Proposal: 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. [2] 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. [3] One model transformation framework is Henshin. [5] It tries to provides model transformation capabilites with a high level of usability. [4] 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. [5] 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. [1] It can be used in Eclipse Desktop IDE, Eclipse Theia, Visual Studio Code and embedded in any website. With these fuctionalities, GLSP fits to create an accessible, intuitive application to create and apply Henshin model transformations.

II. Project Requirements

The following requirements are a list of the functional and non-functional requirements that i came up with. The client should be integrated as a Eclipse Theia client. Additional clients can be added in the future without overhead effort.

Functional requirements:

- EMF XMI instance files should be displayed in a graphical editor
- Henshin rule files should be displayed in a graphical editor
- EMF Ecore metafiles should be displayed in a graphical editor
- The instance editor should display all rules that can be applied to the instance model.
- Parameters of the rules should be editable when applying a rule.
- After applying a rule, the instance model should be updated and displayed in the instance editor as a temporal file that can be used to apply multible rules. The initial instance model should not be changed.
- The instance editor should provide editing functionality for the instance model.
- The Henshin rule editor should provide editing functionality for the transformation rules.
- The Ecore editor should provide editing functionality for the Ecore metamodel.

Once all functional requirements are implemented, the application should fully support a basic model transformation workflow. Its functionality should be equivalent to using the Henshin plugin within the Eclipse editor. It should be possible, that additional Henshin functionalities like State Space analysis or confilct and dependency analysis can be added in the future.

Non-functional requirements:

- The application should be web-based and accessible via a web browser.
- The application should be responsive and work on different screen sizes.
- The application should be user-friendly and intuitive to use.
- The application should be performant and handle large models efficiently.

III. Basic architecture

GLSP uses a client-server architecture. For the backend, the framework supports integration with EMF models as the underlying source for the diagrams. That makes the integration of Henshin easier, because all files are based on EMF. The *HenshinRessourceSet* can be loaded over the EMF integration of GLSP directly. For the XMI instance files, the Henshin rule files and the Ecore metamodel files, a mapping to the GLSP internal graphical model needs to be implemented. [**eclipseGLSP**]

IV. Implementation

V. Preliminary table of contents

CONTENTS

VI. Time schedule

The goal is to finish the project and the thesis at the end of August. From now on that includes 14 weeks. The three main tasks are the development of the application, writing the scientific work about the thesis and writing the final thesis.

TABLE I
PROJECT DEVELOPMENT TASKS

Task No.	**Description**	**Estimated Time Needed**	**Planned Completion**
1	Creating a Prove of Concept to apply rules on an instance model	2 weeks	31. Mai
2	Design Frontend and System architecture	1 week	07. Juni

TABLE II
THESIS WRITING TASKS

Task No.	**Description**	**Estimated Time Needed**	**Planned Completion**
1	basics chapter about model transformations	1 week	07.Juni
1	basics chapter about EMF and the Eclipse Foundation	1 week	07.Juni
1	basics chapter about Henshin and GLSP	1 week	07.Juni

VII. Conclusion

References

[1] Philip Langer and Tobias Ortmayr. *Eclipse GLSP*. https://github.com/eclipse-glsp/glsp. 2025.

[2] 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.

[3] David Steinberg et al. *EMF: Eclipse Modeling Framework 2.0*. 2nd. Addison-Wesley Professional, 2009. ISBN: 0321331885.

[4] 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.

[5] Daniel Strueber. *Henshin*. https://github.com/eclipse-henshin/henshin. 2025.