

Freon – A Web Native Language Development Toolkit



Jos Warmer & Anneke Kleppe
independent

LangDev Workshop, Aachen, September 2022



Freon

- Started out as ProjectIt => Projectional Editor
- Now does a lot more
- Frisian (Eng) or Friesisch (De) for “friend”

Freon Overview (readable ...)



Many aspects

- Metamodel / AST / Structure
- Projections
- Parser
- Unparser
- Scoping
- Typing
- Validation
- Interpreter
- Editor
- Server
- Etc. etc.

Implementation

- TypeScript
- Svelte



Language Development

Automatic defaults, for a quick start

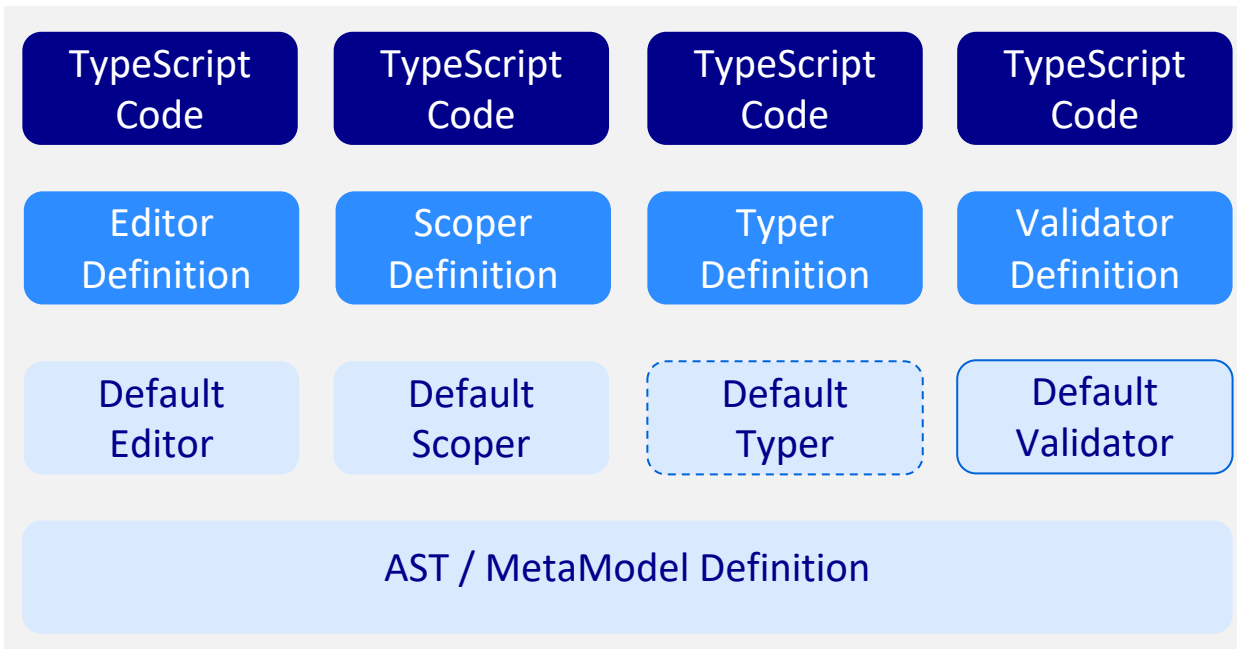
Simple definition languages for tuning (80%)

TypeScript Code for detailed tuning (20%)

*Simple things should be simple,
complex things should be possible*

Alan Kay

Stacked Architecture



Abstract Syntax Tree



Model Units

- Each model consists of model units
- Model Unit correspond to editable units
- Model units specify nodes to be referable or hidden.
- Model units can be partially loaded in browser

Main AST Structure

- Concepts
- Abstract concepts
- Interfaces
- Limited concepts (... enumerations)
- Expressions and binary expressions

Projections & Editor



Projection Boxes

- Text: editable and fixed
- Expressions
- SVG
- Flexible grids
- Tables
- Horizontal & vertical list

Styling

- CSS variables
- CSS Selectors

Search

- Text based
- Structure based
- Future: pattern based

Undo + Redo

Defining Editors



Projection:

- Text: wysiwyg

Tables

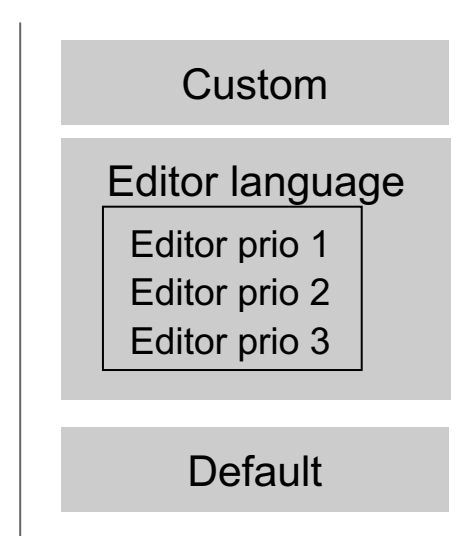
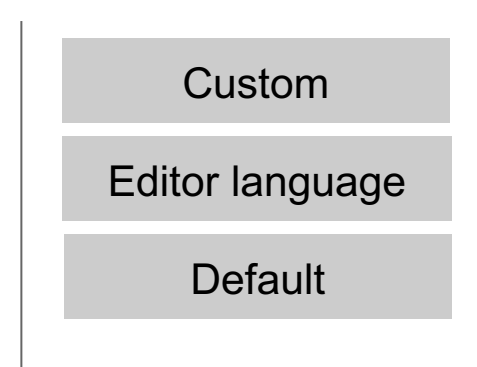
- Inspired by markdown

Multiple projections

- Prioritized
- Dynamically turn on/off
- Named editors

Other

- Triggers
- Reference shortcuts





SCSS variables

```
--freon-label-component-color: black;  
--freon-label-component-background-color: inherit;  
--freon-label-component-font-size: 14px  
--freon-label-component-font-style: normal;  
--freon-label-component-font-weight: normal;
```

SCSS selectors

```
.Entity-name-textbox {  
  --freon-text-component-background-color: lightgreen;  
}  
  
.StringLiteralExpression-value-textbox {  
  --freon-text-component-color: green;  
}
```



Projection definition

- Used to generate unparser
- Used to generate parser

Special Cases

- (un)parser cannot handle tables
- (un)parser cannot handle custom projections
 - A special projection definition named “parser” can be defined and will be used



Scoping defines what you can refer to

- Default: everything is referable from everywhere

Namespaces

- Define concepts that are namespaces
- Define additions to defined namespaces

Scope

- Scope for a reference is the namespace(s) where to look
- Define scope for concepts



Basis

- Which concepts *are* types
 - May add special “type concepts”
- Which concepts *have* a type

Infer

- Defines the type for a concept

Conformance

- Conformsto
- Equalsto



Default validations

- Mandatory properties filled
- Non optional list should have at least one element
- Names of modelunits should be valid identifiers
- Non optional references should be present

Validation language options

- Uniqueness of properties in lists
- Typechecks
- Values of primitive number properties
- Valid identifiers for all concepts
- Non empty lists
- Custom error messages

Interpreter



Interpreter framework

- Framework
- Generates all boilerplate code
- Just write one evaluation function per concept

Work In Progress

- Specify which concepts need to be evaluated (usually not all)
- Reusable expression language(s)
- Better integration with editor / app



Other Features



Import languages

- From MPS: prototype based on MPS Analyser (from Meinte Boersma)
- MPS Plugin for full customer DSL
- From Ecore or others: never tried

Implemented in TypeScript

- Components may run on client and/or server
- Allowing multiple deployment options

Language Diagrams

- Generated for each .ast file

Summary



Freon is becoming a full Language Development Toolkit

www.projectit.org (!)