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

Yasir Imtiaz Khan and Nicolas Guelfi

{yasir.khan, nicolas.guelfi}@uni.lu

UNIVERSITÉ DU LUXEMBOURG

Laboratory for Advanced Software Systems

Problem Description

Algebraic Petri nets (APNs) 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 APNs, model checking and testing are used commonly. The question is how to define APN slicing and how to implement it in a verification tool.

PN Slicing

PN slicing is used to syntactically reduce Petri net model based on a given *criteria*. A criteria is a property for which PN model is analysed. The sliced part constitutes the only part of the PN model that may affect the criteria.

Roughly, we can divide *PN Slicing* into two major classes (shown in Fig.1), which are

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

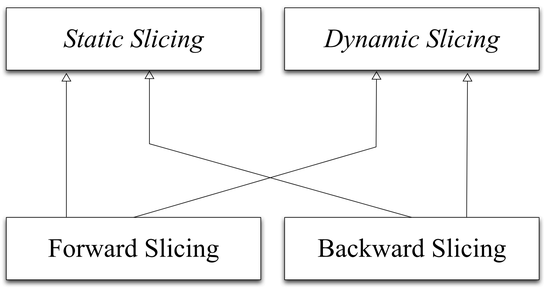
Dynamic Slicing: If the initial markings of places are considered for generating sliced net.

There are two ways to generate the static and dynamic slices that are forward and backward slicing. A Forward slice starts from the initially marked places and then by forward traversal of PN model until the places extracted from the criteria, a sliced net is generated. A Backward slice starts from the places extracted from criteria and then by backward traversal until reaching a fixed point, a sliced net is generated.

We propose slicing algorithms for APNs:

- APN Slicing
- Abstract Slicing
- Concerned Slicing
- Liveness Slicing

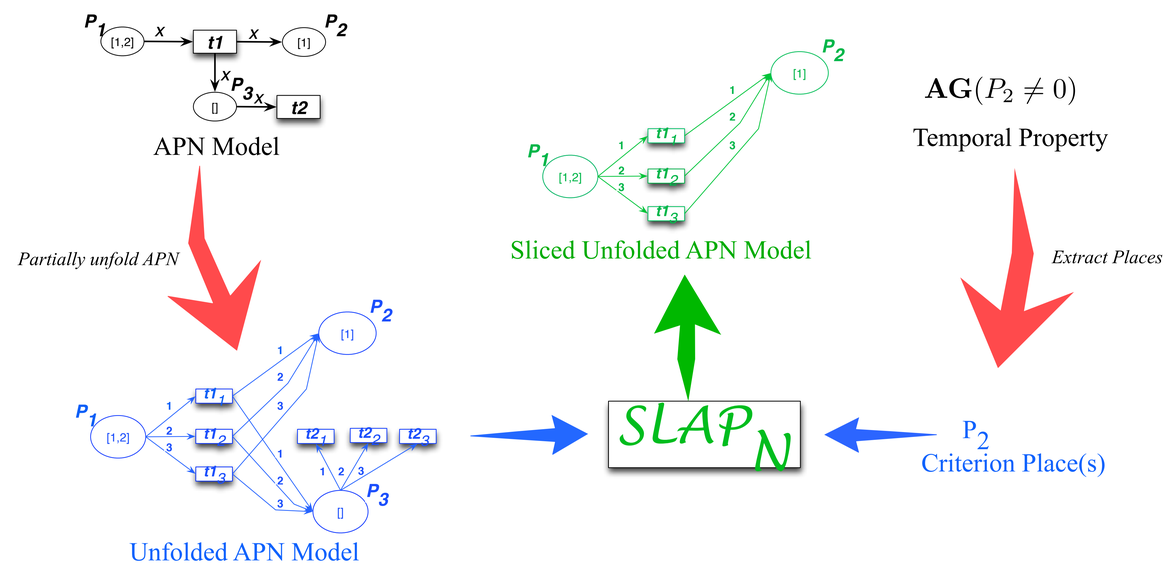
Types of slicing Figure 1



SLAP$_N$ Overview

The objective of SLAP$_N$ is to show the practical usability of slicing by implementing the proposed slicing algorithms. First of all an APN is partially unfolded and, from the temporal description of properties, places are extracted (shown in Fig2). Different slicing algorithms such as *abstract slicing, concerned slicing, APN slicing, safety slicing, liveness slicing*(shown in the meta model of SLAPN$_N$, Fig3) can be used to generate the slice.

SLAPN$_N$ FLOW Figure 2



Slicing Algebraic Petri nets

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 [3] to determine ground substitutions of the algebraic terms over the arcs of an APN.

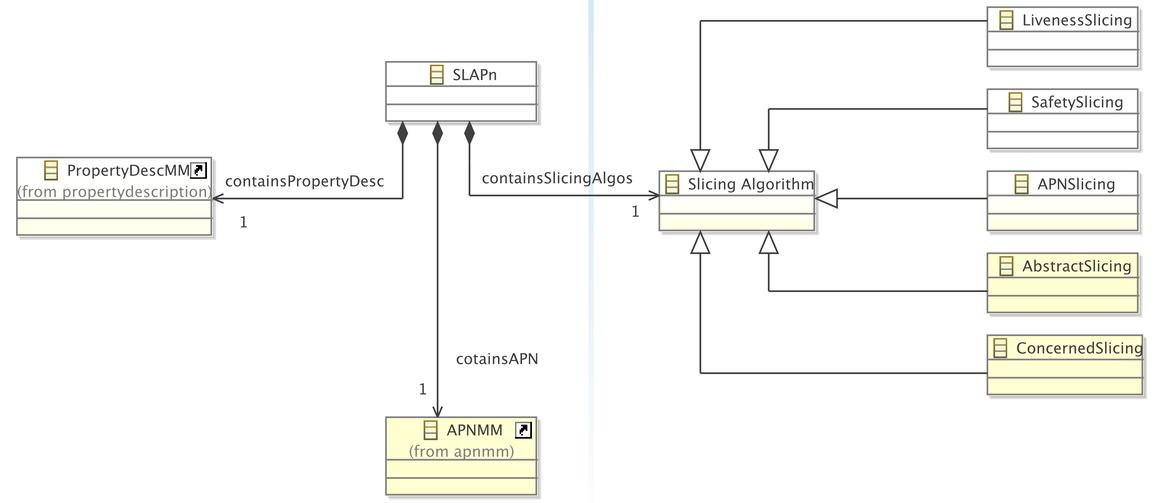
References

[1] Y.I. Khan, M. Risoldi Optimizing Algebraic Petri net model checking by Slicing In *ModBE'13 associated Petri Nets'13*, 2013.

[2] Y.I. Khan, N. Guelfi Slicing High-level Petri nets *echnical Report TR-LASSY-14-03, University of Luxembourg*, 2014.

[3] D. Buchs, S. Hostettler,A. Marechal,A. Linard,M. Risoldi, AlPiNA: A symbolic model checker In *Sprinter Berlin Heidelberg'10*, 2010.

SLAP$_N$ Meta Model Figure 3



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.

Aknowledgement

*This work has been supported by the FNR, Luxembourg, Project RESIsTANT, PHD-MARP-10. Presented at PNSE'14.