DSLs in the Cloud – with Eclipse Technologies

Jan Köhnlein
Miro Spönemann

@jankoehnlein
@sponemann

TypeFox
The Aim: A DSL Tool in the Browser

- Code Generator
- Graphical View
- Workspace and Tool Integration
- Intelligent DSL Editor
- Content Assist
- DSL Validation
Architecture

Browser

IDE Application

Code Editor

Graphical View

Server

Language Server

Text Editor Services

Workspace Services

Graphics Extensions

Language Server Protocol
Technologies

Browser

- Theia Application
- Text Editor
- Sprotty View

Server

- Language Server
- Xtext LS
- Sprotty LS Extension
- ELK Layout

All open-source from eclipse
Next: Xtext Language Server

Browser

- Theia Application
  - Text Editor
  - Sprotty View

Server

- Language Server
  - Xtext LS
  - Sprotty LS Extension
  - ELK Layout

Isp4j
Building a DSL with Xtext

(1) Create Xtext project with LSP support (via lsp4j)
(2) Specify an Xtext Grammar
(3) Write a Code Generator
(4) Generate the Language Server
A Simple StateMachine DSL with Xtext

```plaintext
grammar io.typefox.examples.theia.states.States
  with org.eclipse.xtext.common.Terminals

generate states
  "http://www.typefox.io/examples/theia/states/States"

StateMachine:
  'statemachine' name=ID
  (states+=State | events+=Event)*;

State:
  'state' name=ID transitions+=Transition*;

Event:
  'event' name=ID;

Transition:
  event=[Event] '=>' state=[State];
```

```plaintext
datamachine MrsGrantsSecretCompartment

event
doorClosed
drawOpened
drawOpened
drawOpened
lightOn
panelClosed

state active
  lightOn => waitingForDraw
drawOpened => waitingForLight

state idle
  doorClosed => active

state waitingForLight
  lightOn => idle

state waitingForDraw
  drawOpened => unlockedPanel

state unlockedPanel
  panelClosed => idl
```
Make it Nice with **Xtext**

1. Scoping
2. Validation
3. Formatter
4. Fine tune content assist
5. Code Templates
6. …
Xtext Language Server

Browser
- Theia Application
  - Text Editor
  - Sprotty View

Server
- Language Server
  - Xtext LS
  - Sprotty LS Extension
    - ELK Layout

Isp4j
Next: Theia Application
THEIA

- Cloud and Desktop IDE Framework
- Written in TypeScript

Frontend

<table>
<thead>
<tr>
<th>Browser Icons</th>
</tr>
</thead>
</table>

| File System |

Backend

JSON-RPC
Theia Extensions

Frontend

- Core Extension
- Workspace Extension
- Git Extension
- ... 

Backend

- DSL Theia Extension

Frontend DI Container

Backend DI Container
Scaffolding a Theia Extension

```bash
$ yo theia-extension
? The extension's name states-dsl
StatesDsl
  create package.json
  create lerna.json
identical .gitignore
  create README.md
identical .vscode/launch.json
  create states-dsl/package.json
  create states-dsl/tsconfig.json
  create states-dsl/src/browser/states-dsl-frontend-module.ts
  create states-dsl/src/browser/states-dsl-contribution.ts
  create browser-app/package.json
  create electron-app/package.json

yarn install v1.8.0
info No lockfile found.
[1/4]🔍 Resolving packages...
[2/4]🚚 Fetching packages...
[3/4]🔗 Linking dependencies...
[4/4]📃 Building fresh packages...
success Saved lockfile.
$ lerna run prepare
...

lerna success run Ran npm script 'prepare' in packages:
lerna success  - states-dsl
lerna success  - browser-app
lerna success  - electron-app
✨ Done in 57.10s.
```
Theia Application

Browser

Theia Application

Text Editor

Sprotty View

Language Server

lsp4j

Xtext LS

Sprotty LS Extension

ELK Layout
Next: DSL Theia Extension
Integrate LS and Theia Extension

(1) Copy the LS binary to the extension

(2) Register language to the frontend DI Module

(3) Register language to the backend DI Module
Make it Nice

(1) Syntax highlighting
   (TextMate grammar)

(2) Bracket matching, folding
   (Monaco config)
DSL Theia Extension

Browser
- Theia Application
  - Text Editor
  - Sprotty View

Server
- Language Server
  - Xtext LS
  - Sprotty LS Extension
  - ELK Layout

_lsp4j_
Next: Add Diagrams

Browser

Theia Application

Text Editor

Sprotty View

Server

Language Server

Xtext LS

Sprotty LS Extension

ELK Layout

Isp4j
Sprotty

- A web-based diagramming framework
- TypeScript, SVG, CSS
- Supports LSP
- Moving to Eclipse
Sprotty

CommandStack

Command

SModel

ActionDispatcher

Action

Viewer
Sprotty Client-Server

CommandStack

SModel

ActionDispatcher

Viewer

Action

Command

Action

Diagram Server
Sprotty LSP
Theia-Sprotty

Theia Widget

Theia Backend

CommandStack

ActionDispatcher

SModel

Viewer

Diagram Server

Diagram Generator

LSP

Xtext LS

Command

Action
class StatesDiagramGenerator implements IDiagramGenerator {

    override generate(Resource resource, …) {
        (resource.contents.head as StateMachine).toSGraph()
    }

    def toSGraph(StateMachine sm) {
        new SGraph {
            type = 'graph'
            children = sm.states.map[toSNode] + sm.allTransitions.map[toSEdge])
        }
    }

    def toSNode(State state) {
        new SNode {
            type = 'node'
            children += new SLabel {
                type = 'label'
                text = state.name
            }
        }
    }

    def toSEdge(Transition transition) {
        new SEdge {
            sourceId = transition.fromState.name
            targetId = transition.toState.name
            type = 'edge'
        }
    }
}
Make it Nice

(1) Add custom figures

(2) Style with CSS

(3) Macro layout with Eclipse Layout Kernel (ELK)

(4) Add tracing
Questions?
And What About Graphical Editing?
Don’t do it!

Graphical Views FTW!
If it should work at all...

...it must be text first!
Sprotty: Text-first LSP Graphical Editing

Graphical edit actions => textual LSP requests

(1) Create Node => CodeAction
(2) Rename Label => Rename
(3) Rename X-Ref => CodeAssist
(4) Create Edge => WorkspaceEdit
Demo
References

- www.xtext.org
- github.com/eclipse/lsp4j
- www.theia-ide.org
- github.com/eclipse/sprotty
- github.com/eclipse/elk
- github.com>TypeFox/theia-xtext-sprotty-example
<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Poor</td>
</tr>
<tr>
<td>0</td>
<td>Neutral</td>
</tr>
<tr>
<td>+1</td>
<td>Excellent</td>
</tr>
</tbody>
</table>

Sign in and vote at [eclipsecon.org](http://eclipsecon.org)