Gentleman: a lightweight web-based projectional editor

Creating a language

Louis-Edouard LAFONTANT

Software Modeling & Simulation
geodes
DIRO
Université de Montréal

Introduction

- Computer science graduate specialized in software engineering
- Course instructor at University of Montreal and member of the **GEODES** Software Engineering Research Group
- Strong interest in human-computer interaction (HCI) and the exploitation of dynamic environments.
- Current research with prof. **Eugene SYRIANI**
  - Domain-specific modeling
  - Low-code engineering
  - Projectional editing

  ⇒ **Gentleman** (collab. with E. SYRIANI)

Outline

1. Background
2. Gentleman
3. Demonstration
4. Outlook
5. Questions

Gentleman

MODEL

Domain-specific modeling

- Raising the level of abstraction
  - Solution specified directly using problem domain concepts
  - Software development as a **model-driven activity**
- Leveraging the expertise and knowledge of many experts instead of relying solely on technical experts
  - ✓ Make software more accessible and inclusive
  - ✓ Better productivity and improved quality

However, the adoption is not as widespread as envisioned. **Why?**

Language workbench

Tools that support the efficient definition, reuse, and composition of languages and their IDEs.

Editing environments

- **Parser-based editor (free-form, syntax-directed)**
  Xtext, MetaEdit+, AToMPM
- **Projectional editor: no-parser, multiple notations**
  MPS, WholePlatform

Recurring problems

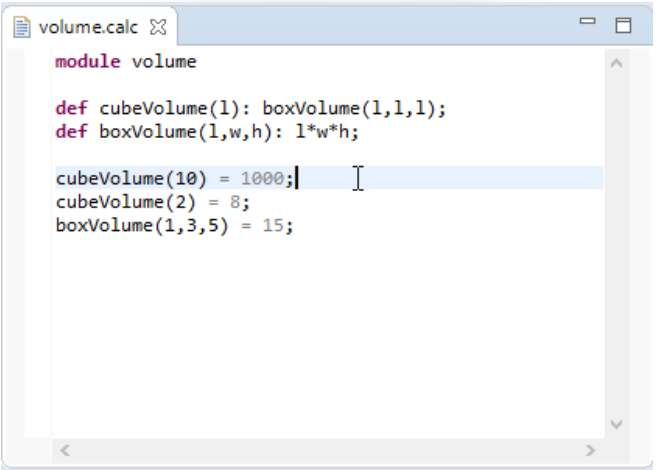
× **Rooted in OOP**

× **Heavy-weight**

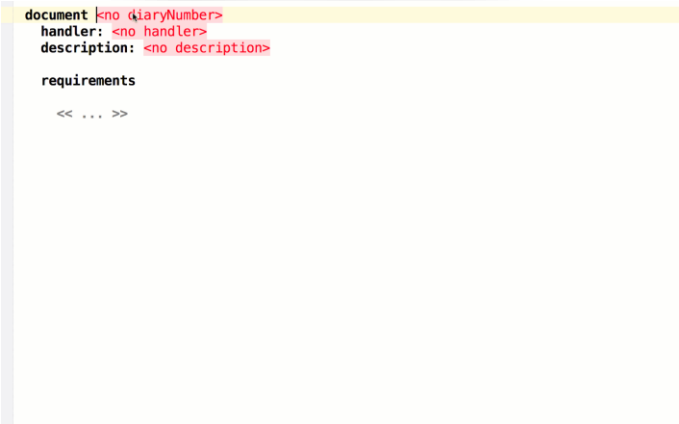
× **Platform-specific**

× **Poor usability**

Xtext



MPS



How can we improve on the situation?

1

Define concepts with **structures and languages unrelated to any programming code** concept (~~UML, OO~~)

2

Move MDE/DSM to a **user-driven approach**, instead of a technology-driven approach

3

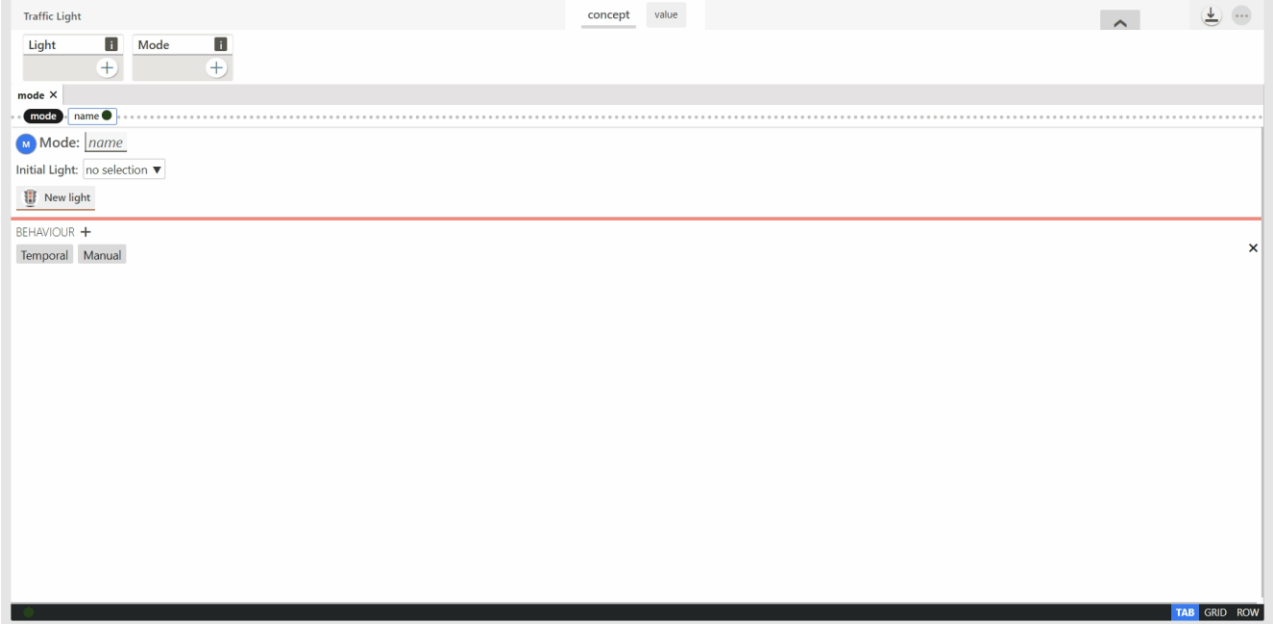
Take **projectional editing to the web** and provide a **lightweight** solution

Gentleman

Goal: Making modeling more accessible to domain experts

Projectional editing

- **Concept**: structure to define the model
- **Projection**: visuals to interact with the model



Features

✓ Lightweight and minimalistic design

✓ Web solution: no installation required

✓ Easy integration

✓ Built with a user-centric approach

✓ Compatible with Ecore model

Concept

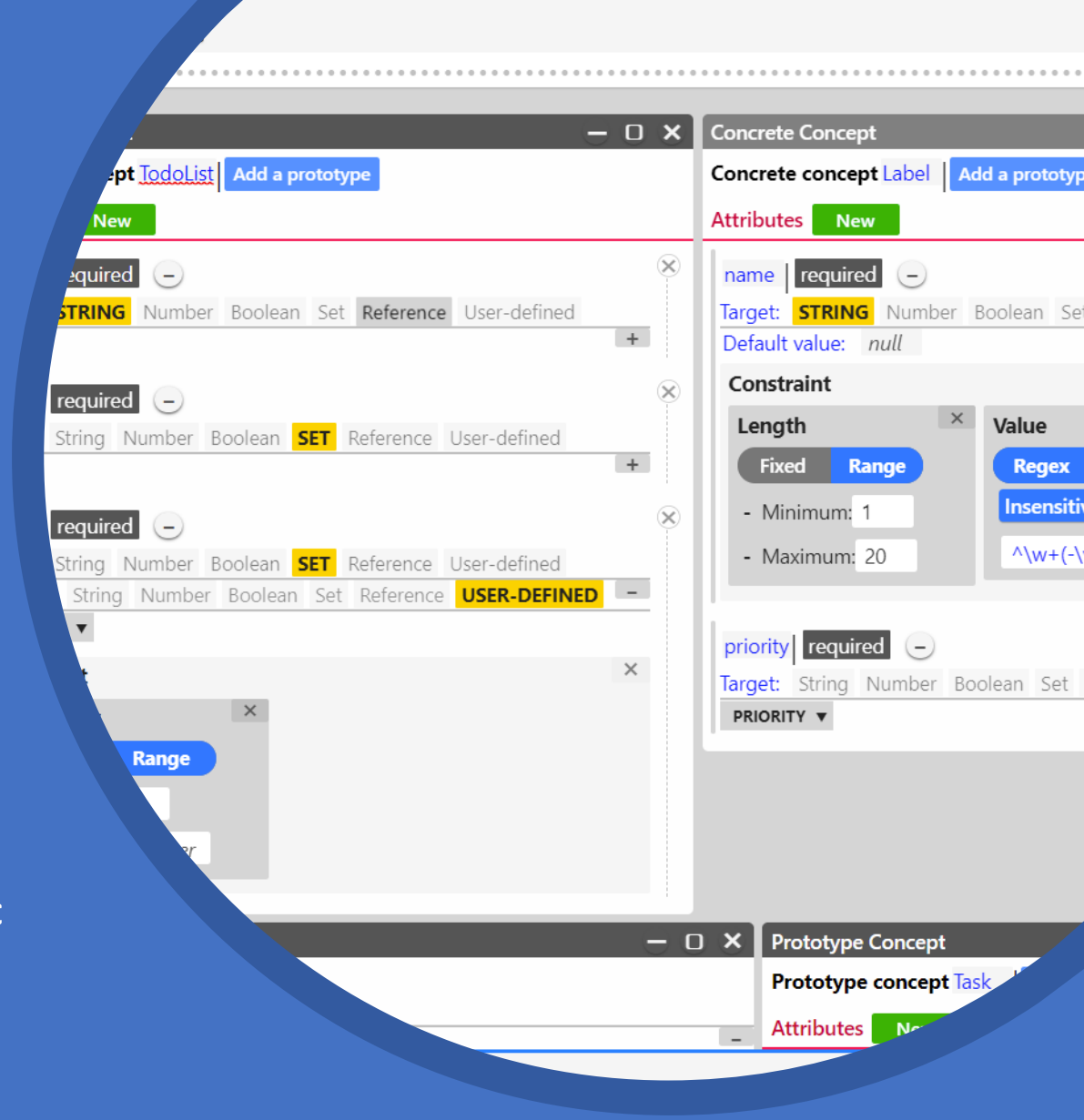
Encapsulates the concepts used to define a model or metamodel

Types of concept

- Primitive: String, Number, Boolean, Set, Reference
- Complex: Concrete, Prototype, Derivative

Relations

- **Attributes (external)**: extrinsic concept characteristic
- **Properties (internal)**: intrinsic concept characteristic



Projection

Representation of a concept that can be visualized and interacted with in the GUI

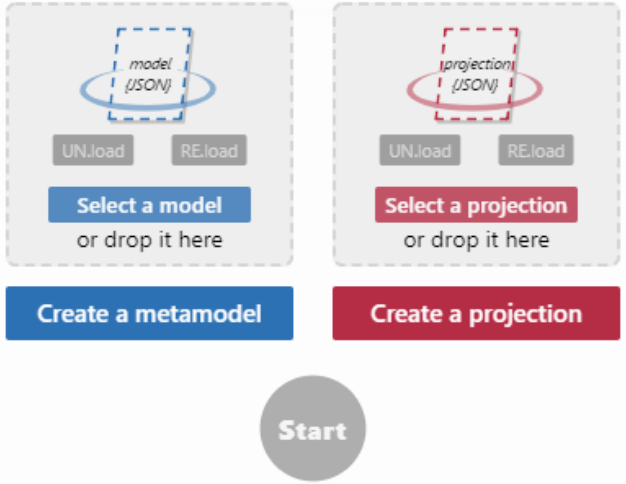
Common elements found in GUI design

- Layout: Flex, Table
- Field: Text, Binary, Choice, Link, List
- Static: Text, Link, Image, Button

⇒ **Help users quickly scan and comprehend the model**

Leverage web technology

- HTML 5 widgets, templates
- CSS style and animation



Demonstration

Concept definition

Projection definition

Integration

Outlook

Currently being looked at

- Add support for graphical notations
- Improve the projection preview experience
- Provide more export options

On the horizon

- Operations and predicate-based properties
- Integration with a modeling server
- Repository of concepts

Discussion and Questions

Thank you!