

Scientific Work: Creating a web-based model transformation UI (with GLSP)

Florian Weidner

Philipps-University Marburg, Germany

Department of Mathematics and Computer Science, Software engineering group

May 06, 2025

I. MOTIVATION AND INTRODUCTION

In software engineering, often Model-Driven Engineering (MDE) is used to increase development productivity and quality. Concepts are modeled closer to the domain, so that they describe important aspects of a solution with human-friendly abstractions. The models can also be used to generate application fragments, that can be directly used as source code. In the process of MDE, many activities need to transform source models into different target models, while following a set of transformation rules. This model transformation process is based on algebraic graph transformations. A metamodel is used to model the structure and rules of the concept. The resulting transformation language can provide automatic model creation, development and maintenance activities. [7] One framework to use MDE is Eclipse Modeling Framework (EMF) by the Eclipse Foundation. It provides a basis for application development, using modeling and code generation facilities. Much frameworks build upon EMF, providing various MDE tools like code generators, graphical diagramming, model transformation or model validation. [8] One model transformation framework is Henshin. [10] It tries to provides model transformation capabilities with a high level of usability. [9] For metamodels it uses EMF Ecore files and for instance models EMF XMI files. The framework enables transformations on XMI instance files with a defined transformation language. It provides a graphical and textual syntax to create these transformation rules. [10] Henshin can be used as a eclipse plugin. Eclipse makes it easy to access, but especially for new users, the heavy editor makes the use of Henshin unintuitive.

Therefore the goal exists to create a graphical option to use the Henshin model transformations without the overhead of the heavy eclipse editor. A web-based graphical editor would make the use of Henshin even more accessible and intuitive.

Graphical Language Server Platform (GLSP) is a open-source framework by the Eclipse Foundation to develop custom diagram editors for distributed web-applications. [6] It can be used in Eclipse Desktop IDE, Eclipse Theia, Visual Studio Code and embedded in any website. With these functionalities, GLSP fits to create an accessible, intuitive application to create and apply Henshin model transformations.

II. BACKGROUND

In this section, the theoretical background of the project and used technologies are described.

A. Eclipse Foundation

The Eclipse Foundation is a not-for-profit, member-supported corporation that provides an environment for individuals and organizations for collaborative and innovative software. [4] The Eclipse Foundation grew out of the Publication of the Eclipse Integrated Development Environment (IDE) code from IBM in 2001. The Eclipse Foundation itself was founded in 2004. The new organization was founded to continue the development of Eclipse IDE as an open source platform. Over time many different projects were initiated by the organization in the Eclipse environment, all running under the Eclipse Public License. [5, 4] In the recent years, the key initiatives of the Eclipse Foundation are contributing to european digital sovereignty, enhancing security measures, innovating Software-Defined Vehicle (SDV), organizing community events and improving their most popular projects. Popular projects are for example the Jakarta EE, a ecosystem for cloud native applications with java, Eclipse Temurin, providing open source Java Development Kits and the Eclipse IDE. [2] In total, the Eclipse Foundation hosts more than 400 open source projects, supported 14 european research projects in 2024 and has 117 organisations participating in commits [2] The scope of this work refers

to many projects of the Eclipse Foundation and the development of a web-based model transformation UI is mainly based on EMF, Henshin and GLSP, which all are projects running under the Eclipse Foundation.

The Eclipse IDE is not the main project but it is still an important part of the Eclipse infrastructure. It is divided into four main components: Equinox, the Platform, the Java Development Tools (JDT) and the Plug-in Development Environment (PDE). Together they provide everything to develop and extend Eclipse-based tools. Equinox and the Platform are the core of the Eclipse IDE. With expanding the core with the JDT or other plugins, the IDE can be used to develop different programming languages, like Java, C/C++ or PHP. [8] Eclipse provides different packages to download, depending on the use case. One package is the Eclipse Modeling Tools package by the Eclipse Modeling Project. It provides tools and runtimes to build model-based applications. It can be used to graphically design domain models and test those models by creating and editing dynamic instances. Also Java code can be generated from the models to get a basis for applications. [1] The base of the Eclipse Modeling Tool is EMF (section II-C). Other modeling tools and projects, that are built on top of the EMF core functionality, provide capabilities for model transformation, database integration, or graphical editor generation. [8]

B. Henshin

One part of the Eclipse Modeling Project for model transformation is Henshin. It can be used as a plugin in the Eclipse IDE and provides a graphical and textual syntax to define model transformation rules and apply them on EMF XMI instance models. It can be used for endogenous transformations, where EMF model instances are directly transformed, and exogenous transformations, where new instances are generated from given instances using a trace model. It also brings efficient in-place execution of transformations using an interpreter with debugging support and a performance profiler. Henshin also provides conflict and dependency analysis, and state space analysis for verification. [10]

C. Eclipse Modeling Framework (EMF)

„Eclipse Modeling Framework (EMF) is a modeling framework and code generation facility for building tools and other applications based on a structured data model.“ [3]

Eclipse Modeling Framework (EMF) is the core part of the Eclipse Modeling Project and consists of three components.

D. GLSP

GLSP is a framework that provides components for the development of GUIs (GUIs) for web-based diagram editors. [6]

III. RELATED WORK/SOFTWARE

Similar already existing tools for model checking: Henshin, Groovy...

A. Implementation

IV. CONCLUSION

REFERENCES

- [1] Eclipse Foundation. *Eclipse Modeling Project*. Accessed: 2025-06-02. 2025. URL: <https://eclipse.dev/modeling/>.

- [2] Eclipse Foundation. *2024 Annual Community Report*. en. https://www.eclipse.org/org/foundation/reports/annual_report.php. Accessed: 2025-6-2. 2024.
- [3] Eclipse Foundation. *Eclipse Modeling Framework*. <https://github.com/eclipse-emf/org.eclipse.emf>. 2025.
- [4] Michael Jastram. *Die Eclipse Foundation wird 20 Jahre alt: Die interessantesten SE-Projekte*. de. <https://www.se-trends.de/eclipse-foundation-wird-20-jahre-alt/>. Accessed: 2025-6-2. Dec. 2024.
- [5] Alexandra Kleijn. *Eclipse und die Eclipse Foundation*. Accessed: 2025-6-2. 2006. URL: <https://www.heise.de/-221997>.
- [6] Philip Langer and Tobias Ortmayr. *Eclipse GLSP*. <https://github.com/eclipse-glsp/glsp>. 2025.
- [7] S. Sendall and W. Kozaczynski. “Model transformation: the heart and soul of model-driven software development”. In: *IEEE Software* 20.5 (2003), pp. 42–45. DOI: 10.1109/MS.2003.1231150.
- [8] David Steinberg et al. *EMF: Eclipse Modeling Framework 2.0*. 2nd. Addison-Wesley Professional, 2009. ISBN: 0321331885.
- [9] Daniel Strüber et al. “Henshin: A Usability-Focused Framework for EMF Model Transformation Development”. In: *Graph Transformation*. Ed. by Juan de Lara and Detlef Plump. Cham: Springer International Publishing, 2017, pp. 196–208. ISBN: 978-3-319-61470-0.
- [10] Daniel Strueber. *Henshin*. <https://github.com/eclipse-henshin/henshin/wiki>. 2025.