Diagram editors with Eclipse GLSP (Graphical Language Server Platform)

Philip Langer
planger@eclipsesource.com
Eclipse Graphical Language Server Platform (GLSP)
Eclipse Graphical Language Server Platform (GLSP)

Applying the architectural pattern of LSP to graphical modeling

- Development of browser-based diagram clients
- Frontend focused on rendering & user interaction
- Encapsulate language smarts on the diagram server
Eclipse Graphical Language Server Platform (GLSP)

- Initialization with parameters
  - URI
  - Diagram type
  - ...
Eclipse Graphical Language Server Platform (GLSP)

- Server has responsibility to obtain model source
Eclipse Graphical Language Server Platform (GLSP)

→ Model source can be anything
  - EMF
  - Xtext
  - Databases
  - JSON
  - ...

© 2021 EclipseSource | http://eclipsesource.com | Philip Langer | Diagram editors with Eclipse GLSP
Eclipse Graphical Language Server Platform (GLSP)

- Server maps source model into **graph model**
  - JSON description of the diagram: typed nodes & edges
- Client translates **graph model** into SVG
  - Map node & edge types to SVG views with Eclipse Sprotty 🌟
  - Generic editing tools

---

© 2021 EclipseSource | http://eclipsesource.com | Philip Langer | Diagram editors with Eclipse GLSP
Eclipse Graphical Language Server Platform (GLSP)

- Communication is based on actions
  - Several types of actions (defined in protocol)
- Extensible with custom actions
  - Custom handlers on server or client
Code available as an example in the GLSP repos
0.9 Release
Available soon (review)
Acts as RC for 1.0

1.0 Release
Planned for EOY 2021
0.9 API unless issues raised

Copy & Paste, Undo & Redo
Auto-completable Label Editing
Visual Feedback
Validation Markers
Navigation
API, Marquee, ...

Tool Palette
Eclipse Incubation
Sprotty

2018

2021
Ideas by Adopters and Contributions by Community

Examplarily applied to publicly available GLSP example
Inspiration for your own diagram editors
Leveraging Frontend Technology & Custom Actions

- Benefit from the flexibility of pure SVG and CSS

- Not just eye candy
  - Make it interactive
  - Combine with modeling language logic (server)
  - Support comprehension and analysis of diagram

→ let’s look at an example
Add Custom Controls and Transient Views

- Frontend user interface can easily be extended
  - UIExtensions
  - Pure HTML, React, …
  - Send and receive custom actions to diagram client & server

- Compute and show transient diagram views
  - E.g. filter elements, show impact, etc.
  - Use auto-layouting if necessary for those views
  - Switch to readonly mode for transient views

→ let’s look at an example
Even More Native VSCode Integration

- Native diagram menu
- Clipboard support
- Integration with VSCode problems view
- Support for navigation

- VSCode GLSP API
  - Define native commands, menu items, etc.
  - Support for native VSCode Context keys
  - Message interceptors for customizing client/server communication
Tables, Server- and Client-side Automatic Routing

- Automatic routing based on Libavoid
- Client-side Libavoid routing
  - Webassembly port
  - Soon to be available in GLSP

Vladyslav Hnatiuk

@Aksem
Constrained Layouts

Custom client-side edit tools

Automatic layouting on the server
Zoom-dependent Level of Detail

- Opens up entirely new opportunities
  - No property views anymore? Zoom in and see/edit all properties
  - Navigation through layered diagrams (e.g. package → classes → activities)

- Hopefully available in GLSP by the end of 2021
Conclusion

- Slim abstractions where control matters (e.g. rendering)
- Highly extensible and open for customization
- Arbitrary “source models” (not only EMF, …)
- Integratable with tool platforms and domain-specific tools
  - Integration with Theia, VSCode, or Eclipse
  - Avoiding lock-in effect with specific tool / app platform
- Join innovative community and discuss your ideas with us!
  - Spotlight session, Tue 26.10., 17:30 CET
  - BOF, Wed 27.10., 18:15 CET

Model validation, diffing and more with EMF.cloud
Wed 27.10., 16:10 CET

Papyrus UML - the first stage of a journey to the cloud
Thu 28.10., 16:10 CET

Building a model application with Theia and EMF.cloud: the DISCO experience
Thu, 28.10., 16:50 CET
Evaluate this Session:

- Please help by leaving feedback on the sessions you attend!
- To rate a session, you must be registered for it in Swapcard BEFORE the talk starts.
- Swapcard will prompt you to leave feedback after the end of each session.
- You may also rate a talk by locating the session from the “Agenda” or “My Event” buttons on the Event Home page. Click on the session and look for the “Give your feedback” box.