Complex Event Processing Modeling by Prioritized Colored Petri Nets (Summary) [⋆]

Hermenegilda Macià[1], Valentín Valero[1], Gregorio Díaz[1], Juan Boubeta-Puig[2], and Guadalupe Ortiz[2]

[1] School of Computer Science, University of Castilla-La Mancha, Albacete
{hermenegilda.macia,valentin.valero,gregorio.diaz}@uclm.es
[2] Dep. of Computer Science and Engineering, University of Cádiz, Cádiz, Spain
{juan.boubeta,guadalupe.ortiz}@uca.es

Summary of the contribution

Complex Event Processing (CEP) is a technology that allows us to process and correlate large volumes of data by using event patterns, aiming at promptly detecting specific situations that could require special treatment. The event types and event patterns for a particular application domain are implemented by using an Event Processing Language (EPL). Although some current model-driven tools allow end users to easily define these patterns, which are then transformed automatically into a particular EPL, the generated code is syntactically but not semantically validated. To deal with this problem, a Prioritized Colored Petri Net model (PCPN) for CEP is proposed and conducted in this paper.

Thus, we have not only an event pattern graphical representation, but also the capability to perform formal analysis, and therefore semantic analysis, by means of the PCPN model obtained and the CPN Tools. This formal analysis is twofold. On the one hand, users can interact with the model itself by performing a step by step debugging, since the tool allows to simulate the model. With this in mind, users can specify a concrete scenario by providing the initial marking to check whether the model works as expected. By doing this, users can observe the results of the individual steps of the simulation, which represent the different EPL operators, as we can observe at the end of the case study where a user can detect whether the preferred operator has been used, that is, if the specified pattern behaves as expected. On the other hand, there are certain advantages of performing automatic simulations. An automatic simulation allows us to actually execute the EPL code and compare the obtained output, that is, we can compare whether the results obtained from a given input are the same when we execute the EPL code in the Esper EPL online tool and in CPN Tools.

Acknowledgements

This work was partially supported by projects TIN2015-65845-C3, subprojects 2-R and 3-R, TIN2014-53986-REDT and PR2016-032.

[⋆] This work has been published in IEEE Access Volume 4: 7425-7439, 2016. Impact Factor: 1.270 DOI: doi.org/10.1109/ACCESS.2016.2621718