GRAPHICAL VIEWS FOR WEB-BASED MODELING TOOLS WITH THEIA AND SPROTTY

JAN KÖHNLEIN, MIRO SPÖNEMANN
CREATE WEB-BASED TOOLS

- IDE framework
- Using LSP
- Next-gen Eclipse RCP
- Extend with custom editors and views (called widgets)
- For desktop and web
ECLIPSE SPROTTY

CREATE GRAPHICAL VIEWS

- Diagram framework
- Standalone or embedded in Theia, Eclipse IDE or VS Code
- Client and server
- Integrated with EMF / Xtext
- Base for Eclipse GLSP
CLIENT ARCHITECTURE

- **Model Source**
- **Action Dispatcher**
- **Command Stack**
- **Viewer**

Flow:
- **Model Source** → **Action Dispatcher**
- **Command Stack** → **SModel**
- **Viewer** → **Action Dispatcher**
- **Action Dispatcher** → **Model Source**
OUTLINE

▸ Semantic model vs. view model
▸ How do you determine the content of a diagram?
▸ How do you edit a model using a diagram?
MODELING TOOLS

SEPARATION OF MODEL AND VIEWS

Model

View
MODELING TOOLS

SEPARATION OF MODEL AND VIEWS

Model

Subset View
overview / filter / manual composition
MODELING TOOLS

SEPARATION OF MODEL AND VIEWS

Model

Multiple Views

aspects / projections / viewpoints / perspectives
MODELING TOOLS

SEPARATION OF MODEL AND VIEWS

Semantic Model

Transformation (M2M)

View Model

overview / filter / manual composition
MODELING TOOLS

SEPARATION OF MODEL AND VIEWS

Semantic Model
server

SModel

Sprotty
client
"mindsKind": "Class",
"isAbstract": false,
"layout": "vbox",
"selected": false,
"position": {
  "x": 2136.560450164477,
  "y": 12
},
"size": {
  "width": 62.5625,
  "height": 47.375
},
"layoutOptions": {
  "paddingTop": 0,
  "paddingBottom": 0
},
"type": "node:Class",
"id": "Class-Eclipse:Entity",
"children": [
  {
    "layout": "vbox",
    "position": {
      "x": 8,
      "y": 0
    },
    "size": {
      "width": 46.5625,
      "height": 47.375
    }
  }
]
DIAGRAM AUTHORING: PERSISTENCE, LAYOUT, RECONCILING
WHAT'S SHOWING IN THE DIAGRAM?

**Full model**
- mapping
- lifecycle hooks

**Filtered submodel**
- GUI for filter rules
- persistence of rules

**Fully Manual**
- GUI for composition
- model browser
- persistence
- reconciliation
WHAT'S SHOWING IN THE DIAGRAM?

**Fully Manual**
- GUI for composition
- model browser
- persistence
- reconciliation

**Graphical Editing**
- GUI for editing
- sync back to model
FULL MODEL DIAGRAM
FULL MODEL DIAGRAM

SEMANTIC MODEL → TRAFO → SMODEL

USER CHANGE
MANUAL DIAGRAM - EXTERNAL FILE

SEMANTIC MODEL → DIAGRAM CONFIG → TRAFO → SMODEL
MANUAL DIAGRAM - IN MODEL

SEMANTIC MODEL

DIAGRAM CONFIG

TRAFO

SMODEL
Don't store the SModel

- reduce to things the user really wants to change
- text-based formats to allow versioning, merging etc.

Consider putting small information in the semantic model

- Fully-automated views don't need to be persisted
Macro-layout to arrange nodes and route edges

Sprotty uses Eclipse Layout Kernel (ELK)

Modes

- fully automatic
- manual/on-demand
- ELK interactive
RECONCILING

- Decide when (not) to reconcile
- Provide „stable“ IDs/traces
- Fully-automated views don't need any reconciling
GRAPHICAL EDITING
BI-DIRECTIONAL MAPPING

SEMANTIC MODEL

MAPPING TRAFO

SMODEL

INVERSE MAPPING TRAFO

USER CHANGE
BI-DIRECTIONAL MAPPING

SEMANTIC MODEL → MAPPING TRAFO → INVERSE MAPPING TRAFO → SMODEL
BI-DIRECTIONAL MAPPING

SEMANTIC MODEL

MAPPING TRAFO

INVERSE MAPPING TRAFO

SMODEL
BI-DIRECTIONAL MAPPING

GRAPHICAL EDITING
GRAPHICAL EDITING

BI-DIRECTIONAL MAPPING

SEMANTIC MODEL

AVOID BI-DIRECTIONAL SYNC

SMODEL
GRAPHICAL EDITING

UNI-DIRECTIONAL MAPPING

SEMANTIC MODEL → MAPPING TRAFO → SMODEL

USER CHANGE
UNI-DIRECTIONAL MAPPING

SEMANTIC MODEL

MAPPING TRAFO

SMODEL

USER CHANGE
GRAPHICAL EDITING

UNI-DIRECTIONAL MAPPING

SEMANTIC MODEL

MAPPING TRAFO

SMODEL

SEMANTIC CHANGE
GRAPHICAL EDITING

UNI-DIRECTIONAL MAPPING

SEMANTIC MODEL → MAPPING TRAFO → SMODEL
CONSIDERATIONS

- Avoid bi-directional mapping
  - Map user changes to semantic model changes
  - e.g. LSP code actions, content assist, rename
- Maybe diagram content editing is enough
KEEP IT SIMPLE!

- Ask yourself (or your team / customers):
  - What problems do we want to solve with diagrams?
  - What user interaction patterns are most suitable in our scenario?
EVALUATE THE SESSIONS
Sign in and vote using the conference app or eclipsecon.org
REFERENCES

- https://theia-ide.org/
- https://github.com/eclipse/sprotty
- https://github.com/TypeFox/npm-dependency-graph
- https://github.com/TypeFox/theia-xtext-sprotty-example
- https://www.youtube.com/watch?v=hrGc30iIdiA
- https://www.youtube.com/watch?v=IrFQKdjzvXU