DSL Engineering with Xtext

Sven Efftinge - itemis.com

DOMAIN-SPECIFIC LANGUAGE

A Domain Specific Language (DSL) is a computer programming language focused on a particular domain.

Why DSLs?

- Software is eating the world!
- Abstractions!
- Better maintainability
- Work more closely with business people

What's needed for a DSL?

Lexer

Parser

Abstract Syntax Tree

Interpreter

Code Generator

Type Checker

Validation

Linker

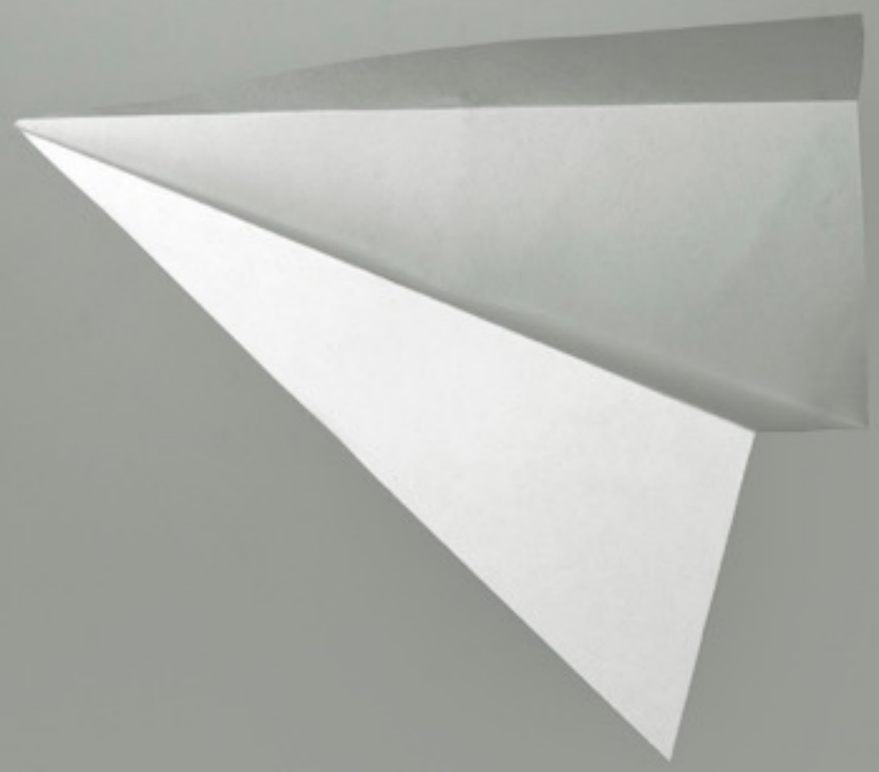
Type Inference Engine

IDE features

- ☑ syntax highlighting
- ☑ content assist
- ☑ validation
- ☑ mark occurrences
- ☑ match braces
- ☑ toggle comment
- ☑ smart editing
- ☑ formatting
- ☑ find references
- ☑ goto declaration
- ☑ goto implementation
- ☑ hover information
- ☑ rename refactoring
- ☑ quick fixes
- ☑ type hierarchy
- ☑ call hierarchy
- ☑ debugging
- ☑ outline
- ☑ folding
- ☑ smart word navigation
- ☑ expand selection
- ☑ dirty state management
- ☑ organize imports
- ☑ templates


eclipse


IntelliJIDEA



Xtext

Language Engineering Framework

Grammar-Driven (Single Sourcing)

Proven Compiler Architecture

Great Defaults

Flexible (Dependency Injection)

Supports Multiple Platforms

Continous Integration
(Maven & Gradle)

History...

May 2008

FIRST COMMIT

June, 2009

VERSION 0.7

March 2010

AWARD: MOST INNOVATIVE PROJECT

June, 2010

VERSION 1.0

June, 2011

VERSION 2.0

June, 2012

VERSION 2.3 (XBASE)

March 2013

VERSION 2.4

December 2013

VERSION 2.5

May, 2014

VERSION 2.6

May, 2014

XTEXTCON

Sep, 2014

VERSION 2.7

March, 2015

VERSION 2.8

May, 2015

XTEXTCON

Yesterday!

VERSION 2.9

TOTAL 5.747.523 LINES OF CODE

27.539 OCCURRENCES OF @TEST

MyHome.rules

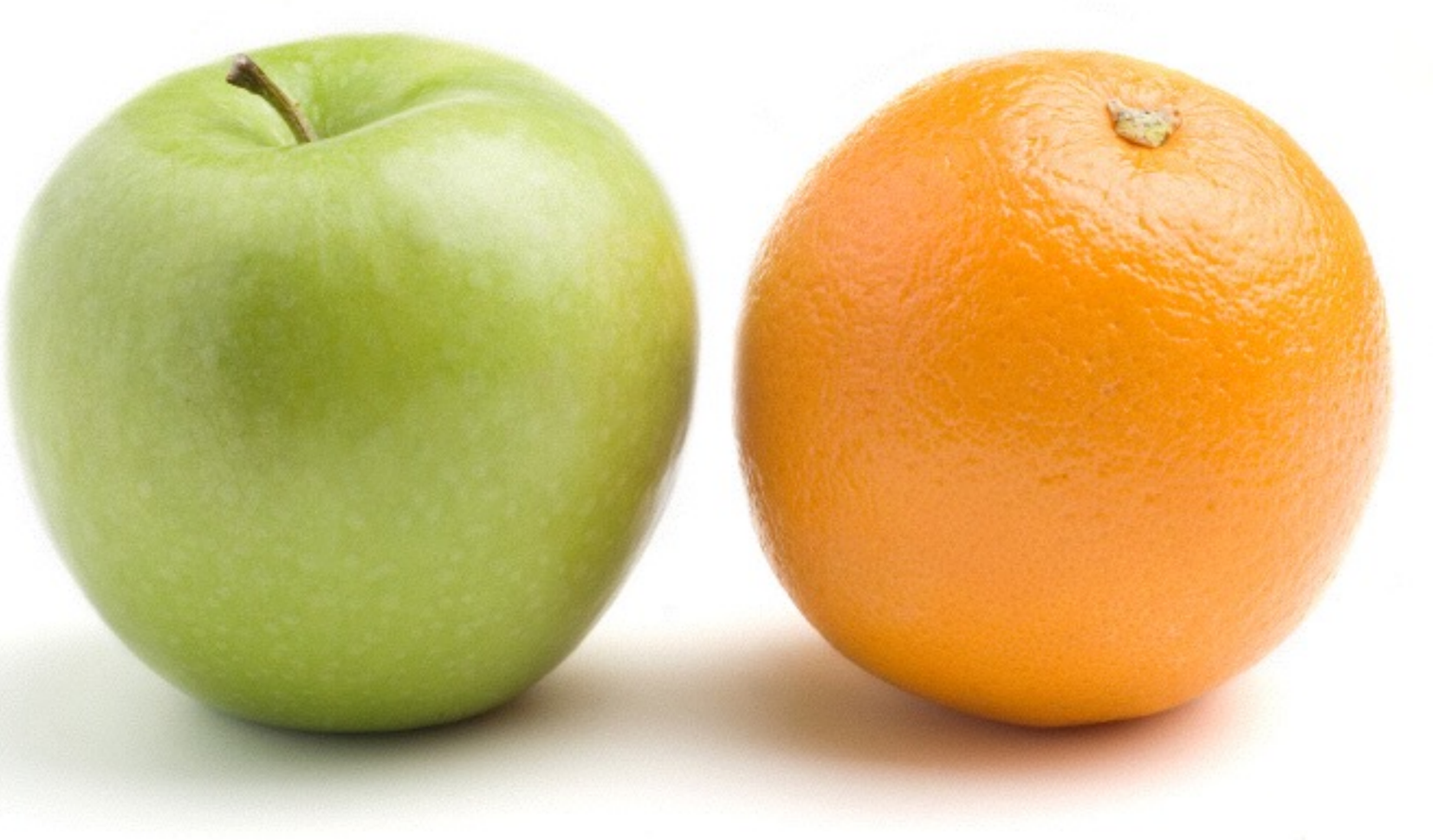
```
Device Window can be OPEN, CLOSED
Device Heating can be ON, OFF

Rule "Close window when heating is turned on"
    when Heating.ON
    then fire(Window.OPEN)

Rule "Turn heating off when window is opened"
    when Window.OPEN
    then fire(Heating.OFF)
```

Outline

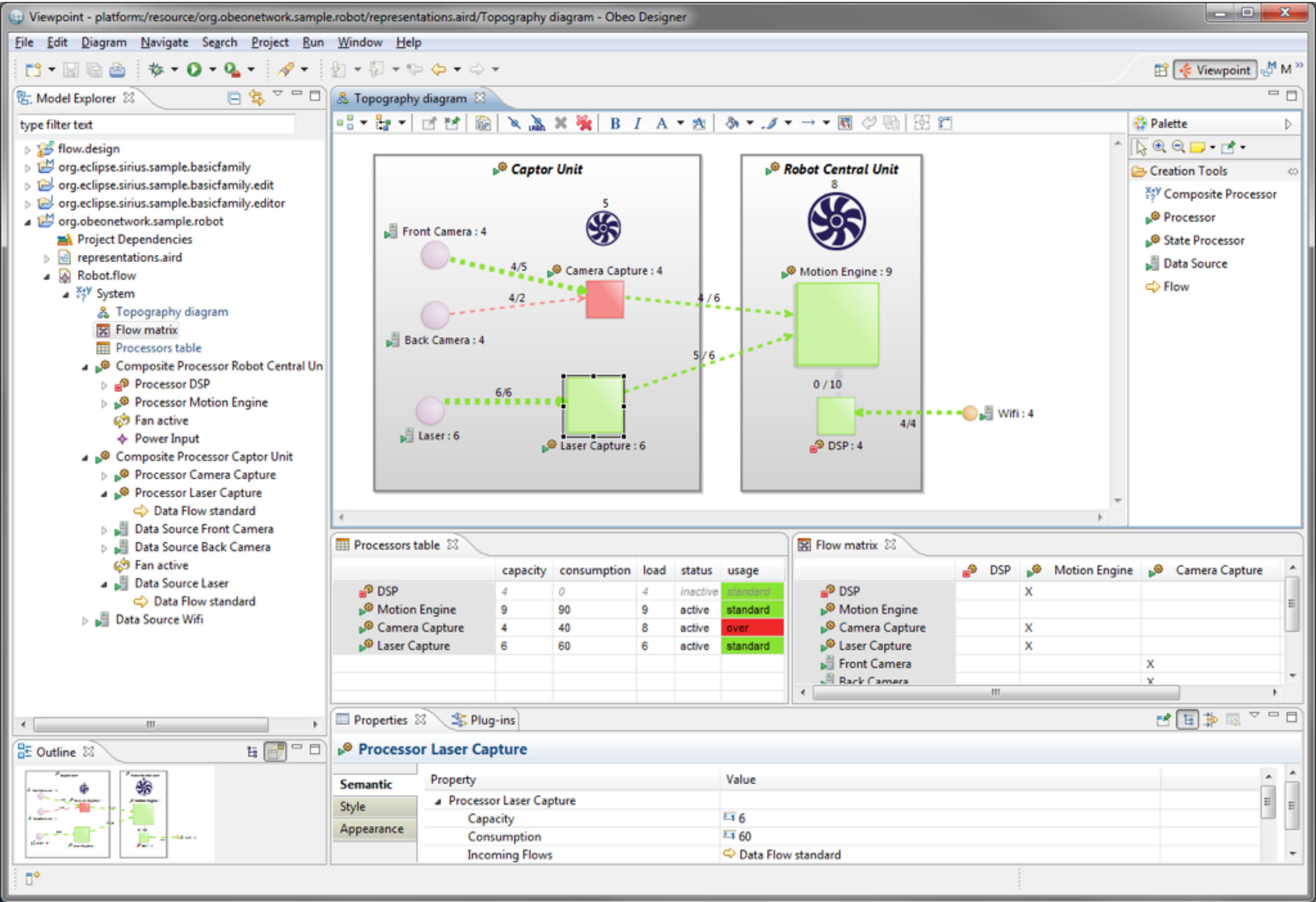
- MyHome
  - Window
  - Heating
  - Close window when h
  - Turn heating off whe

DEMO

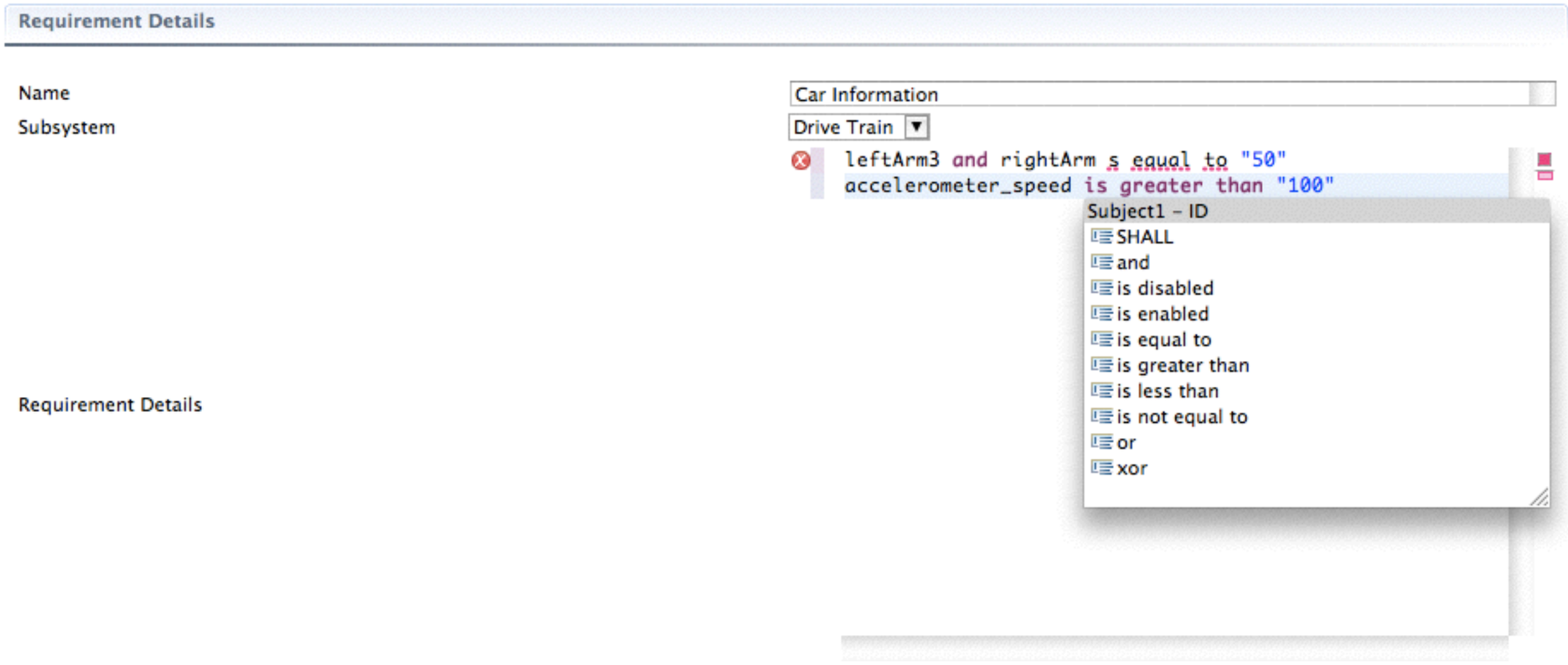
Beyond Simple Text



Sirius & Xtext



forms with embedded editors



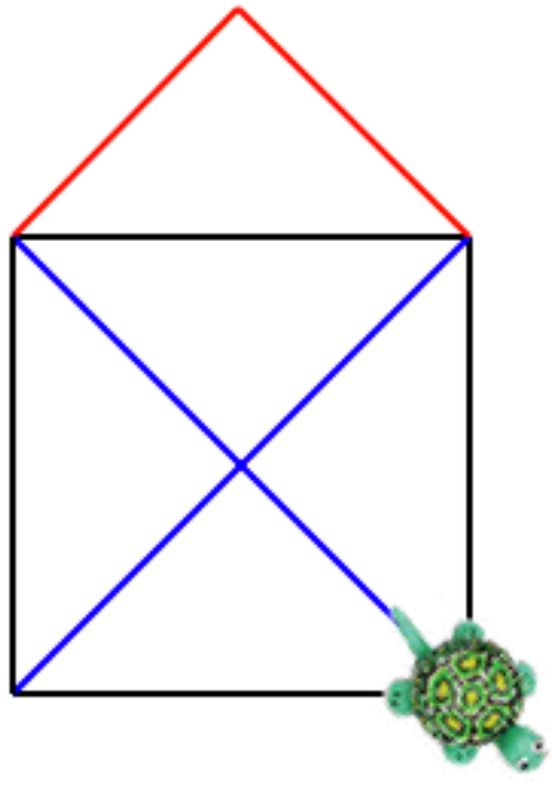
decision tables formatted

```
                 | 1           | 2 | 3           |
=================+=============+===+=============+
Address Present  |> hasAddress |   |             |
-----------------+-------------+---+-------------+
Subject Present  | hasSubject  |   | hasSubject  |
=================+=============+===+=============+
Send Mail        | x           |>  |             |
-----------------+-------------+---+-------------+
Error Message    |             |> x|             |
=================+=============+===+=============+
```

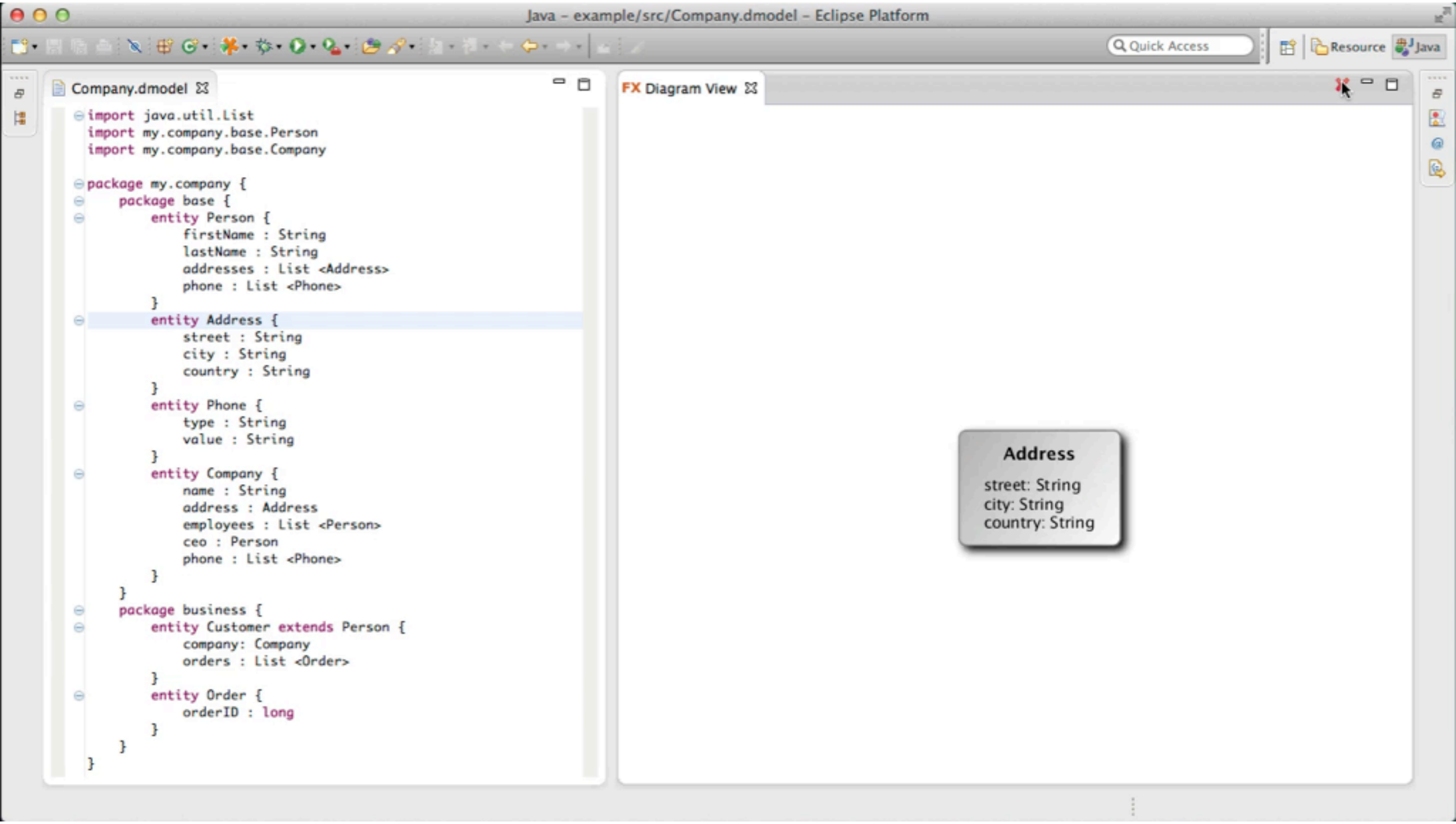
Nikolaus.tortoiseshell

```
begin
  val length = 150
  val diagonal = length * sqrt(2)
  lineWidth = 2
  square(length)
  turnRight(45)
  lineColor = blue
  forward(diagonal)
  turnLeft(90)
  lineColor = red
  forward(diagonal / 2)
  turnLeft(90)
  forward(diagonal / 2)
  turnLeft(90)
  lineColor = blue
  forward(diagonal)
end

sub square
  int length
begin
  for (i : 1..4) {
    forward(length)
    turnRight(90)
  }
```

Tortoise View



Graphical Views



Google

IBM

BOSCH

intuit

NOKIA
Connecting People

SAP

BMW

Aleri

Cisco Systems

GE

SAIC
From Science to Solutions

Community Experience Distilled

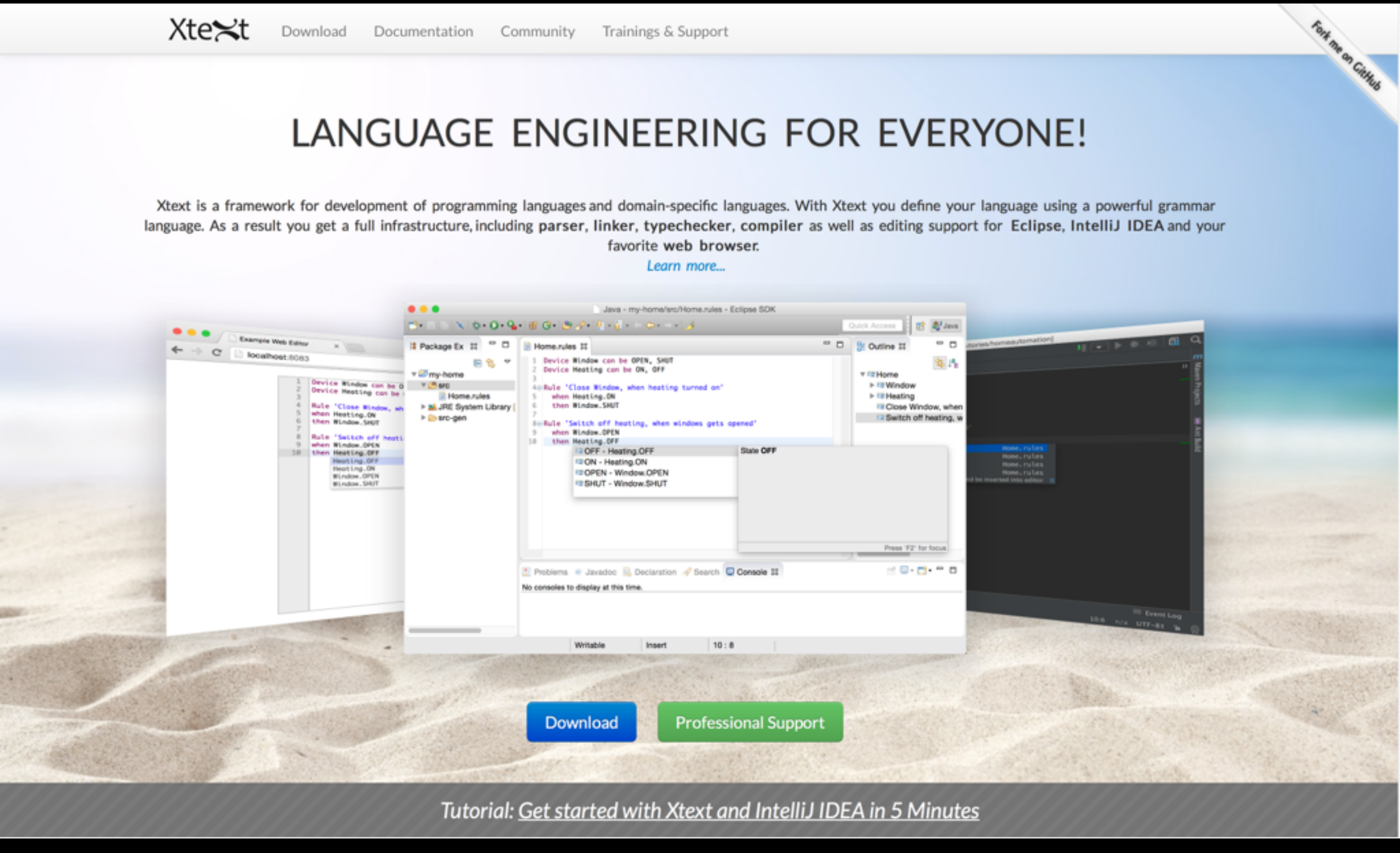
Implementing Domain-Specific Languages with Xtext and Xtend

Learn how to implement a DSL with Xtext and Xtend using easy-to-understand examples and best practices

Lorenzo Bettini

[PACKT] open source*

www.xtext.org



Tutorial: Get started with Xtext and IntelliJ IDEA in 5 Minutes