A basic tool for the modeling of Marked-Controlled Reconfigurable Petri Nets*

Marisa Llorens and Javier Oliver

Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia (UPV)
Camino de Vera s/n, E-46022 Valencia, Spain
{mllorens, fjoliver}@dsic.upv.es

In [1, 4], we introduced the model of *net rewriting systems* (NRS) and a subclass of this model, *reconfigurable nets* (RN), to analyze, simulate and verify concurrent and distributed systems that are subject to structural dynamic changes. Both models arise from two different lines of research that were conducted in the field of the Petri net formalism [6]. The first line covers various proposals for merging Petri nets with *graph grammars* [2, 3, 7], while the second line, which is best represented by *Valk's self-modifying nets* [8], considers Petri nets whose flow relations can vary at runtime. *Reconfigurable nets* attempt to combine the most relevant aspects of both of these approaches and constitute a class of models for which each of the fundamental properties of Petri nets (place boundedness, reachability, deadlock and liveness) are decidable. The translation of this model into Petri nets is automatic [4]. This equivalence ensures that the model is amenable to automatic verification tools. In contrast, the class of *net rewriting systems* is Turing powerful [4]. In [5], we introduced *marked-controlled net rewriting systems* (MCNRS) and *marked-controlled reconfigurable nets* (MCRN) as an extension of net rewriting systems and reconfigurable nets, respectively, where the enabling of a rewriting rule not only depends on the net topology, but also depends on the net marking according to some net places named control places. In MCNRS, a system configuration is described as a Petri net, and a change in configuration is described as a graph rewriting rule, which consists of replacing part of the system (the part that matches the left-hand side of the rewriting rule) with another one (given by the right-hand side of the rewriting rule). A MCRN is a MCNRS where a change in configuration is limited to the modification of the flow relations of the places in the domain of the rewriting rule involved; i.e., the set of places and transitions is left unchanged by rewriting rules.

In this work, we introduce a basic tool to analyze the structure and dynamic behavior of systems that are modeled using our nets in order to evaluate them and suggest improvements or changes. The goal of our MCReNet tool is to analyze, simulate, and verify real systems that are modeled by MCRN. On the basis of the equivalence between Petri nets and MCRN [5], we have initially

* This work has been partially supported by the EU (FEDER) and the Spanish MEC under grant TIN2005-09207-C03-02, by the ICT for EU-India Cross-Cultural Dissemination Project ALA/95/23/2003/077-054 and by the Programa de Apoyo a la Investigación y Desarrollo (PAID-04-06) de la Universidad Politécnica de Valencia.

opted to develop a tool that exploits the capabilities of the Petri net tools that are currently available [9]. The main goal of our tool is the implementation of a translation algorithm of a MCRN into an equivalent Petri net. We have chosen PIPE as the graphical editor, simulator and analyzer. PIPE is an open source, platform independent tool for Petri nets developed at the Imperial College of Science Technology and Medicine in the University of London. The last version of PIPE can be obtained at `sourceforge.net/projects/pipe2/`. MCReNet integrates PIPE with the translator. Therefore, the user can draw the initial state and rewriting rules of the MCRN, which is then translated into its equivalent Petri net. In addition, a simulation and an analysis of this net can be performed. All these processes are integrated in the MCReNet tool. MCReNet can be obtained by sending an email to the author (`mllorens@dsic.upv.es`). We are now working on the implementation of an algorithm that obtains the state of a MCRN from the corresponding Petri Net. The idea is that users can animate the Petri net (firings of enabled transitions) and, at any time, they can find and see the equivalent state of the equivalent MCRN. As further work, we will develop a tool that is completely autonomous. Our new tool will be able to directly carry out the editing, simulation and analysis of net properties on MCRN. We are currently working on the theoretical aspects of the analysis methods for Petri nets in order to apply them directly to MCRN.

References

1. E. Badouel, M. Llorens, and J. Oliver. Modeling Concurrent Systems: Reconfigurable Nets. In *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications* (PDPTA'03), vol. IV, pp. 1568–1574, USA, 2003.
2. P. Baldan. Modelling Concurrent Computations: From Contextual Petri Nets to Graph Grammars. PhD Thesis, University of Pisa TD-1/00, 2000.
3. A. Corradini. Concurrent Computing: From Petri Nets to Graph Grammars. Invited talk at the *Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation*, Elsevier, *ENTCS*, vol. 2, 1995.
4. M. Llorens and J. Oliver. Structural and Dynamic Changes in Concurrent Systems: Reconfigurable Petri Nets. In *IEEE Transactions on Computers*, vol. 53, no. 9, pp. 1147–1158, September 2004.
5. M. Llorens and J. Oliver. Introducing Structural Dynamic Changes in Petri Nets: Marked-Controlled Reconfigurable Nets. In *Proc. Int. Conf. on Automated Technology for Verification and Analysis* (ATVA'04), Springer-Verlag, *LNCS*, vol. 3299, pp. 310–323, Taipei, Taiwan, 2004.
6. T. Murata. Petri Nets: Properties, Analysis and Applications. In *Proc. of the IEEE*, vol. 77, no.4, pp. 541–580, 1989.
7. H. Schneider. Graph Grammars as a Tool to Define the Behavior of Processes Systems: From Petri Nets to Linda. In *Proc. 5th Int. Conf. on Graph Grammars and their Application to Computer Science*, pp. 7–12, Williamsburg, USA, 1994.
8. R. Valk. Self-modifying Nets, a Natural Extension of Petri Nets. In *Proc. Int. Coll. on Automata, Languages and Programming* (ICALP'78), Springer-Verlag, *LNCS*, vol. 62, pp. 464–476, Udine, Italy, 1978.
9. The Home Page of Petri Net Tools: `http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/`.