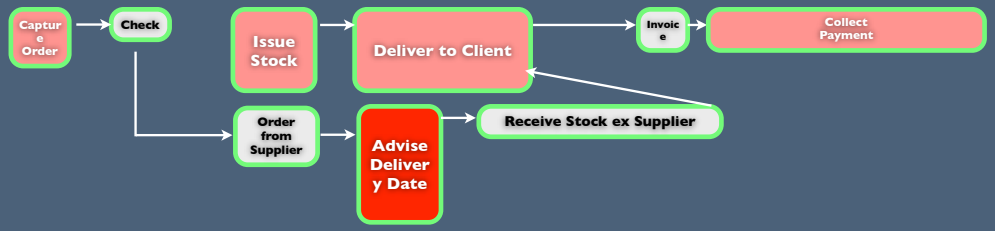
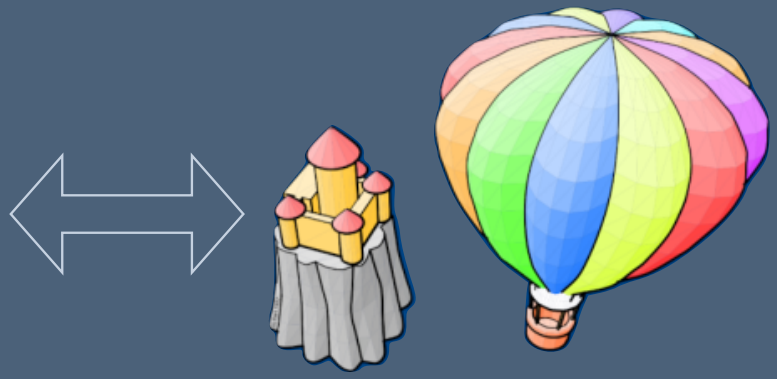




GLOSS - A GLSP[1] Model Server on the Smalltalk Platform

International Workshop on Smalltalk Technologies • Lille, France • 8th-11th July 2024

1 Graphical Language Server Protocol

Graham McLeod • University Duisburg-Essen & inspired.org
Gareth Cox • inspired.org

UNIVERSITÄT DUISBURG ESSEN

inspired!

Agenda

- Introduction
- Graphical Modelling
- IDEs
- Language Server Protocol
- Graphical Language Server Protocol
- Our project
- Results, Recommendations, Conclusions

inspired!

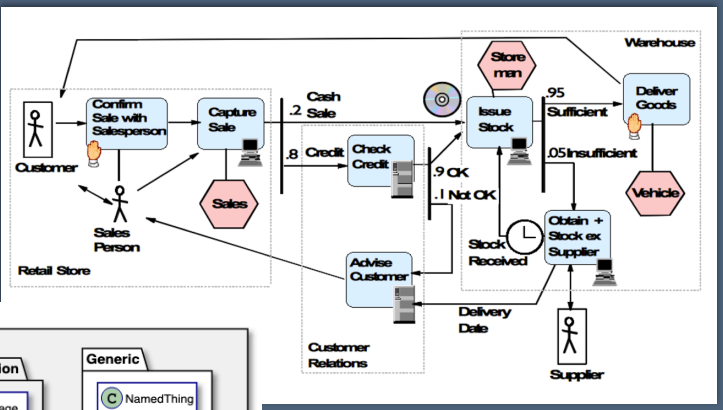
Introduction

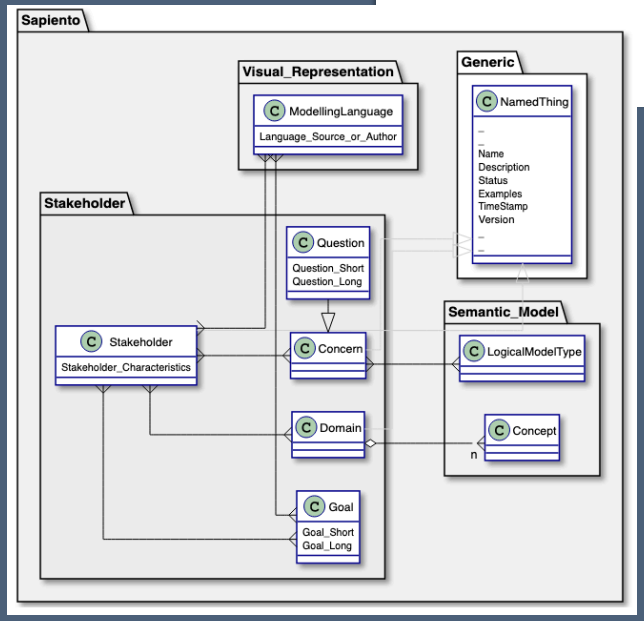
- Inspired is a boutique consultancy in Enterprise Architecture (Business, Application, Data, Technology and Solutions)
- We do methods engineering, training, consulting, and tools
- We offer a SaaS enterprise modelling platform written in Smalltalk (VAST for server, Pharo for graphical modelling client)
- Personal 30 year history with Smalltalk
- Also doing research into Visual Language design

Graphical Languages

e.g. UML, BPMN, Archimate

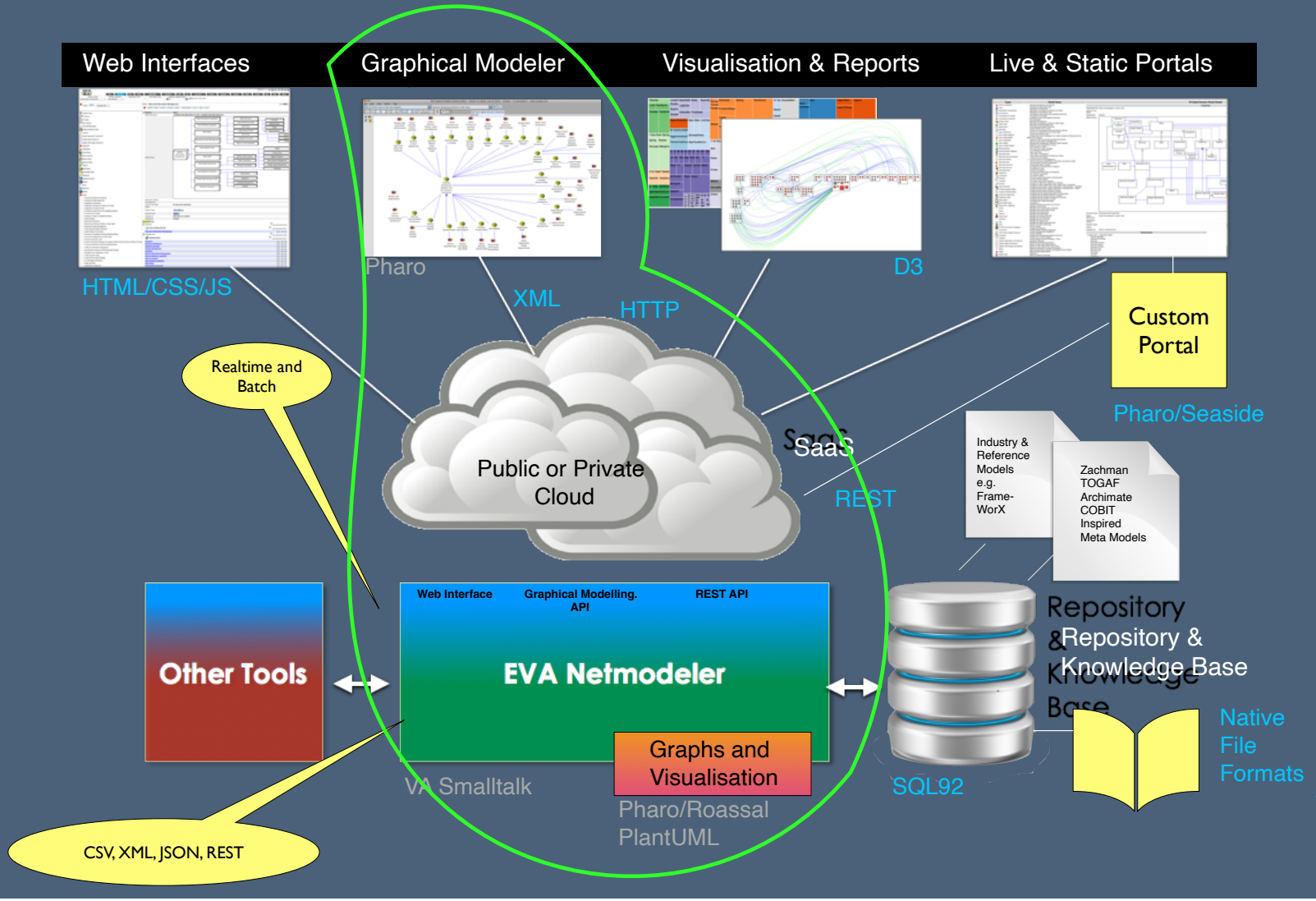
- Express semantics (meaning) relevant to a domain
  - Types/Classes and instances
  - Relationships
  - Properties/Property Values
- Represent these graphically through
  - Symbols
  - Lines
  - Adornments





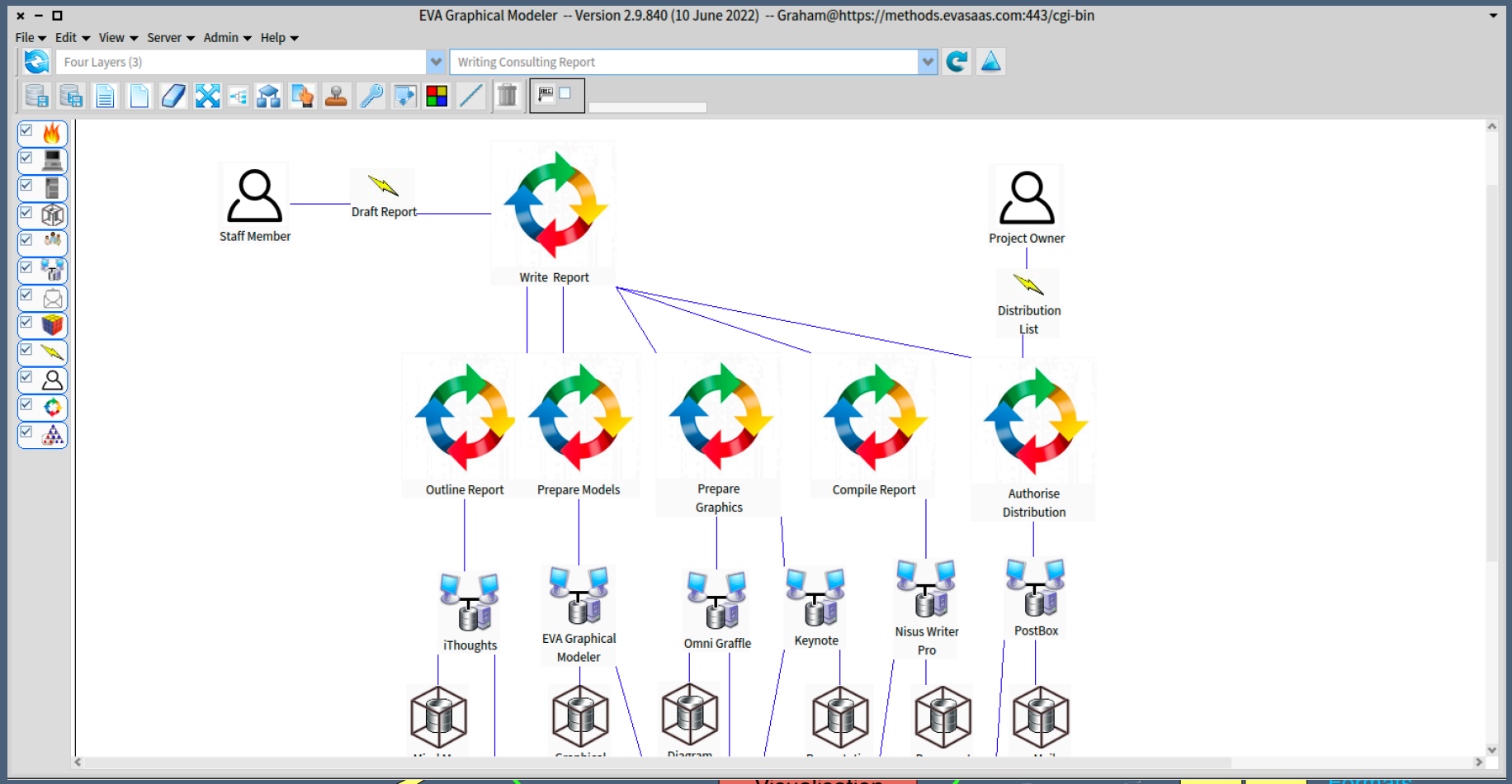
inspired!

EVA Netmodeler



inspired!

EVA Netmodeler



VA Smalltalk

Visualisation

Pharo/Roassal

PlantUML

SQL92

Formats

CSV, XML, JSON, REST

inspired!

EVA GM Model Handling

Manages how a Model Type is defined in terms of Types to include and what symbol should be used to represent them (in Model Type diagram) and instances (in Models)

Manages model instances and their relationships to the repository instances they represent

Manages model relationships and the relationships between instances they represent

Note that model type definition is held as XML property

Note that model visual only information such as position, size, colour override, relationship path is stored in XML property

Vector symbols and line characteristics are defined in a Logo derived script held on symbol instances

inspired!

EVA GM Model Handling

Manages how a Model Type is defined in terms of Types to include and what symbol should be used to represent them (in Model Type diagram) and instances (in Models)

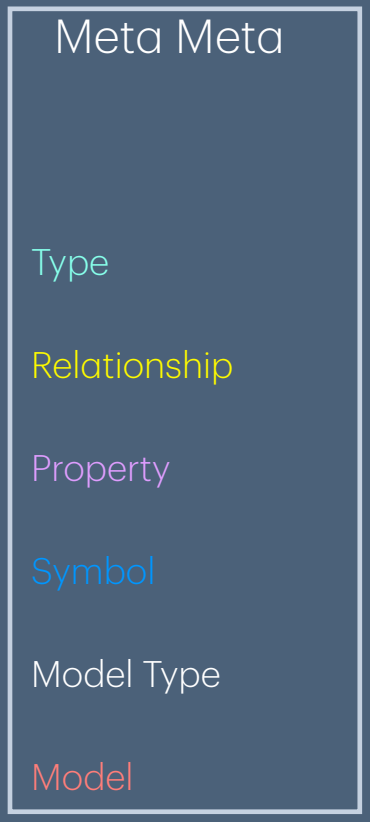
Manages model instances and their relationships to the repository instances they represent

Manages model relationships and the relationships between instances they represent

Note that model type definition is held as XML property

Note that model visual only information such as position, size, colour override, relationship path is stored in XML property

Vector symbols and line characteristics are defined in a Logo derived script held on symbol instances



inspired!

EVA GM Model Handling

Manages how a Model Type is defined in terms of Types to include and what symbol should be used to represent them (in Model Type diagram) and instances (in Models)

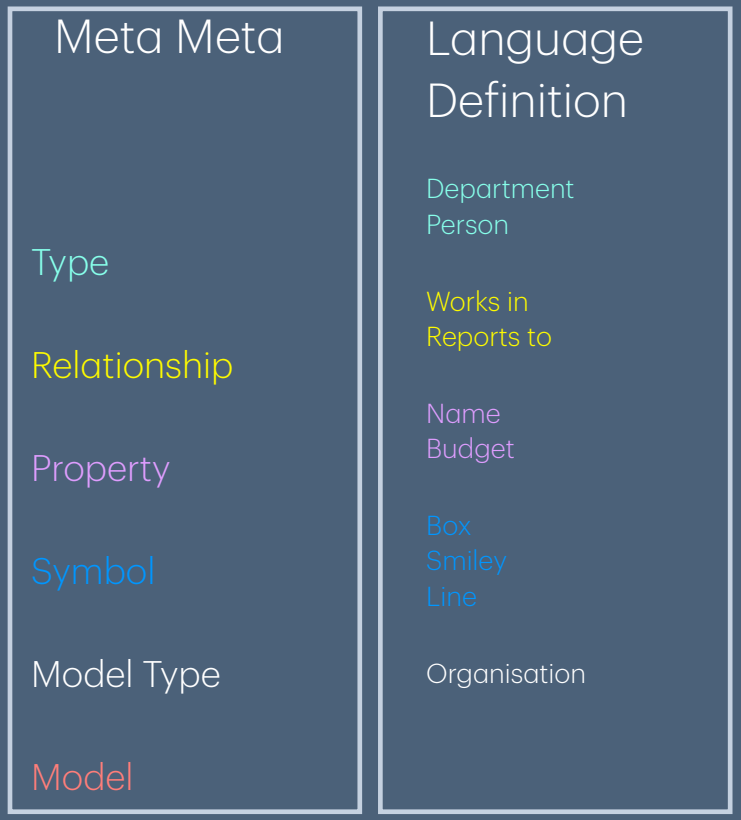
Manages model instances and their relationships to the repository instances they represent

Manages model relationships and the relationships between instances they represent

Note that model type definition is held as XML property

Note that model visual only information such as position, size, colour override, relationship path is stored in XML property

Vector symbols and line characteristics are defined in a Logo derived script held on symbol instances



inspired!

EVA GM Model Handling

Manages how a Model Type is defined in terms of Types to include and what symbol should be used to represent them (in Model Type diagram) and instances (in Models)

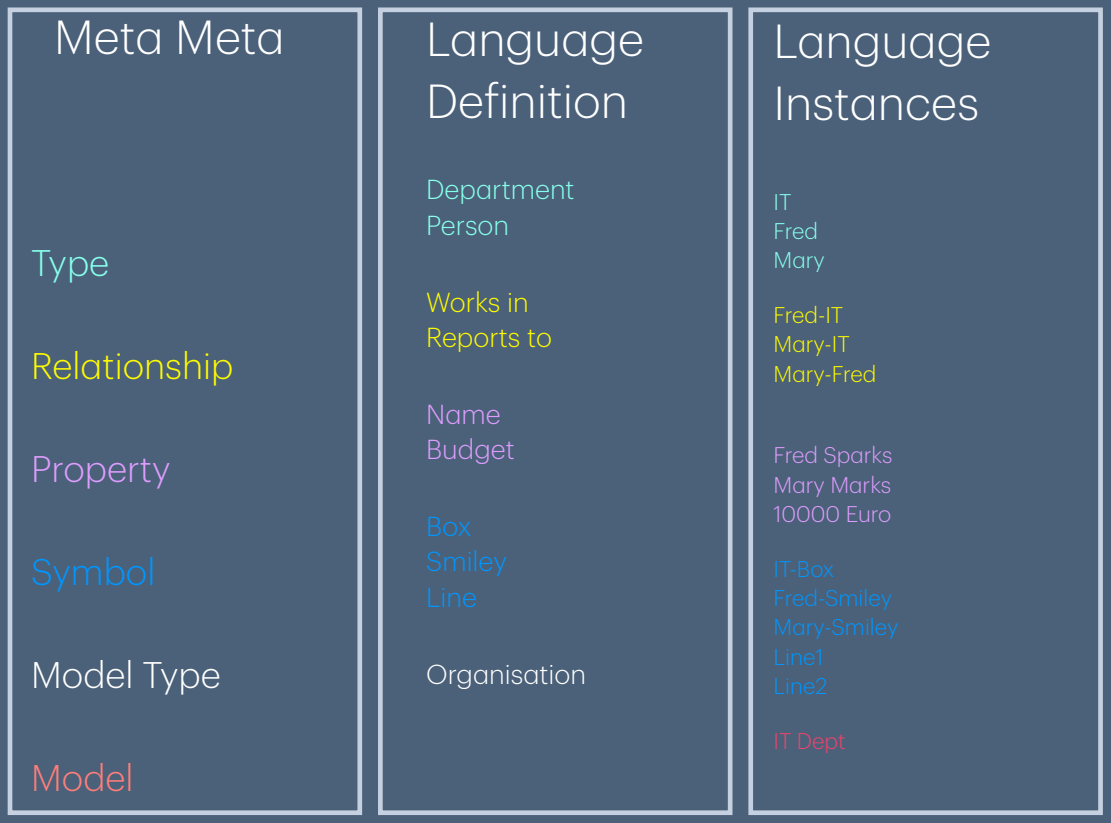
Manages model instances and their relationships to the repository instances they represent

Manages model relationships and the relationships between instances they represent

Note that model type definition is held as XML property

Note that model visual only information such as position, size, colour override, relationship path is stored in XML property

Vector symbols and line characteristics are defined in a Logo derived script held on symbol instances



inspired!

EVA GM Model Handling

Manages how a Model Type is defined in terms of Types to include and what symbol should be used to represent them (in Model Type diagram) and instances (in Models)

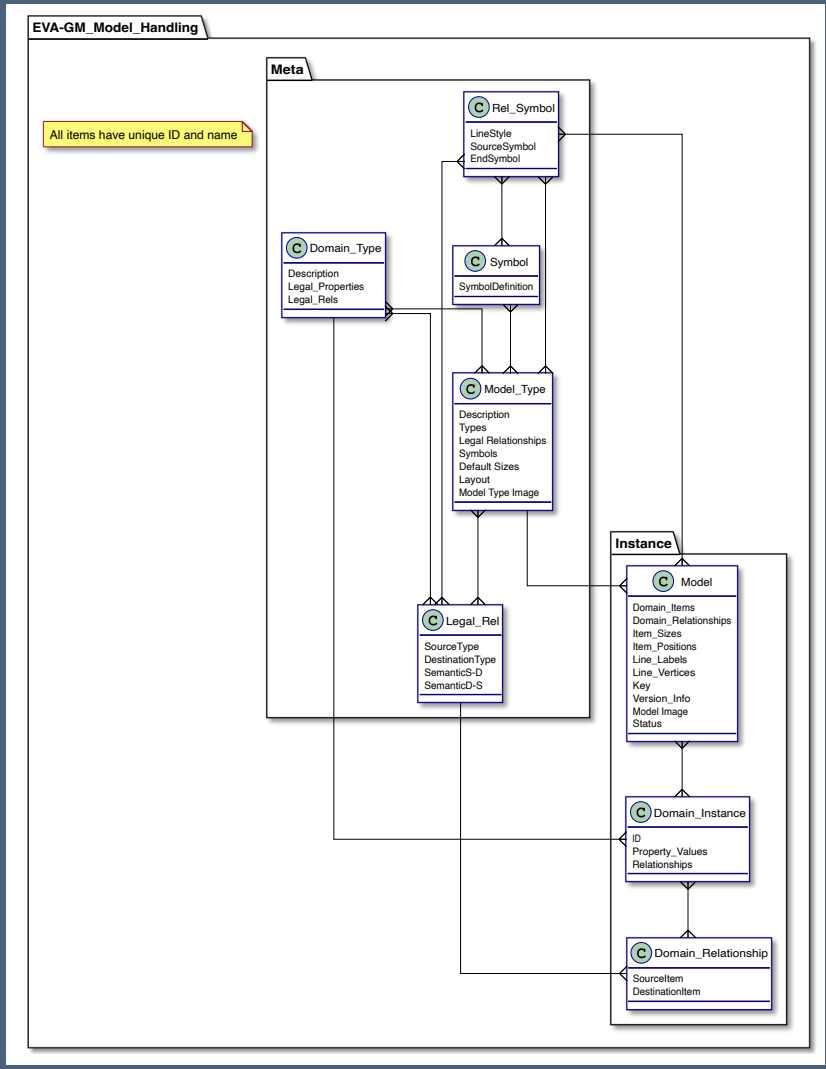
Manages model instances and their relationships to the repository instances they represent

Manages model relationships and the relationships between instances they represent

Note that model type definition is held as XML property

Note that model visual only information such as position, size, colour override, relationship path is stored in XML property

Vector symbols and line characteristics are defined in a Logo derived script held on symbol instances



inspired!

Language Server Protocol (LSP)

- Languages have proliferated in recent years
- Development of IDEs was often lacking, or at least very labour intensive
- Platforms such as the Java Virtual Machine and .Net Common Language Runtime used to support execution of multiple languages
- IDEs started to support many languages
  - Needed code completion, syntax highlighting etc.
  - Should not embed all the language knowledge with UI and Tools
- Microsoft and others implemented the Language Server Protocol to separate the concerns

inspired!

Language Server

- Understands the language semantics and syntax rules
- Provides source to an IDE
- Processes changes made by user to update source
- Provides modified code for re-rendering

Editor Client

- Manages User Interfaces
- Provides code completion, syntax highlighting
- Deals with editing actions
- Renders the provided source code
- Can be desktop or web based

inspired!

GLSP Architecture

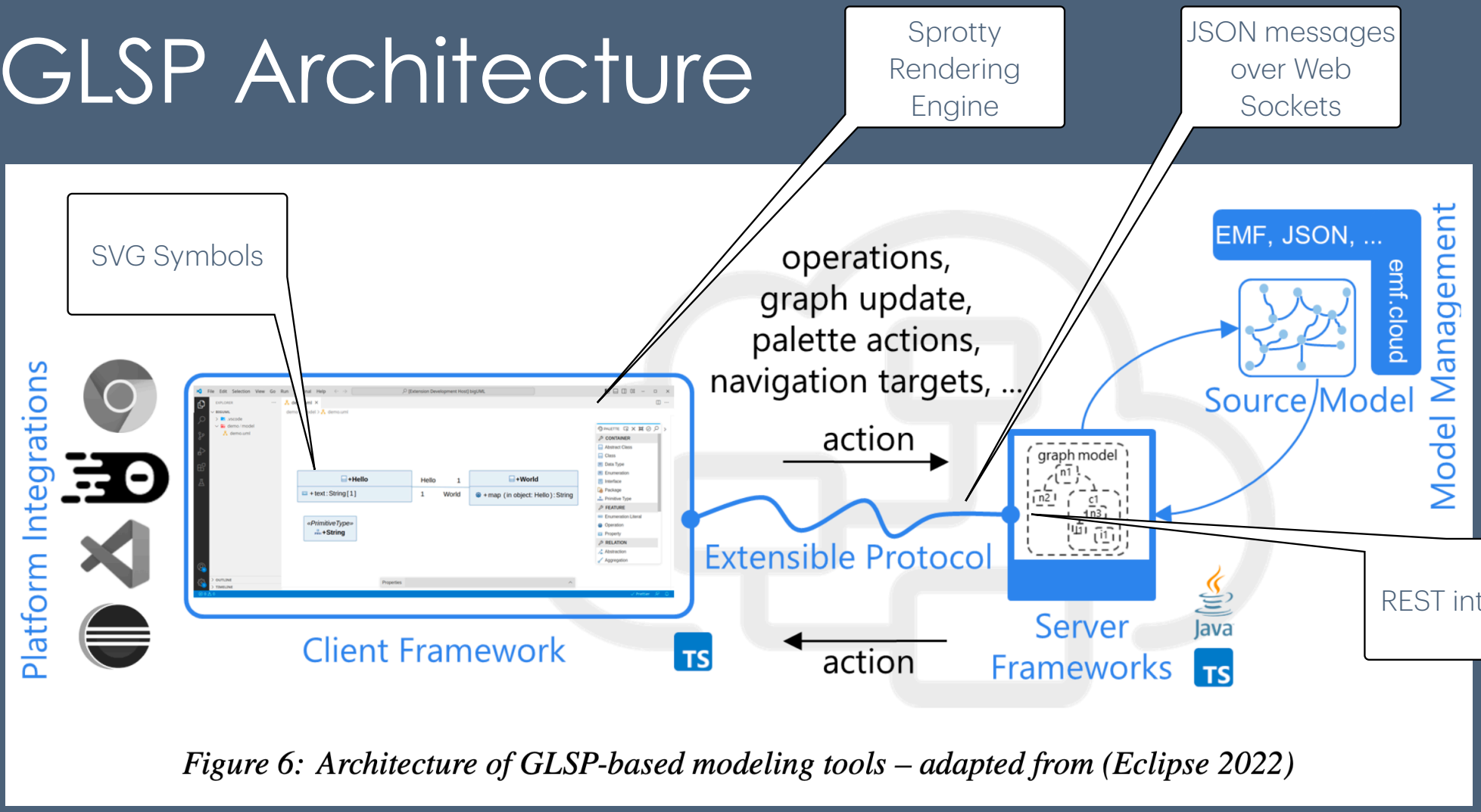
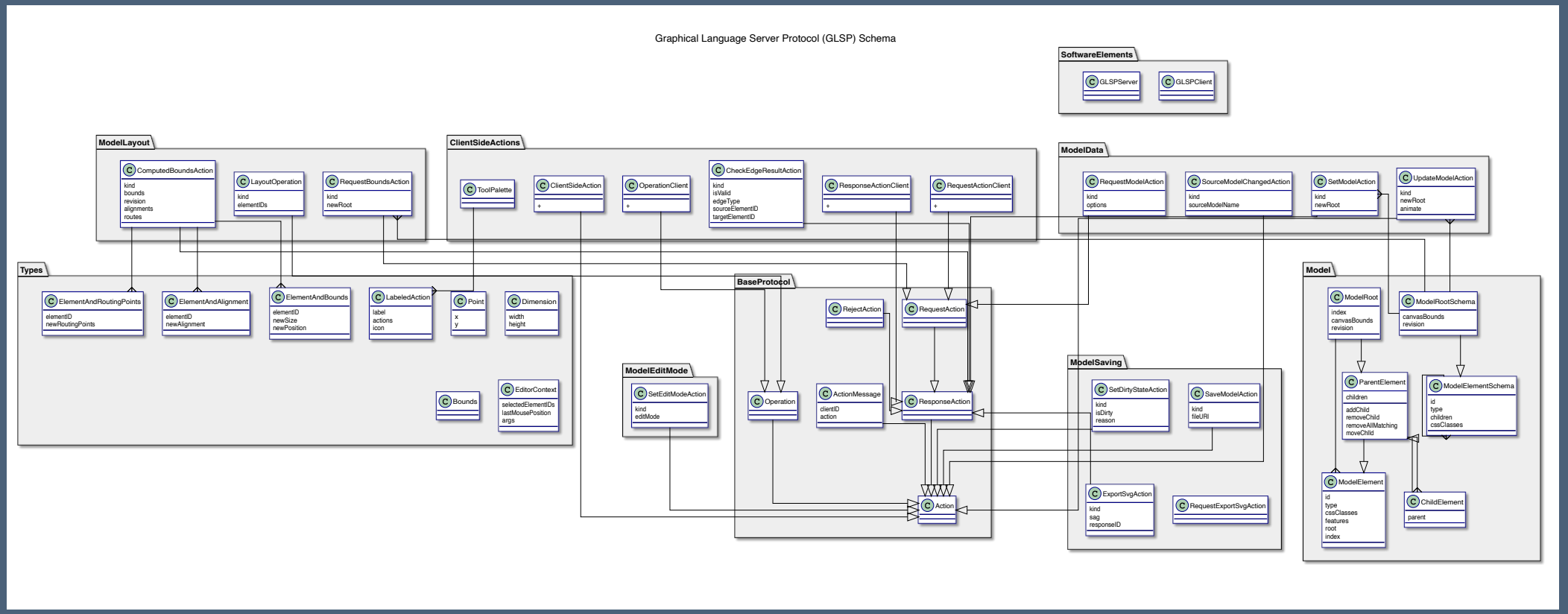


Figure 6: Architecture of GLSP-based modeling tools – adapted from (Eclipse 2022)

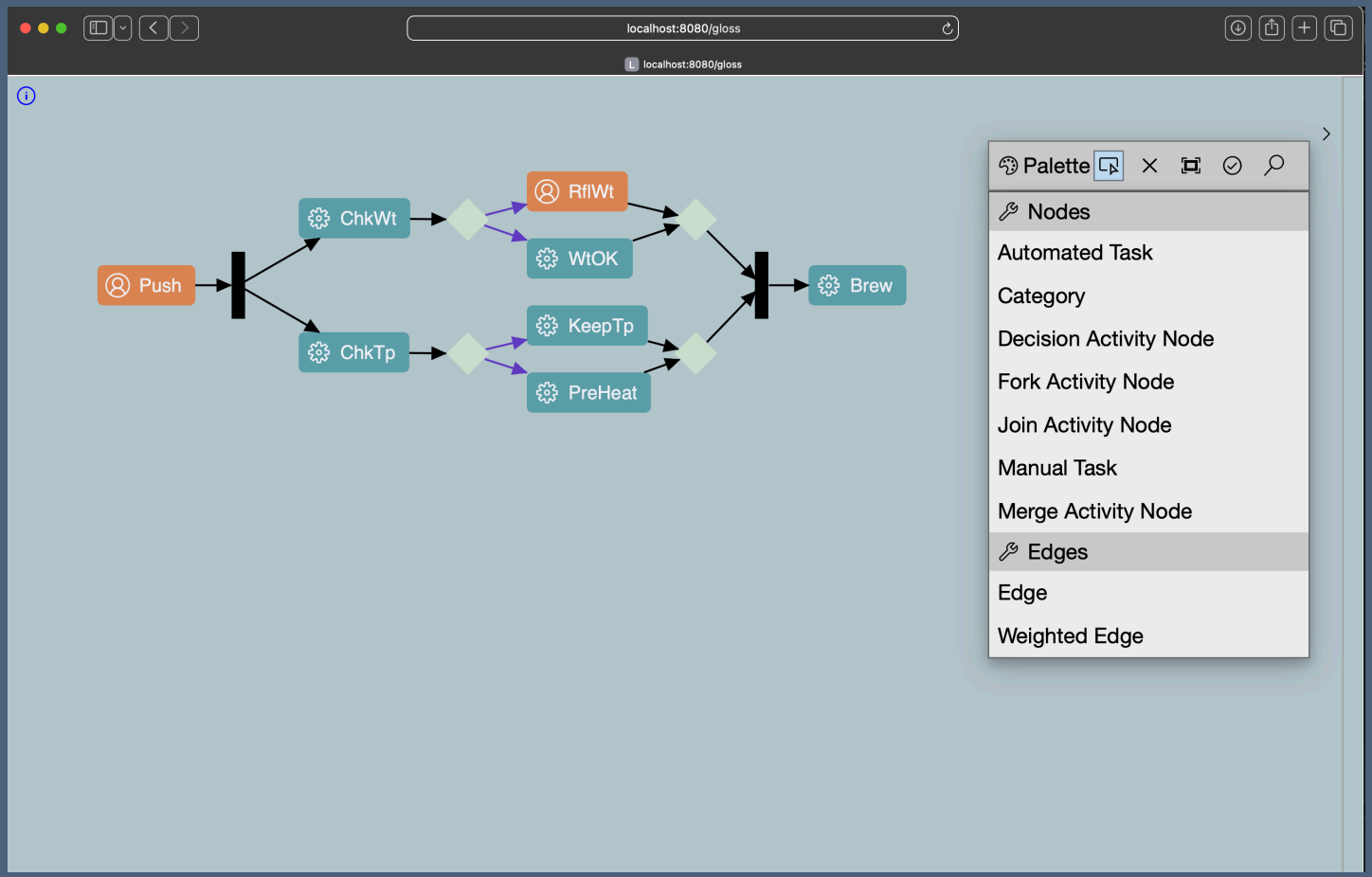
inspired!

GLSP Data Model



inspired!

GLSP Client for Workflow



inspired!

Our Project

- Examined the documentation for GLSP
- Installed and ran the example application (TypeScript)
- Inspected the objects in the browser and the messages sent to and received from server
- Implemented Classes in Smalltalk to hold the model, respond to the requests / actions
- Added ability to hold multiple models
- Added ability to serve the symbols from the server
- Added container UI elements to allow selection of model type and model
- Allowed model to be saved under different name, or deleted

inspired!

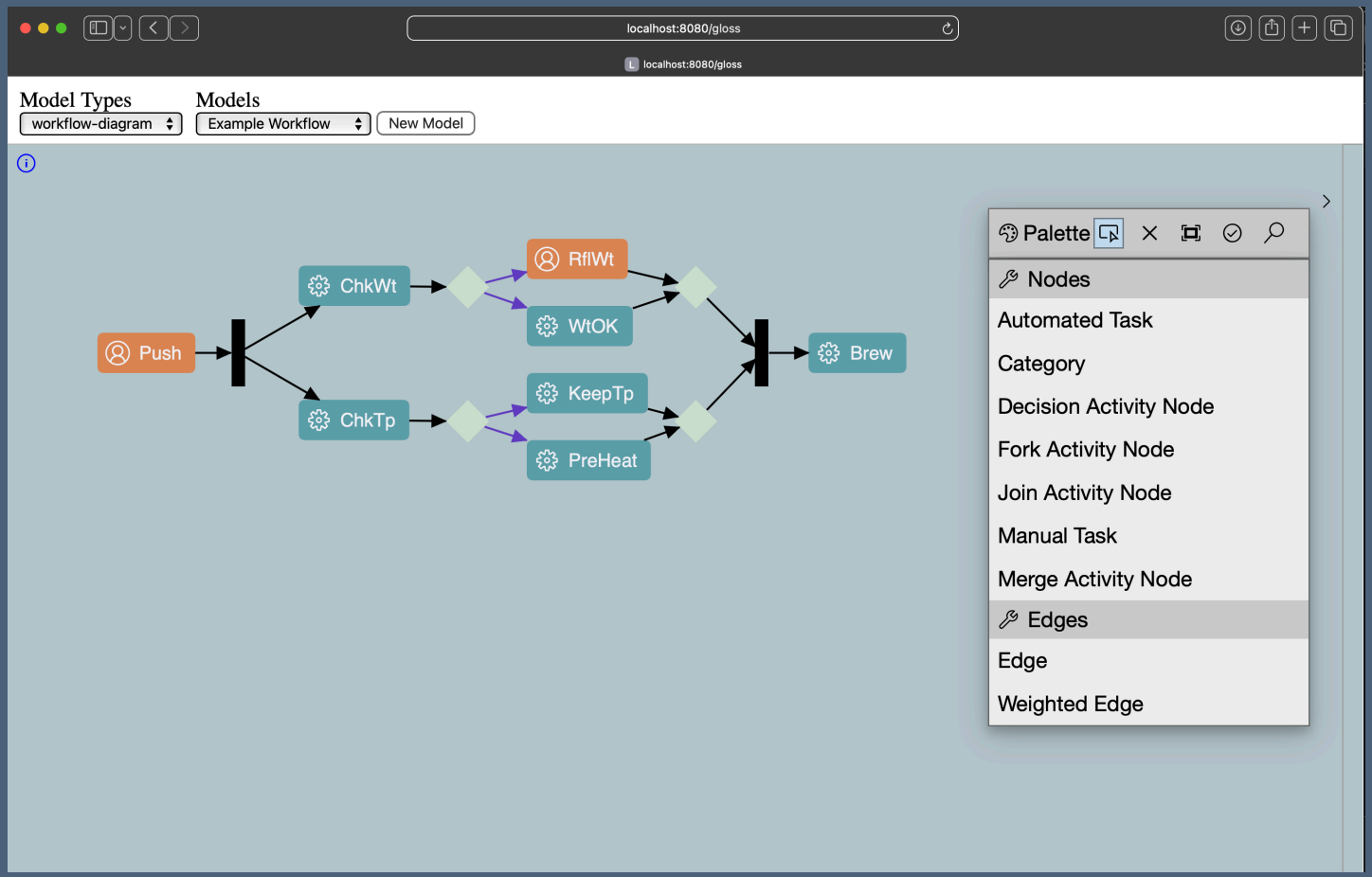
Technology and Architecture

- Pharo as IDE
- Zinc for HTTP and Web Sockets
- Teapot for minimal Web Server / REST
- Roassal to visualise classes
- NeoJSON to serialise and deserialise JSON

inspired!

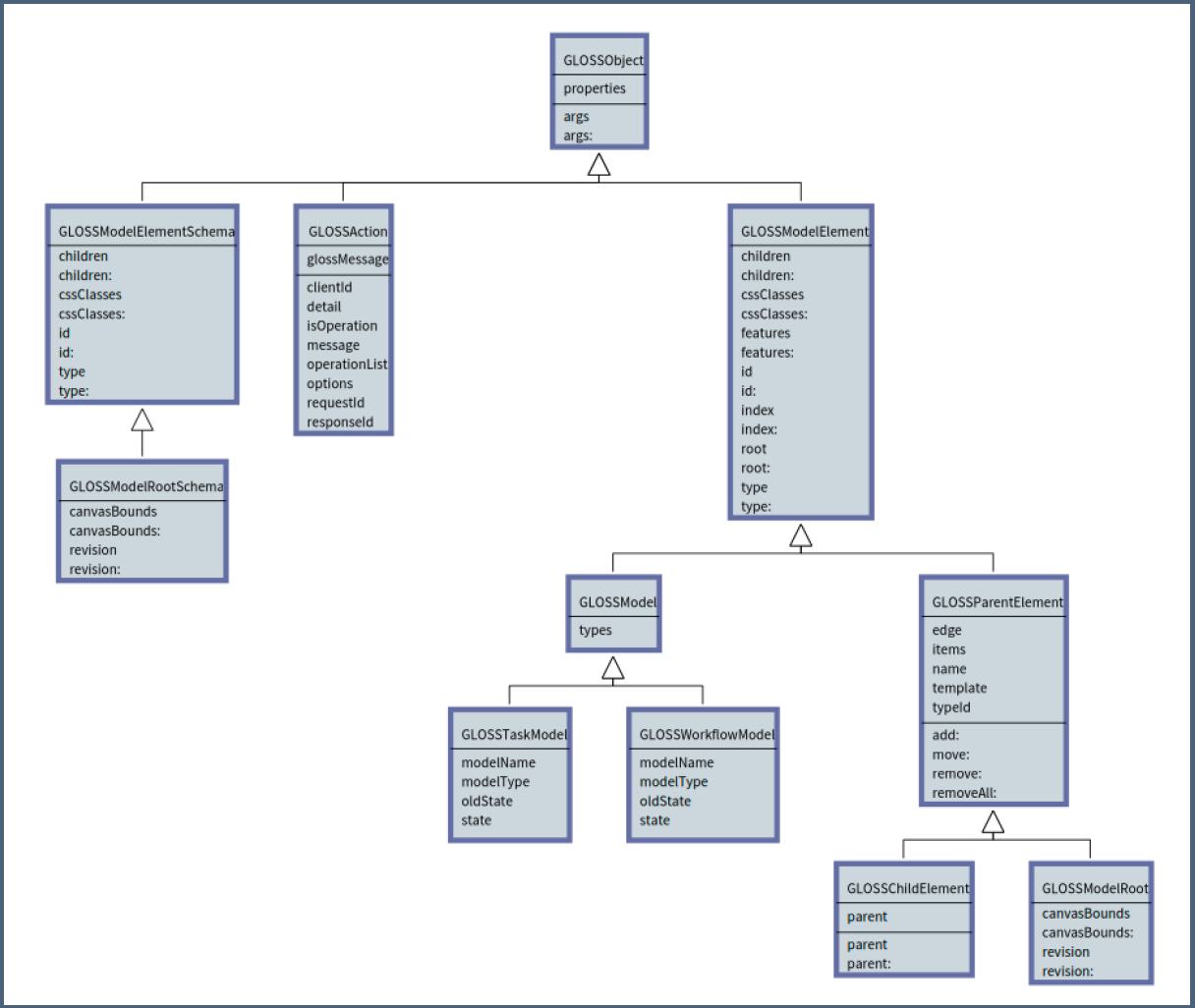
GLSP Client for Workflow

[Demo]



inspired!

GLOSS Classes



inspired!

GLSP Recommendations (1)

	Limitation	Suggested Remedy
1	No explicit understanding of a Model Type	Extend protocol to include notion of a Model Type having a defined set of concepts, relationships and associated symbols. This should be achieved in data rather than code.
2	Ditto	Provide for requests for list of Model Types by client and response by server
3	No provision for dealing with multiple model instances	Extend protocol to allow for requesting models of a given type
4	Symbols defined in code and in client	Define symbols in data (still SVG) and on server. Provide for transfer of required symbols for model type to client
5	Symbol types limited to SVG	Could extend to allow lists and images
6	Does not allow re-use of model item across model instances (naming issue; item not managed separately from model)	Manage items as objects with own identity. Associate with models by reference.

inspired!

GLSP Recommendations (2)

	Limitation	Suggested Remedy
7	Does not support including items already on server in models.	Extend protocol to allow fetching item instance lists to facilitate this behaviour
8	Does not support fetching item details for editing purposes	Extend protocol to allow fetching item details and returning captured details to server. These could be exchanged as a JSON map.
9	No support for identifying related items to facilitate rapid visual model construction	Extend protocol to allow fetching items related to current focus item (name and type) where these are consistent with the current model type (permitted types and relationships)
10	Position and size of items is held as property of item, not per model. Inhibits reuse of item in multiple models	Refactor item location, size, etc which varies per model to the model. Keep domain details of item with item.
11	Does not provide for “expansion” of a model element to a drill down model	Extend the protocol to allow such a request from the client to the server and recording the related model with the item in the upper model.

inspired!

Project Observations (1)

- Example GLSP Implementations in Javascript and Typescript
- We worked with the Typescript example
- GLSP architecture assumes a client deals with only one model type!
- Connection between a concept and representation is realised in the client and is hard coded
- Symbols are implemented as functions returning SVG, again hard coded
- There is no explicit meta data about model type, mapping of concepts to visual representation etc.
- No edit of model elements (can probably be done via other Eclipse / LSP features, but not in examples)
- Efficiency and Responsiveness - Full model render for every change
- Low level of abstraction - Class per concept in the model type
  - Symbols hard coded for a given concrete syntax (Function per symbol with embedded SVG paths)
  - Large number of classes to deal with Actions and Responses (Class each - could just be methods on a couple of classes)
  - Only dealt with 1 endPoint, 1 Model Type, 1 Notation

inspired!

Project Observations (2)

	TypeScript	**Smalltalk**
Server	58k SLOC / 7,6 Mb	3k SLOC / <1 Mb
Client.	95k SLOC / 9,8 Mb (1 model type)	20k SLOC ~400k (unlimited model types, >> functionality)

- Our own antique Graphical Modeller
  - Handles many model types
  - Does meta and instance models
  - Does arbitrary (user extensible) symbols
  - Supports multiple symbol styles (vector, raster, container, list)
  - Has advanced features (Sessions and security, expand node as model, alignment, size, summarise models, visually compare models, launch node editing UI, store rendered model on server,§ etc.)
  - <20k SLOC

inspired!

Future Work

- Extend GOSS to return Smalltalk class properties and relationships allowing suitable client to display Smalltalk models (beyond just inheritance tree)
- May port the GLSP support to our EVA Server
- May use GLSP or extension thereof to allow our GM to access GLSP servers
- Could create our own GLSP client in the browser, but this will require vector graphics support (e.g. via CUIS and Morphic 3 or similar)
- Collaborate?

inspired!

References(1)

[1] J.R. Bourne, Object-Oriented Engineering: Building Engineering Systems Using Smalltalk-80. CRC Press, 1992

[2] D. Bork and P. Langer, Catchword: Language Server Protocol - An Introduction to the Protocol, its Use, and Adoption for Web Modeling Tools. Enterprise Modeling and Information Systems Architectures, Vol 18, No 9 2023.

[3] D. Bork, P. Langer and T. Ortmayr, A Vision for Flexibile GLSP-based Web Modeling Tools. IFIP Working Conference on The Practice of Enterprise Modeling 2023.

[4] Stephane Ducasse et al, Meta-Environment and Executable Meta-Language Using Smalltalk: an Experience Report. Software and Systems Modeling, 2009.

[5] R. Rodriguez-Echeverria et al, Towards a Language Server Protocol Infrastructure for Graphical Modeling. 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, 370-380, 2018.

[6] S. Sudan and M. Piel, DynaGraph: a Smalltalk Environment for Self-Reconfigurable Robots Simulation. European Smalltalk User Group (ESUG'04), Köthen, Germany, 2004.

[7] Sprotty.org, Diagram Visualization Tools for Your Web Applications, Open Source Community, https://sprotty.org, 2023.

[8] T. Thomasma and O.M. Ulgren, Modeling of a manufacturing cell using a graphical simulation
system based on Smalltalk-80. In Proceedings of the 19th conference on Winter simulation (pp. 683-691). December 1987.

[9] B.T.M. Anh, S. Stinckwich, M. Ziane, B. Roche, and H.T. Vinh. KENDRICK: A Domain Specific
Language and platform for mathematical epidemiological modelling. In The 2015 IEEE RIVF International Conference on Computing & Communication Technologies-Research, Innovation, and Vision for Future (RIVF) (pp. 132-137). January 2015.

[10] D. Sternberg, F. Budinsky, E. Merks and M Paternostro, EMF: Eclipse Modeling Framework. Pearson Education. 2008.

[11] N. Gunasinghe and N. Marcus, Language Server Protocol and Implementation. Apress. 2021.

[12] Microsoft, Language Server Protocol Overviews. https://microsoft.github.io/language-server-
protocol/overviews/lsp/overview. Accessed Mar 2024.

[13] Inspired.org, Enterprise Value Architect : https://www.inspired.org/eva-home. 2022.

[14] Benoit Verhaeghe, Pharo Language Server: https://github.com/badetitou/Pharo-LanguageServer, accessed Mar 2024.

[15] Sven van Caekenberghe, Zinc: an industrial quality level HTTP/S framework. http://books.pharo.org/booklet-Zinc/pdf/2020-03-23-Zinc.pdf accessed May 2024.

[16] Damien Cassou, Stéphane Ducasse, Luc Fabresse, Johan Fabry, Sven van Caekenberghe, Enterprise Pharo a Web Perspective. http://files.pharo.org/books-pdfs/entreprise-pharo/2016-10-06-EnterprisePharo.pdf accessed April 2024.

[17] Damien Cassou, Stéphane Ducasse, Luc Fabresse, Johan Fabry, Sven van Caekenberghe. Enterprise Pharo a Web Perspective. Square Bracket Associates, 2015. hal-01223026v2

[18] Patrick Rein and Christoph Thiede, Squeak by Example (Edition 6). Software Architecture Group, Hasso Plattner Institute, Germany. 2023.

[19] Philipp-Lorenz Glaser, Developing Sprotty-based Modeling Tools for VS Code. BACHELOR'S
THESIS. Faculty of Informatics TU Wien. 2022.

inspired!

References(1)

[20] Alexandre Bergel, Agile Visualization with Pharo. Apress. 2022

[21] Object Management Group, OMG® Unified Modeling Language® (OMG UML®) Version
2.5.1. December 2017. Accessible at: https/www.omg.org/spec/UML/

[22] Object Management Group, Business Process Model and Notation (BPMN) Version 2. January
2011. Accessible at: https://www.omg.org/spec/BPMN/2.0/PDF

[23] The Open Group, Archimate® 3.2 Specification. October 2022.

[24] Graham McLeod, An Advanced Meta-meta Model for Visual Language Design and Tooling.
EMISA Journal / Models at Work. Practice of Enterprise Modeling, Vienna, 2022.

[25] Amaud Roques, PlantUML. February 2024. https://github.com/plantuml/plantuml

[26] Ethan Cerami, Web services essentials: distributed applications with XML-RPC, SOAP, UDDI
& WSDL. O'Reilly Media, Inc., 2002.

[27] Chris Roth, Using Microsoft Visio 2010. Pearson Education, 2011.

[28] Ivan B Liss and Thomas C. McMillan. The implementation of a simple turtle graphics package. ACM SIGCSE Bulletin 19, no. 4 1987.

[29] Steven Kelly and Juka-Pekka Tolvanen. Collaborative modelling and metamodelling with
MetaEdit+. ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE, pp. 27– 34 2021.

Online Resources

[a] Instantiations.com, VAST Platform Documentation https://www.instantiations.com/vast- support/documentation/1300/#page/Welcome/welcome.html
for VA Smalltalk

[b] Pharo by Example, https://github.com/SquareBracketAssociates/UpdatedPharoByExample/ for Pharo

[c] H. Fernandes with K. Dickey and J. Vuletich. The Cuis Book. https://cuis-smalltalk.github.io/TheCuisBook/

[d] Draw2D Documentation. https://freegroup.github.io/draw2d/#/api/draw2d

[e] World Wide Web Consortium, Scalable Vector Graphics (SVG) 2. 8 March 2023. https://svgwg.org/svg2-draft/

[f] ECMA International, ECMA-404 The JSON data interchange syntax 2nd edition. December
2017. https://ecma-international.org/publications-and-standards/standards/ecma-404/

[g] Theia-ide.org, The Eclipse Theia Platform. Theia-ide.org. February 2022. https://theia-ide.org

[h] Microsoft Corp, Visual Studio Code. Version 1.89. April 2024. https://code.visualstudio.com/

[i] Whatwg.org, WebSockets Standard. websockets.spec. April 2024. https://websockets.spec.whatwg.org/

[j] OMLAB, The ADOxx Meta Modelling platform. 2024 https://www.adoxx.org/live/home

inspired!

Graham McLeod

graham@inspired.org

www.inspired.org

LinkedIn: Graham McLeod

At inspired.org we specialise in transformative business strategy integrated with executable architecture (including business architecture) and empowering methods to deliver meaningful and desirable change.

inspired!