Modeling and Verifying Distributed Systems with Petri Nets

Souheib Baarir
LIP6, CNRS UMR 7606 and
Université Paris Ouest Nanterre La Défense
200, avenue de la République, Nanterre, France
Souheib.Baarir@lip6.fr

Fabrice Kordon
LIP6, CNRS UMR 7606,
Université P. & M. Curie
4, place Jussieu, 75005 Paris, France
Fabrice.Kordon@lip6.fr

I. Introduction

Nowadays, systems tend to be more and more distributed. Distribution brings a huge complexity and a strong need to deduce possible (good and bad) behaviors on the global system, from the known behavior of its actors.

For such systems, we know that classical development methods are not adequate since the coverage of possible executions is too low [1]. This is an old observation that led people to investigate the use of formal methods.

The objective of this tutorial is to focus on one technique that is suitable for the modeling and verification of distributed systems: *Petri nets*. After the tutorial, they should be able to model a problem with Petri nets, express properties and use a tool to check them.

II. Petri Nets

Petri nets [2], [3], [4] are a mathematical notation (or formalism) that are dedicated to the modeling, analyzing and verifying reactive and distributed systems. Their strength are their simple but precise semantics and their clear graphical notation. Moreover, there are many state of the art methods and algorithms for analysis and verification, most of them being implemented in software tools (see the Petri Net Tool Database [5]).

III. Content of the Tutorial

The course introduces Petri nets and their theory by means of examples from different application domains. A first focus will be on traditional Petri net classes, in particular on Place/Transition-Systems and on some standard concepts and properties such as reachability, deadlocks, invariants, etc.

We will then introduce the basic concepts of colored Petri Nets that allow for a more compact specification with the same semantics.

Finally, we will focus on more evolved temporal properties, like those expressed by LTL/CTL. They allow to check for more sophisticated situations that occurs in distributed systems such as the absence of livelock.

IV. Practice during the tutorial

The tutorial will contain several practical exercises the students will perform with a dedicated tool. The idea is to let participants design their specification and check for some properties with real tools.

For this practice, we selected the *CosyVerif* environment [6]. This tool is jointly developed by three laboratories in the Paris area where well known teams act in formal methods: LIP6, LIPN and LSV.

CosyVerif is open source software. Students willing to attend are thus greatly encouraged to upload the environment (available on Mac, Linux and Windows) prior to the tutorial. All the required information is provided on the web site [6].

References

[1] J. Gogen and Luqi, "Formal methods: Promises and problems," *IEEE Software*, vol. 14, no. 1, pp. 75–85, 1997.

[2] C. Girault and R. Valk, Eds., *Petri Nets and System Engineering*. Springer Verlag, 2003, ch. 2, pp. 9–23.

[3] K. Jensen and L. Kristensen, *Coloured Petri Nets - Modeling and Validation of Concurrent Systems*. Springer, 2009.

[4] M. Diaz, *Petri Nets: Fundamental Models, Verification and Applications*. Wiley, 2009.

[5] Petri Net Community, "The Petri Net Tool Database." [Online]. Available: http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html

[6] LIP6, LIPN, and LSV, "The CosyVerif Verification Environment." [Online]. Available: http://www.cosyverif.org

978-0-7695-4845-6/12 $26.00 © 2012 IEEE
DOI 10.1109/IWAISE.2012.24