

Verification of Reconfigurable Petri Nets

Julia Padberg

HAW Hamburg, Department of Informatics,
<http://users.informatik.haw-hamburg.de/~padberg/>

Abstract We introduce a family of modeling techniques consisting of Petri nets together with a set of rules. For reconfigurable Petri nets, e.g. in [3] not only the follower marking can be computed but also the structure can be changed by rule application to obtain a new net. Motivation is the observation that in increasingly many application areas the underlying system has to be dynamic in a structural sense. Complex coordination and structural adaptation at run-time (e.g. mobile ad-hoc networks, dynamic hardware reconfiguration, communication spaces, ubiquitous computing) are main features that need to be modelled adequately. The distinction between the net behaviour and the dynamic change of its net structure is the characteristic feature that makes reconfigurable Petri nets so suitable for modeling systems with dynamic structures.

For rule-based modification of Petri nets we use the framework of net transformations that is inspired by graph transformation systems [2]. The basic idea behind net transformation is the stepwise modification of Petri nets by given rules. The rules present a rewriting of nets where the left-hand side is replaced by the right-hand side. The abstract semantics we introduce in [4] is a graph with nodes that consist of isomorphism classes of the net structure and an isomorphism class of the current marking. Arcs between these nodes represent computation steps being either a transition firing or a direct transformation. Based on this semantics we can define properties and model-check these properties.

Model checking is a widely used technique to prove properties such as liveness, deadlock or safety for a given model. Here we present model checking of reconfigurable Petri nets [7,6]. The main task is to flatten the two levels of dynamic behavior that reconfigurable nets provide, the firing of transitions on the one hand and the transformation of the nets on the other hand. We show how to translate a reconfigurable net into Maude modules [1]. Maude's LTL model-checker is then used to verify properties of these modules. The correctness of this conversion is proven as the corresponding labelled transitions systems are bisimilar.

In an ongoing example reconfigurable Petri nets are used to model and to verify partial dynamic reconfiguration of field programmable gate arrays using the tool **ReconNet** ([5] or see <https://reconnetblog.wordpress.com/>).

Keywords: Petri Nets, Verification, Reconfigurable Petri Nets

References

1. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Jose F. Quesada. Maude: specification and programming in rewriting logic. *Theor. Comput. Sci.*, 285(2):187–243, 2002.
2. H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in TCS. Springer, 2006.
3. Hartmut Ehrig and Julia Padberg. Graph grammars and Petri net transformations. In *Lectures on Concurrency and Petri Nets*, pages 496–536, 2004.
4. J. Padberg. Abstract interleaving semantics for reconfigurable Petri nets. *ECE-ASST*, 51, 2012.
5. Julia Padberg, Marvin Ede, Gerhard Oelker, and Kathrin Hoffmann. Reconnet: A tool for modeling and simulating with reconfigurable place/transition nets. *ECE-ASST*, 54, 2012.
6. Julia Padberg and Alexander Schulz. Model checking reconfigurable Petri nets with Maude. In Rachid Echahed and Mark Minas, editors, *Graph Transformation, 9th Int. Conf. on*, volume 9761 of *Lecture Notes in Computer Science*, pages 54–70. Springer, 2016.
7. Alexander Schulz. Model checking of reconfigurable Petri nets. Master's thesis, University of Applied Sciences Hamburg, 2015.