



## UVL Playground

Final Presentation | Jannis Dommer, Stefan Vill | 10.01.2024



universität  
**ulm**

# Motivation

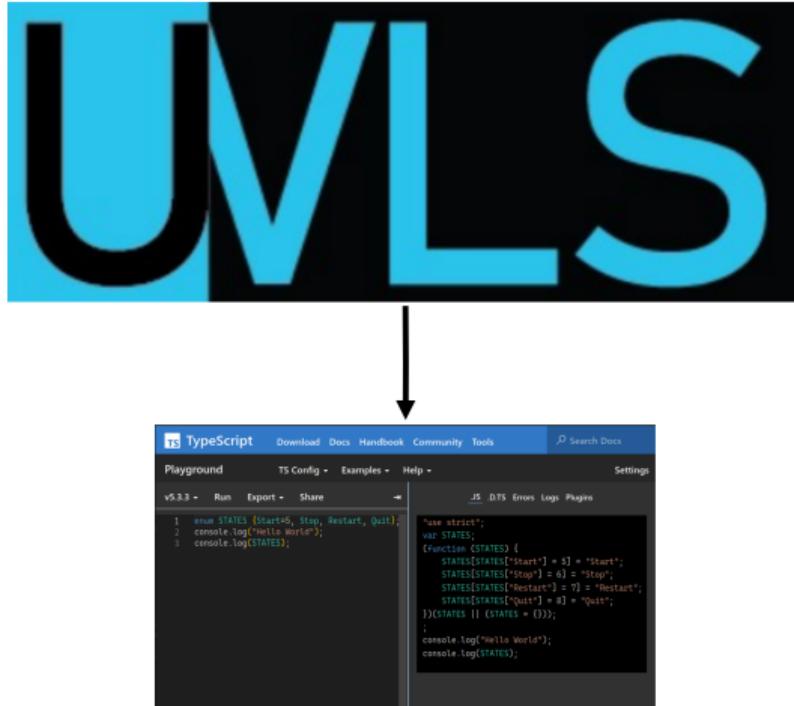
The collage consists of four screenshots:

- TypeScript Playground:** A screenshot of the TypeScript playground interface showing a code editor with the following code:

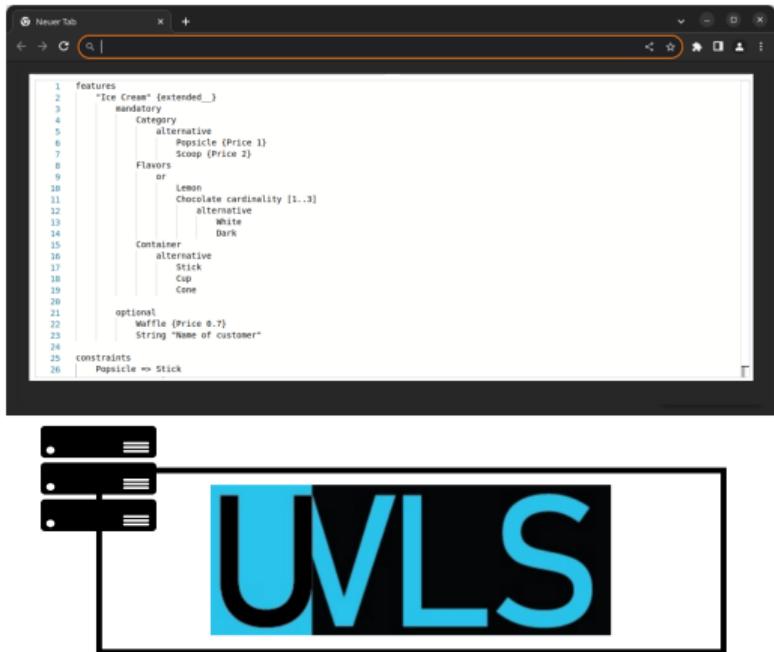
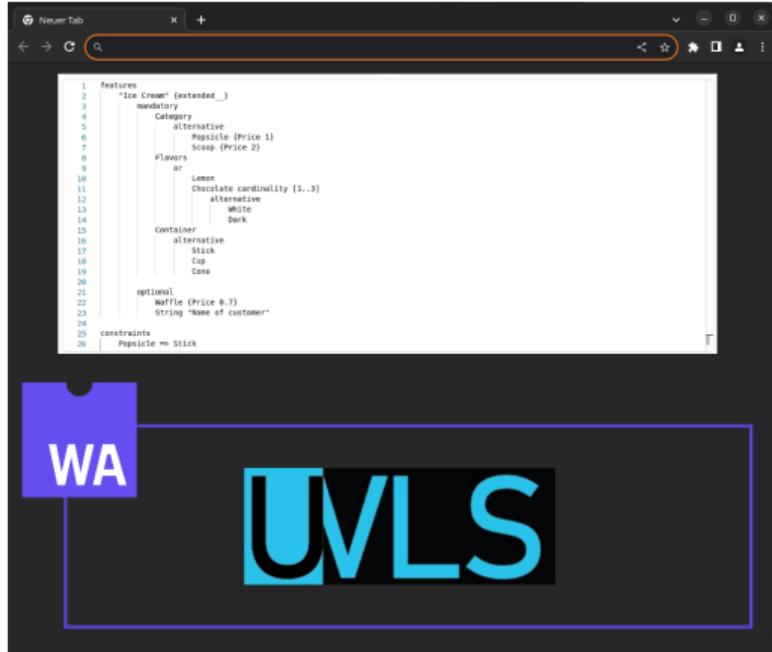
```
enum STATES { Start=5, Stop, Restart, Quit};console.log('Hello World');console.log(STATES);
```

 and a preview window showing the output "Hello World".
- LLVM IR:** A screenshot of the LLVM IR (Intermediate Representation) view, showing assembly-like code for a function named `main`. It includes annotations such as `!entry-block`, `!function-type`, and `!global`.
- UVL ist cool:** A screenshot of the UVL playground interface, which is a Brainfuck interpreter. It shows a code editor with Brainfuck code and a preview window displaying the output "Hello World".
- Milkdown Playground:** A screenshot of the Milkdown playground interface, which is a WYSIWYG Markdown editor. It shows a preview window with a polar bear image and a toolbar with various editing icons.

# 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

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

UVL Playground

Playground Tour ?

Introduction to UVL ?

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]; constraints[Constraints] --- feature1_2[feature1]
```