Behavioral Interfaces for Executable DSLs

Dorian Leroy[1], Erwan Bousse[2], Manuel Wimmer[3], Tanja Mayerhofer[4], Benoit Combemale[5], Wieland Schwinger[6]

Abstract: This work summarizes our paper [Le20] originally published in the Journal of Software and Systems Modeling in 2020 about a novel language engineering approach.

Keywords: Domain Specific Languages, Metamodeling, Language Workbenches

A large amount of domain-specific languages (DSLs) are used to represent behavioral aspects of systems in the form of behavioral models [BCW17]. Executable domain-specific languages (xDSLs) enable the execution of behavioral models [Ma13]. While an execution is mostly driven by the model's content (e.g., control structures, conditionals, transitions, method calls), many use cases require interacting with the running model, such as simulating scenarios in an automated or interactive way or coupling system models with environment models. The management of these interactions is usually hard-coded into the semantics of xDSLs, which prevents its reuse for other xDSLs and the provision of generic interaction tools.

To tackle these issues, we propose a novel metalanguage for complementing the definition of xDSLs with explicit behavioral interfaces to enable external tools to interact with executable models in a unified way. A behavioral interface defines a set of events specifying how external tools can interact with models that conform to xDSLs implementing the interface. Additionally, we define two types of relationships involving behavioral interfaces: the implementation relationship and the subtyping relationship. An implementation relationship ties a behavioral interface to a given operational semantics implementation. Subtyping relationships allow to build event abstraction hierarchies, indicating that events from one interface can be abstracted or refined as events from another interface.

We implemented the proposed metalanguage in the GEMOC Studio, an Eclipse-based language and modeling workbench for xDSLs, and evaluate the approach with three

[1] Institute of Business Informatics - Software Engineering, JKU Linz, Altenbergerstr. 69, 4040 Linz, Austria dorian.leroy@jku.at
[2] Nantes Software Modeling Group, University of Nantes, Chemin de la Houssiniere, 44322 Nantes, France erwan.bousse@ls2n.fr
[3] Institute of Business Informatics - Software Engineering, JKU Linz, Altenbergerstr. 69, 4040 Linz, Austria manuel.wimmer@jku.at
[4] Business Informatics Group, TU Wien, Favoritenstr. 9-11, 1040 Wien, Austria mayerhofer@big.tuwien.ac.at
[5] DiverSE Team, University of Rennes 1, 263 Avenue du General Leclerc, 35042 Rennes, France benoit.combemale@irisa.fr
[6] Institute of Telecooperation, JKU Linz, Altenbergerstr. 69, 4040 Linz, Austria wieland.schwinger@jku.at

doi:10.18420/SE2021_25

demonstration cases: (*i*) we show that the proposed metalanguage can be used to define the behavioral interface of xDSLs; (*ii*) we show that behavioral interfaces enable the definition of generic tools and their reuse across several xDSLs; (*iii*) we show that a single behavioral interface can be subtyped by several xDSLs, allowing to interact with and reason about the execution of models through a common behavioral interface.

References

[BCW17] Brambilla, M.; Cabot, J.; Wimmer, M.: Model-Driven Software Engineering in Practice. Morgan & Claypool, 2017.

[Le20] Leroy, D.; Bousse, E.; Wimmer, M.; Mayerhofer, T.; Combemale, B.; Schwinger, W.: Behavioral interfaces for executable DSLs. Softw. Syst. Model. 19/4, pp. 1015–1043, 2020, URL: https://doi.org/10.1007/s10270-020-00798-2.

[Ma13] Mayerhofer, T.; Langer, P.; Wimmer, M.; Kappel, G.: xMOF: Executable DSMLs Based on fUML. In (Erwig, M.; Paige, R. F.; Wyk, E. V., eds.): Proceedings of the 6th International Conference on Software Language Engineering (SLE). Vol. 8225. Lecture Notes in Computer Science, Springer, pp. 56–75, 2013, URL: https://doi.org/10.1007/978-3-319-02654-1%5C_4.