

On the Interplay between Model-Driven Engineering, Domain-Specific Languages, and Petri Nets

Manuel Wimmer

CDL-MINT, TU Wien, Austria
wimmer@big.tuwien.ac.at

Abstract Model-Driven Engineering (MDE) [2] is a paradigm to tame the complexity of modern systems, automate engineering tasks, and support inter-human communication. Domain-Specific Languages (DSLs) [4] are an important ingredient for MDE as they allow domain experts (who may not necessarily be software experts) to express their concerns with more familiar concepts than with traditional general purpose languages. However, DSLs typically have a smaller user base which increases the relative development and maintenance cost, not only for the languages themselves, but also for their accompanying tools [7]. Now the question arises how to reduce this overhead. Can well-studied general purpose languages such as Petri Nets help in this setting?

In my talk, I will demonstrate that Petri Nets can play a very central role in this setting, especially when it comes to the semantics of DSLs [3]. In the last decade, several metamodeling frameworks emerged, e.g., see AToMPM¹, GEMOC Studio², GME³, MetaEdit+⁴ to name just a few examples, which allow not only the definition of the syntax of modeling languages and the generation of supporting tools such as model editors and validators, but also semantic concerns may be specified and automatically turned into tool support such as execution engines [5], debuggers [1], and verification engines [8]. However, having a unified methodology to develop DSLs systematically to benefit from rich out-of-the-box tool support is still an open research challenge where Petri Nets may form an important cornerstone.

Furthermore, I will present some lessons learned from past projects and outline new application areas for DSLs combined with Petri Nets in the realm of Industry 4.0. In particular, I will report on our experiences of realizing model transformation engines based on Coloured Petri Nets [10] and their utilization for specifying the semantics of declarative model transformation languages [9]. Finally, recent advances are outlined how the prescriptive models of MDE are combined with the descriptive models of process mining to allow for continuously improved software systems [6].

¹ <https://atompm.github.io>

² <http://gemoc.org/studio.html>

³ <http://w3.isis.vanderbilt.edu/Projects/gme/index.html>

⁴ <https://www.metacase.com/products.html>

Keywords: Meta-Modeling, Petri Nets, Domain-Specific Modeling, Language Workbench, Model Transformations, Process Mining

References

1. Bousse, E., Leroy, D., Combemale, B., Wimmer, M., Baudry, B.: Omniscient debugging for executable DSLs. *Journal of Systems and Software* **137**, 261–288 (2018). <https://doi.org/10.1016/j.jss.2017.11.025>
2. Brambilla, M., Cabot, J., Wimmer, M.: *Model-Driven Software Engineering in Practice, Second Edition*. Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers (2017). <https://doi.org/10.2200/S00751ED2V01Y201701SWE004>
3. Bryant, B.R., Gray, J., Mernik, M., Clarke, P.J., France, R.B., Karsai, G.: Challenges and directions in formalizing the semantics of modeling languages. *Comput. Sci. Inf. Syst.* **8**(2), 225–253 (2011). <https://doi.org/10.2298/CSIS110114012B>
4. Kosar, T., Bohra, S., Mernik, M.: Domain-specific languages: A systematic mapping study. *Information & Software Technology* **71**, 77–91 (2016). <https://doi.org/10.1016/j.infsof.2015.11.001>
5. Mayerhofer, T., Langer, P., Wimmer, M., Kappel, G.: xMOF: Executable DSMLs Based on fUML. In: *Proceedings of the 6th International Conference on Software Language Engineering (SLE)*. pp. 56–75 (2013). https://doi.org/10.1007/978-3-319-02654-1_4
6. Mazak, A., Wimmer, M.: On Marrying Model-driven Engineering and Process Mining: A Case Study in Execution-based Model Profiling. In: *Proceedings of the 6th International Symposium on Data-driven Process Discovery and Analysis (SIM-PDA)*. pp. 78–88 (2016), <http://ceur-ws.org/Vol-1757/paper6.pdf>
7. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. *ACM Comput. Surv.* **37**(4), 316–344 (2005). <https://doi.org/10.1145/1118890.1118892>
8. Meyers, B., Deshayes, R., Lucio, L., Syriani, E., Vangheluwe, H., Wimmer, M.: ProMoBox: A Framework for Generating Domain-Specific Property Languages. In: *Proceedings of the 7th International Conference on Software Language Engineering (SLE)*. pp. 1–20 (2014). https://doi.org/10.1007/978-3-319-11245-9_1
9. Wimmer, M., Kappel, G., Schönböck, J., Kusel, A., Retschitzegger, W., Schwinger, W.: A Petri Net Based Debugging Environment for QVT Relations. In: *Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. pp. 3–14 (2009). <https://doi.org/10.1109/ASE.2009.99>
10. Wimmer, M., Kappel, G., Schönböck, J., Kusel, A., Retschitzegger, W., Schwinger, W.: TROPIC: A Framework for Model Transformations on Petri Nets in Color. In: *Companion Proceedings to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. pp. 783–784 (2009). <https://doi.org/10.1145/1639950.1640013>