Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

A DSL-DRIVEN WEB TOOL FOR MODELING AND SIMULATION OF RDEVS ARTIFACTS

Clarisa Espertino[1], María Julia Blas[2], Silvio Gonnet[2]

[1]Facultad Regional Santa Fe, Universidad Tecnológica Nacional, Santa Fe, Santa Fe, ARGENTINA
[2]Instituto de Desarrollo y Diseño INGAR, CONICET-UTN, Santa Fe, Santa Fe, ARGENTINA

ABSTRACT

Routed Discrete Event System Specification (RDEVS) is a formalism that extends the Discrete Event System Specification (DEVS) to support the Modeling and Simulation (M&S) of routing processes in discrete event-based scenarios. We present a web-based tool that, following the principles of Model-Driven Engineering (MDE), combines textual modeling with code generation to reduce development time, ensure consistency with the RDEVS formalism, and promote independence from external software tools.

1 INTRODUCTION

RDEVS (Blas et al. 2022) is an extension of DEVS (Zeigler et al. 2018) designed to model routing processes in discrete event simulations. It supports the representation of systems where events are routed through interconnected components, making it suitable for domains like communication networks and logistics.

As in the construction of DEVS models (Van Tendeloo et al., 2016), RDEVS modeling relies on libraries or desktop tools that offer textual or graphical modeling capabilities but lack integration and web accessibility, as shown by previous works deployed in Eclipse-based environments (Blas and Gonnet 2021, Espertino et al., 2024). To address these limitations and to decouple the M&S process of RDEVS from Eclipse (used both as a development and target platform), this paper introduces a tool that embeds a Domain-Specific Language (DSL) and modeling infrastructure within a browser-accessible environment.

2 WEB ARCHITECTURE AND TOOL IMPLEMENTATION

Principles of Domain-Specific Modeling (DSM) and DSL guided the development. At the core of DSM, modelers describe their solutions using concepts and notations that directly reflect the problem domain. In our case, the domain is described through routing processes modeled under the RDEVS formalism.

The proposed tool enables the specification of networks composed of nodes, edges, and behaviors with precision and simplicity through a DSL. Such a DSL is based on two complementary levels of syntax: concrete and abstract. The former is specified through a context-free grammar (CFG) implemented with ANTLR, which defines how valid RDEVS models should be written, allowing automated parsing and early detection of textual errors. On the other hand, the abstract syntax is grounded in network theory, representing systems as graphs of interconnected nodes. It captures the structural meaning of RDEVS models through a metamodel defined using the Eclipse Modeling Framework (EMF). This metamodel describes key language elements (i.e., network, node, node type, and link). However, not all network structures are valid representations of RDEVS routing processes, as a network must satisfy certain conditions (e.g., avoiding isolated or self-connected nodes). To enforce these requirements, OCL is used to specify constraints embedded directly into the metamodel as OCL annotations. Such constraints are used by EMF as part of the metamodel validation. This procedure ensures that only valid network models (i.e., those accurately mapped to routing processes) proceed to further phases. Components and technologies highlighted in yellow in Figure 1 were used to implement both syntax levels in the web tool.

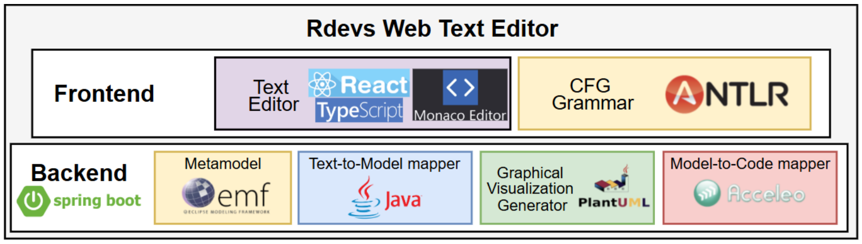

Figure 1: High-level architecture, components and technologies of the web-based software tool.

Finally, a model-to-code mapper (highlighted in red in Figure 1) was developed using Acceleo, a template-based code generator. It automatically translates EMF model instances into executable Java code using an existing RDEVS Java library, minimizing manual effort and reducing the risk of errors. In addition to the modeling capability, the tool includes a graphical visualization component that automatically renders the modeled scenario, implemented using PlantUML (component highlighted in green in Figure 1). In this way, the tool helps modelers to inspect and verify the network structure, promoting better understanding and error detection in complex models.

As a result, we developed a web-based tool (available here) supporting textual modeling, validation, graphical visualization, and automatic code generation of RDEVS simulation models. The components described above were organized into backend and frontend layers as illustrated in Figure 1.

3 MODELING WORKFLOW: THE MODELER PERSPECTIVE

The modeler specifies a scenario using the DSL through a special editor that provides syntax highlighting and real-time detection of grammatical errors. Once the input is parsed by ANTLR, the tool maps the specification into an EMF model instance. Errors related to the metamodel and its constraints are displayed during the user-triggered validation process. If the modeler chooses to generate the graphical representation, the corresponding module builds it based on the metamodel instance. After successful validation, the model instance is used to generate Java code that is downloaded. An example can be found here.

4 CONCLUSIONS

A functional web-based tool for the textual modeling of RDEVS simulation scenarios was successfully developed. Unlike previous solutions that require desktop environments and separate graphical and textual modeling, this tool offers a hybrid web platform accessible directly from any browser. Within this architecture, a structured DSL tailored to RDEVS is embedded, allowing modelers to focus on scenario definition without deep programming knowledge, while ensuring consistency with the RDEVS formalism.

Although the current version generates Java code, using EMF as a modeling foundation decouples the model specification from the implementation language, enabling future extensions to other code generators.

REFERENCES

Blas, M. J., and S. Gonnet. 2021. Computer-aided design for building multipurpose routing processes in discrete event simulation models. *Engineering Science And Technology, An International Journal*, 24(1), 22-34.

Blas, M. J., H. Leone, and S. Gonnet. 2022. DEVS-based Formalism for the modeling of routing processes. *Software and Systems Modeling*, 21(3):1179-1208.

Espertino, C., M. J. Blas, and S. Gonnet. 2024. Mapping RDEVSNL-Based Definitions of Constrained Network Models to Routed DEVS Simulation Models. *Journal of the Brazilian Computer Society* 30 (1):17-34.

Van Tendeloo, Y., & H. Vangheluwe. 2016. An evaluation of DEVS simulation tools. *SIMULATION*, 93(2): 103-121.

Zeigler, B. P., A. Muzy, and E. Kofman. 2018. Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations. 3rd ed. London, Academic Press.