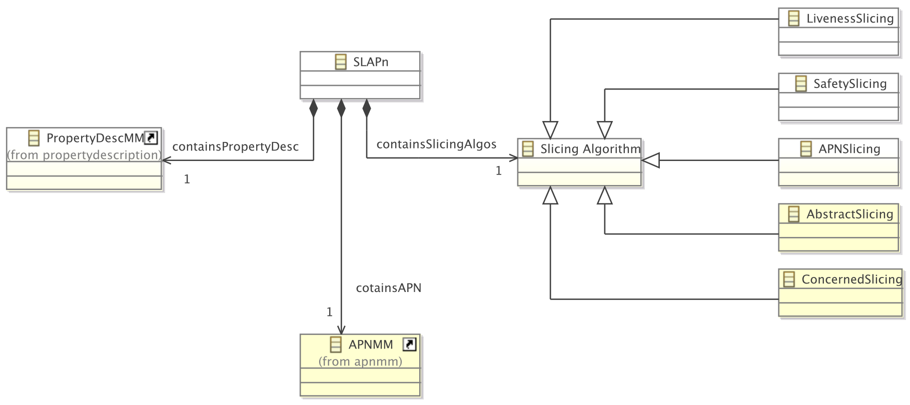

Fig. 2. SLAP$_N$ Meta Model

2 Conclusion and Future Work

– APN slicing can be used as a pre-processing step towards the verification of systems modelled in APNs. The sliced APN model can then be used to generate state space.
– Our work is the first effort to define and implement the proposed slicing algorithms.
– As a future work, we consider to integrate SLAP$_N$ with the existing model checkers such as AlPiNA [3].
– We intend to develop SLAP$_N$ as a generic tool over the PN classes such as timed PN, colored PN.

References

1. D. Buchs, S. Hostettler, A. Marechal, and M. Risoldi. Alpina: A symbolic model checker. In J. Lilius and W. Penczek, editors, *Applications and Theory of Petri Nets*, volume 6128 of *Lecture Notes in Computer Science*, pages 287–296. Springer Berlin Heidelberg, 2010.
2. Y. I. Khan. Slicing high-level petri nets. Technical Report TR-LASSY-14-03, University of Luxembourg, 2014.
3. Y. I. Khan and M. Risoldi. Optimizing algebraic petri net model checking by slicing. *International Workshop on Modeling and Business Environments (ModBE'13, associated with Petri Nets'13)*, 2013.

This work has been supported by the National Research Fund, Luxembourg, Project RESIsTANT, ref.PHD-MARP-10.