

# Proposal: Web-based Model Checking UI with GLSP

Florian Weidner

Philipps-University Marburg, Germany

Department of Mathematics and Computer Science, Software engineering group

May 06, 2025

## I. OPEN QUESTIONS

- Welchen Umfang hat die Arbeit? 60-100
- Was ist ein guter name: Henshin Web

## II. MEETING NOTES

Meeting 08.05.2025:

- Wissenschaftliches Arbeiten: Grundlagenkapitel
- erster schritt anwendung der regeln auf instance Models
- schreiben über grundlagen -> eclipse an sich
- henshin interpreter api
- 2 stufen: instance transformationen -> regeln spezifizieren

Meeting 22.05.2025 Fragen:

- POC zeigen/besprechen (Henshin maven package)
- Name für Software/Projekt?
- Anforderungen abgleichen/ergänzen
- Daniel Strüber
- undo redo transformation
- matches anzeigen?

Meeting 18.06.2025 Fragen:

- Editor zeigen (Indexing, Layout, Undo/Redo)
- Nächste Entwicklungs Schritte: XMI Editor stabilisieren -> Henshin Rule Viewer
- Was fehlt bei scientific work?
- Anmeldung vorgehen? anmeldung auf website ausfüllen
- deployment

## III. IMPLEMENTATION EXPERIENCES

### A. Inclusion of Henshin into the GLSP Maven Project

- Glsp nutzt maven
- Henshin kein offizielles maven package -> einfachster weg zum einbetten.
- Möglichkeiten:
  - Henshin SDK als maven package
  - Henshin SDK als git submodule
  - Henshin SDK als jar file
- Henshin SDK jar importiert Plugins und hat nicht alle .class dateien direkt im jar.
- jars vereinen nicht möglich -> Henshin Plugins sind signiert.
- Shading nicht möglich -> GLSP shaded auch und das ist nicht doppelt möglich.
- Aktuelle Lösung: Henshin Plugins einzeln als maven package importieren.
- Es hat sich herausgestellt, das das model plugin eine weitere abhängigkeit auf nashorn (JavaScript Engine) hat -> registrieren von Nashorn über Service Provider Interface (SPI) (META-INF/services).

### B. Layout der GLSP UI

-