



## UVL Playground

Final Presentation | Jannis Dommer, Stefan Vill | 10.01.2024



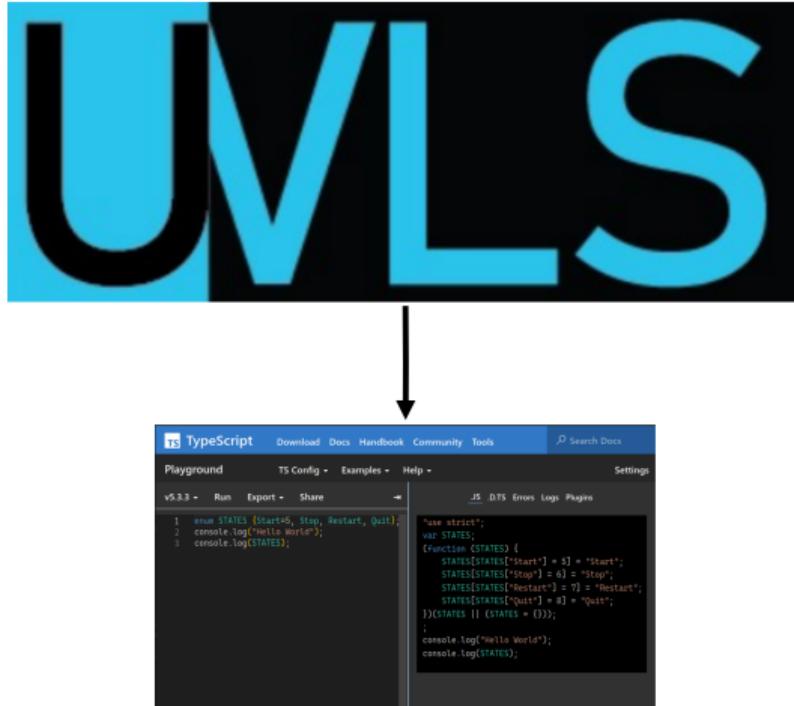
universität  
**ulm**

# Motivation

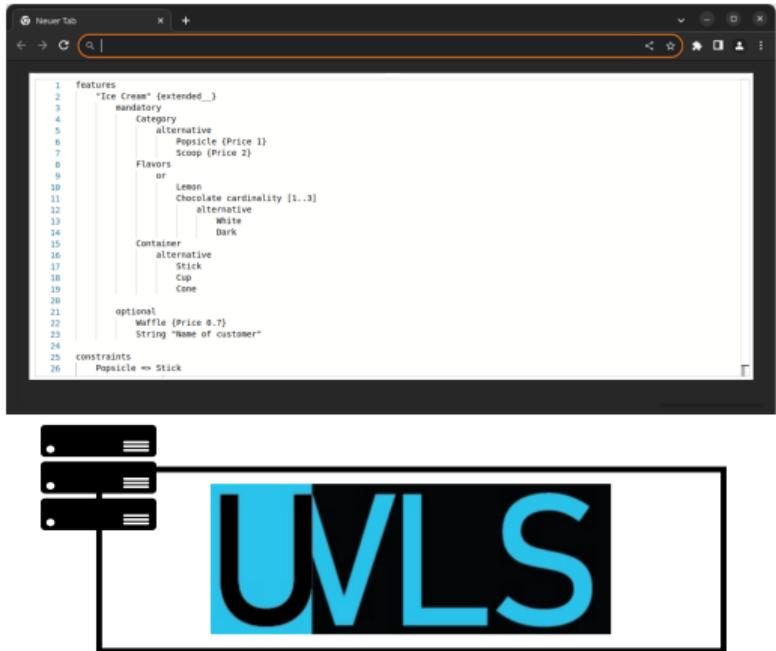
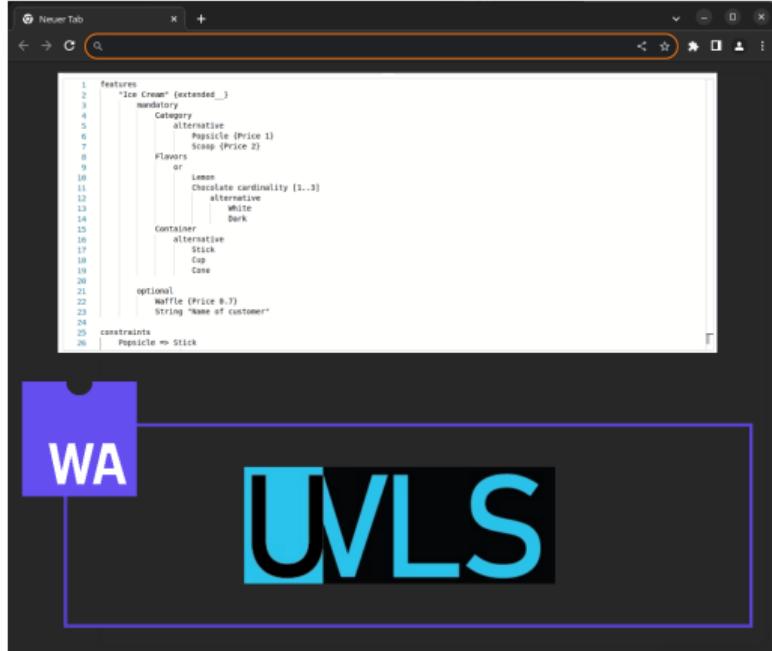
The collage illustrates the diversity of developer playgrounds:

- TypeScript Playground:** Shows a code editor with a TypeScript file containing a simple program that logs "Hello World" and defines a state object. It also shows LLVM IR and assembly code outputs.
- Brainfuck Playground:** Shows a code editor with a Brainfuck program that prints "Hello world". It includes a step-by-step debugger interface.
- UVL Playground:** Shows a code editor with a Brainfuck program. A modal window titled "UVL ist cool" is displayed, featuring a polar bear image and text about the playground's features.
- Milkdown Playground:** Shows a code editor with a Milkdown editor interface. A modal window titled "Milkdown" provides information about the editor's features, including support for GitHub Flavored Markdown and various styling options.

# Motivation



# Architektur



# Probleme mit WebAssembly

- Wenig WebAssembly kompatible dependencies
- Systemaufrufe
- UVLS enthält Webserver für config
- Treesitter mit C-Schnittstelle
- Z3 mit stdin / stdout Schnittstelle
- Module stark gekoppelt
- Erzwingt fork

# Dedizierter UVL Language Server

- Kommunikation auf Websockets umstellen
- Monaco mit Language Server Protocol ausgestattet
- Multi-User UVLS

## Infrastruktur und Tooling

- Traefik Reverseproxy als TLS Endpoint
- CI / CD Pipeline
- Sonarqube Linter



</liveDemo>

# Zusätzliche Features

- Feature Model im Darkmode
- Automatisches Update von Feature Model
- Tutorial für Playgroun und UVL
- Upload / Download
- Laden von Beispielen
- Größenlimitierung von Feature Modellen
- Speichern von Feature Modellen im Browser
- Reconnects bei Verbindungsabbruch

UVL Playground

Interactively learn the Universal Variability Language [?](#)

Get a tour of the Playground [?](#)

Select an Example [▼](#)

```
1 features
2   feature1
3   or
4     feature2
5     feature3
6
7 constraints
8   feature1
```

```
graph TD; feature1[feature1] --> feature2[feature2]; feature1 --> feature3[feature3]; feature1 --- constraint[Constraints feature1]
```