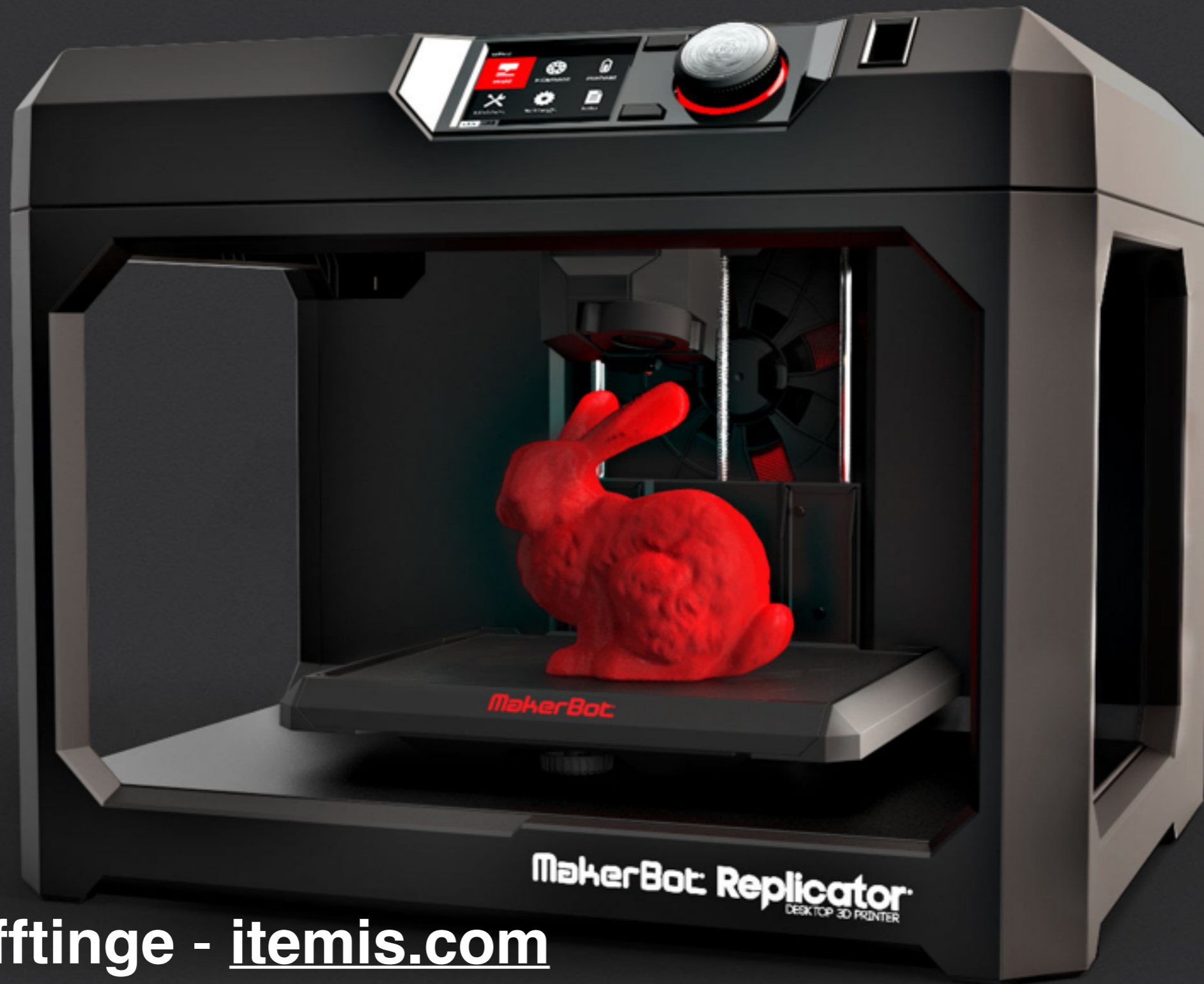


DSL Engineering with Xtext



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

Code Generator

Interpreter

Validation

Linker

Type Checker

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

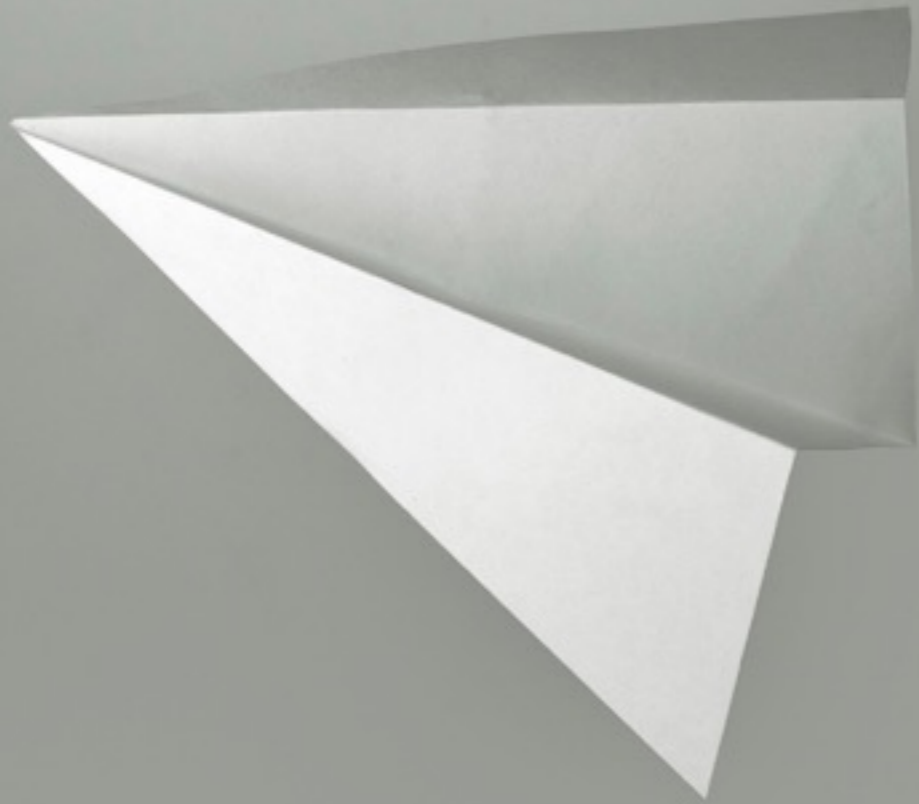


eclipse



IntellijIDEA





Xtext

Language Engineering Framework

Grammar-Driven (Single Sourcing)

Proven Compiler Architecture

Great Defaults

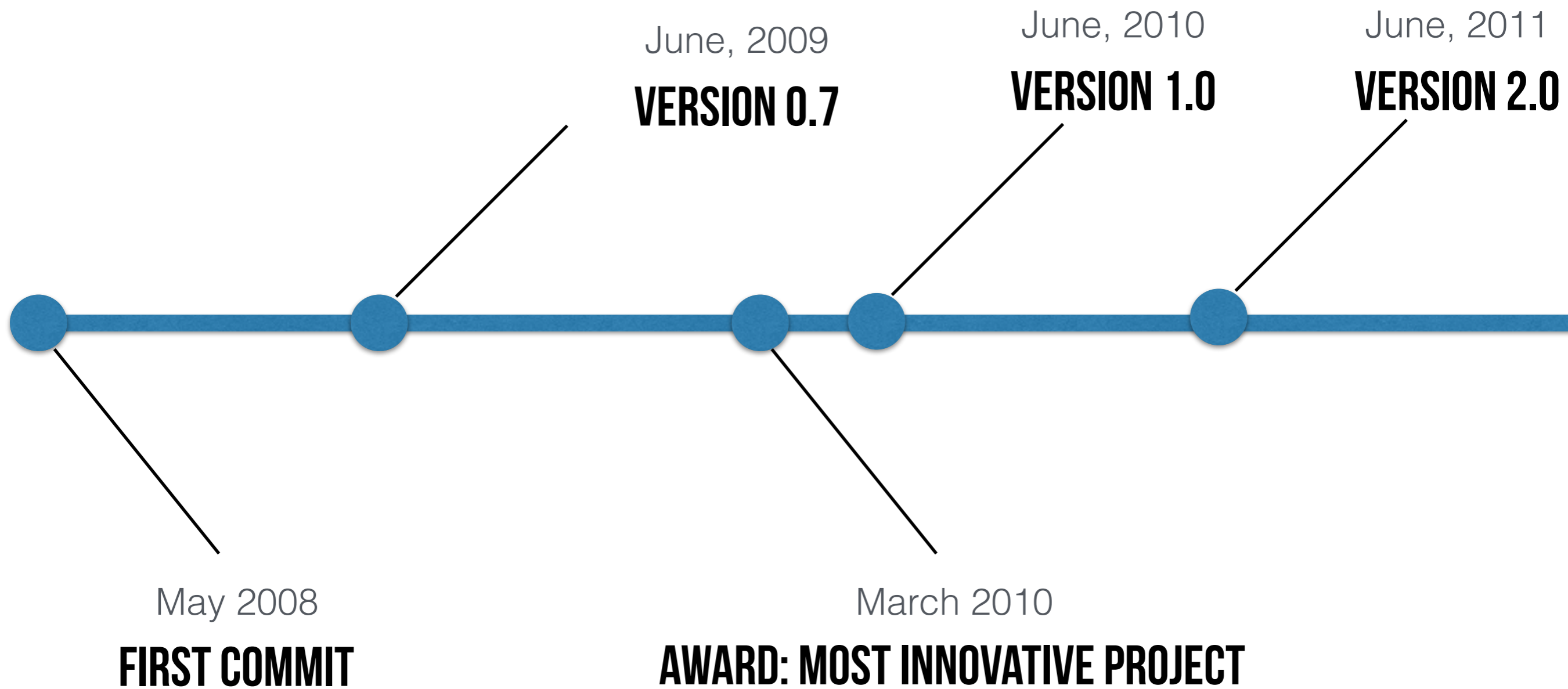
Flexible (Dependency Injection)

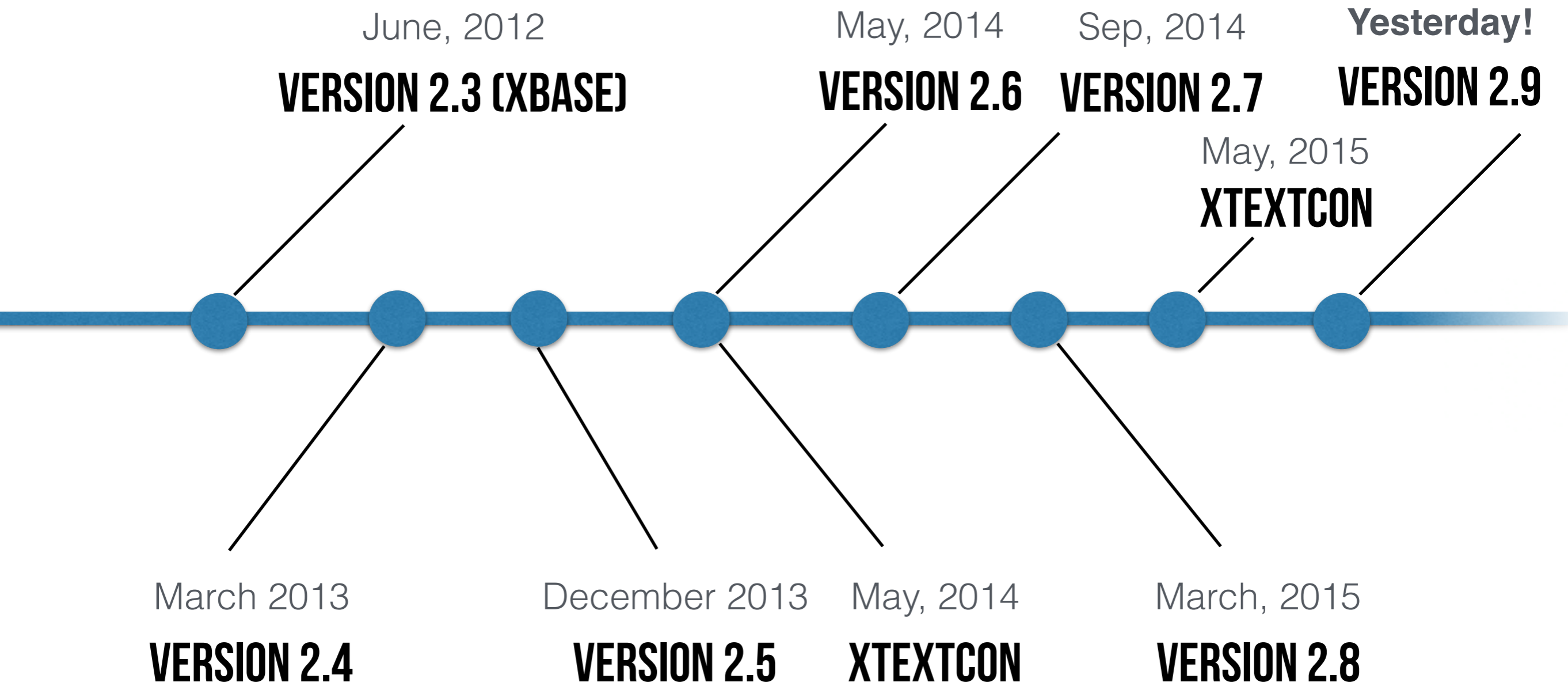
Supports Multiple Platforms

Continuous Integration
(Maven & Gradle)



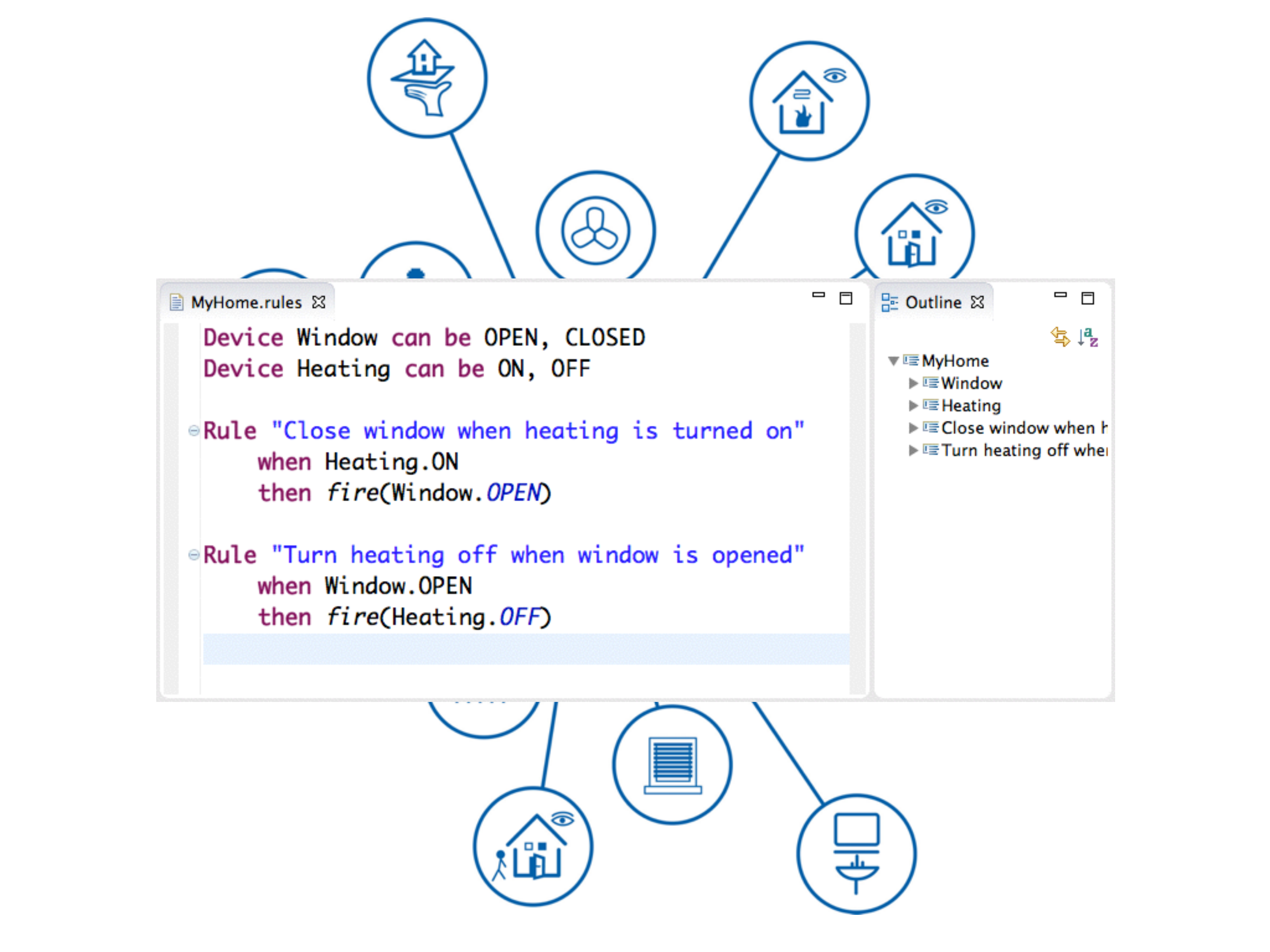
History...





A space-themed background featuring a view of Earth from space on the left, showing city lights and a blue atmosphere. On the right, the Moon is visible, partially illuminated. The background is filled with a starry field.

TOTAL 5.747.523 LINES OF CODE
27.539 OCCURRENCES OF @TEST

A background diagram consisting of several circular icons connected by lines. The icons include: a hand holding a house, a house with a flame, a house with an eye, a house with a person, a window blind, and a sink with a faucet.

MyHome.rules

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

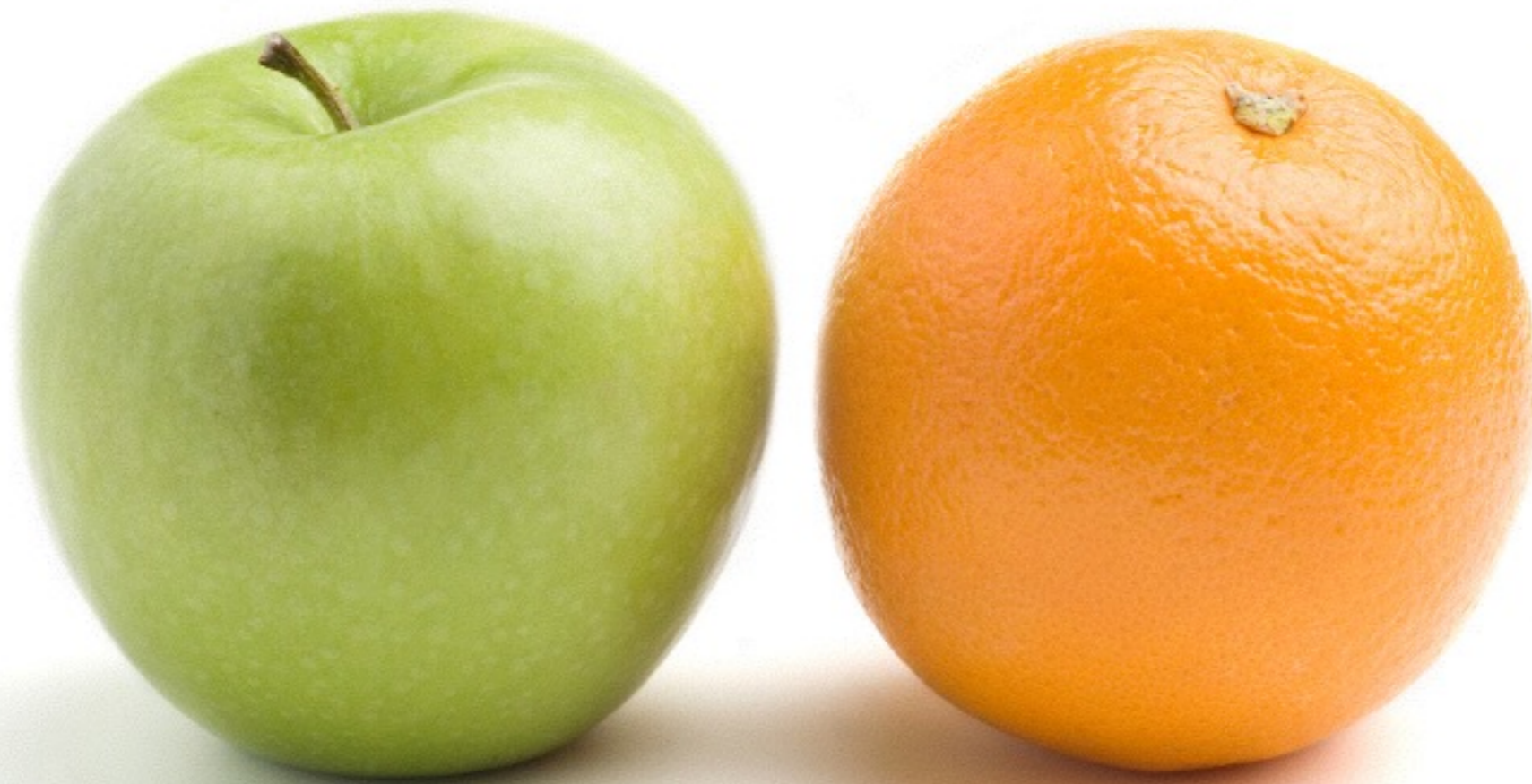
Rule "Close window when heating is turned on"
when Heating.ON
then fire(Window.OPEN)

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 when

Beyond Simple Text



Sirius & Xtext

Viewpoint - platform:/resource/org.obeonetwork.sample.robot/representations.aird/Topography diagram - Obeo Designer

File Edit Diagram Navigate Search Project Run Window Help

Model Explorer

type filter text

- flow.design
- org.eclipse.sirius.sample.basicfamily
- org.eclipse.sirius.sample.basicfamily.edit
- org.eclipse.sirius.sample.basicfamily.editor
- org.obeonetwork.sample.robot
 - Project Dependencies
 - representations.aird
 - Robot.flow
 - System
 - Topography diagram
 - Flow matrix
 - Processors table
 - Composite Processor Robot Central Unit
 - Processor DSP
 - Processor Motion Engine
 - Fan active
 - Power Input
 - Composite Processor Captor Unit
 - Processor Camera Capture
 - Processor Laser Capture
 - Data Flow standard
 - Data Source Front Camera
 - Data Source Back Camera
 - Fan active
 - Data Source Laser
 - Data Flow standard
 - Data Source Wifi

Topography diagram

Processors table

	capacity	consumption	load	status	usage
DSP	4	0	4	inactive	standard
Motion Engine	9	90	9	active	standard
Camera Capture	4	40	8	active	over
Laser Capture	6	60	6	active	standard

Flow matrix

	DSP	Motion Engine	Camera Capture
DSP		X	
Motion Engine			
Camera Capture		X	
Laser Capture		X	
Front Camera			X
Back Camera			X

Properties Plug-ins

Processor Laser Capture

Semantic	Property	Value
Style Appearance	Processor Laser Capture	
	Capacity	6
	Consumption	60
	Incoming Flows	Data Flow standard

Outline

forms with embedded editors

Requirement Details

Name

Subsystem

Car Information

Drive Train ▾

✘ leftArm3 and rightArm s equal to "50"
accelerometer_speed is greater than "100"

Subject1 - ID

- SHALL
- and
- is disabled
- is enabled
- is equal to
- is greater than
- is less than
- is not equal to
- or
- xor

Requirement Details

decision tables formatted

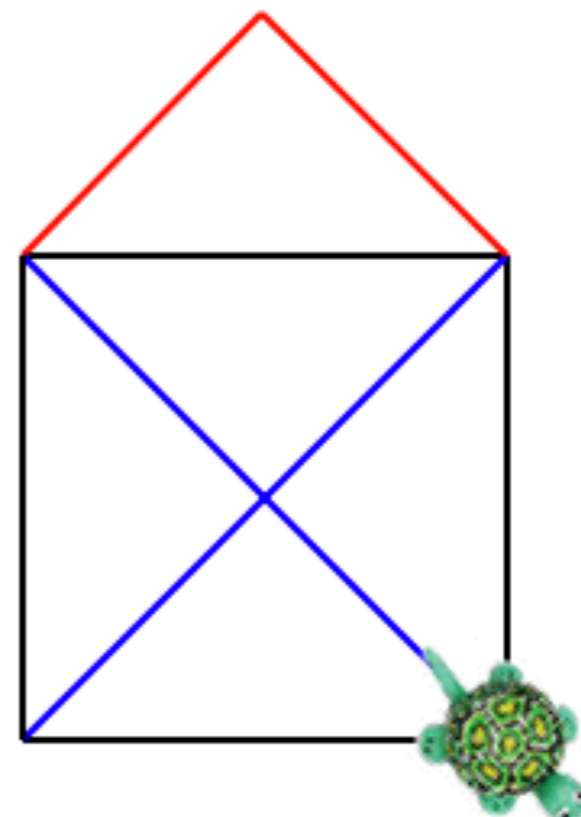
	1	2	3	
Address Present	>	hasAddress		
Subject Present		hasSubject		hasSubject
Send Mail	x	>		
Error Message		>	x	

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)
  }
end
```

Tortoise View



Graphical Views

The screenshot displays the Eclipse IDE interface for a Domain Model (Company.dmodel). The left pane shows the source code, and the right pane shows the graphical representation of the selected entity, Address.

```
import java.util.List
import my.company.base.Person
import my.company.base.Company

package my.company {
  package base {
    entity Person {
      firstName : String
      lastName : String
      addresses : List <Address>
      phone : List <Phone>
    }
    entity Address {
      street : String
      city : String
      country : String
    }
    entity Phone {
      type : String
      value : String
    }
    entity Company {
      name : String
      address : Address
      employees : List <Person>
      ceo : Person
      phone : List <Phone>
    }
  }
  package business {
    entity Customer extends Person {
      company : Company
      orders : List <Order>
    }
    entity Order {
      orderID : long
    }
  }
}
```

The graphical view (FX Diagram View) shows a box representing the **Address** entity with the following attributes:

- street: String
- city: String
- country: String

Google

IBM

NOKIA
Connecting People



BOSCH

intuit.

Users

SAP

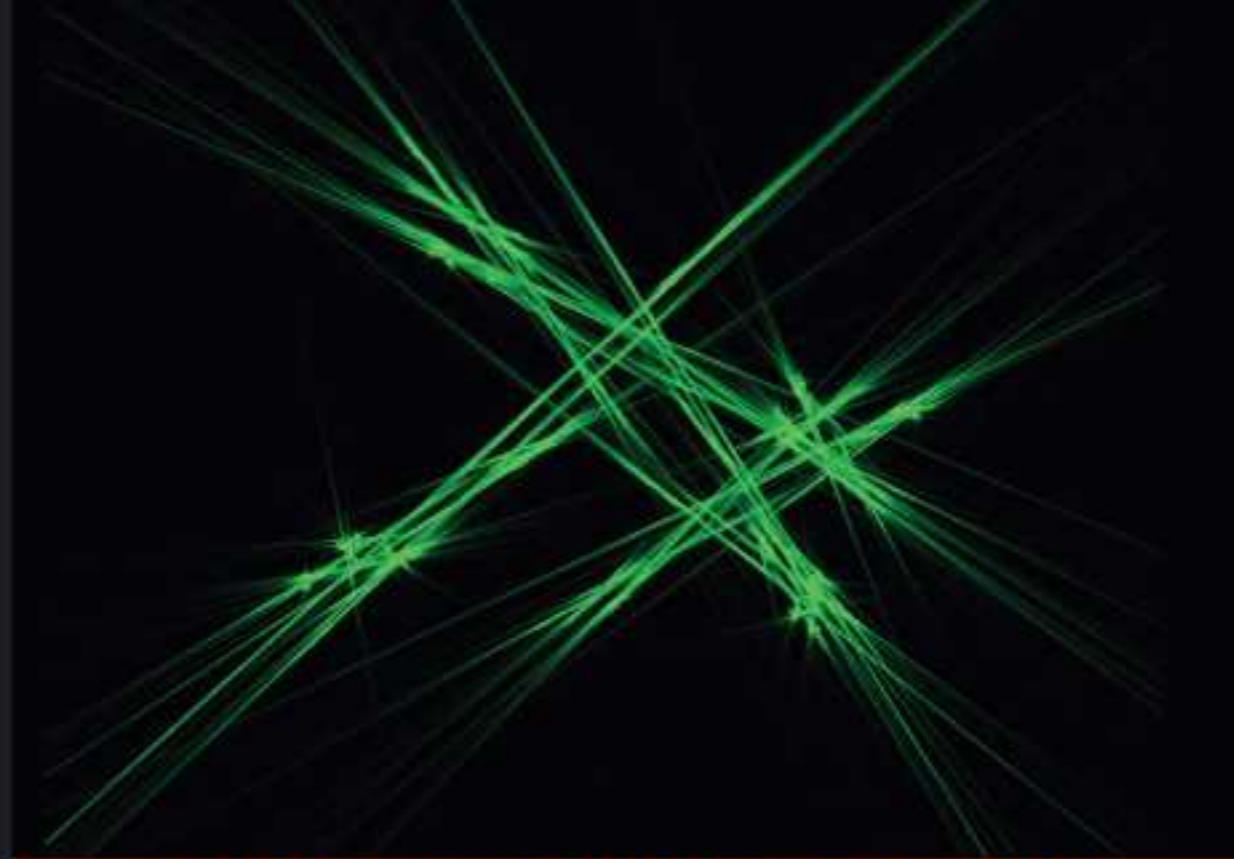


CISCO SYSTEMS



Aleri

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*
PUBLISHING

www.xtext.org

Xtext

Download

Documentation

Community

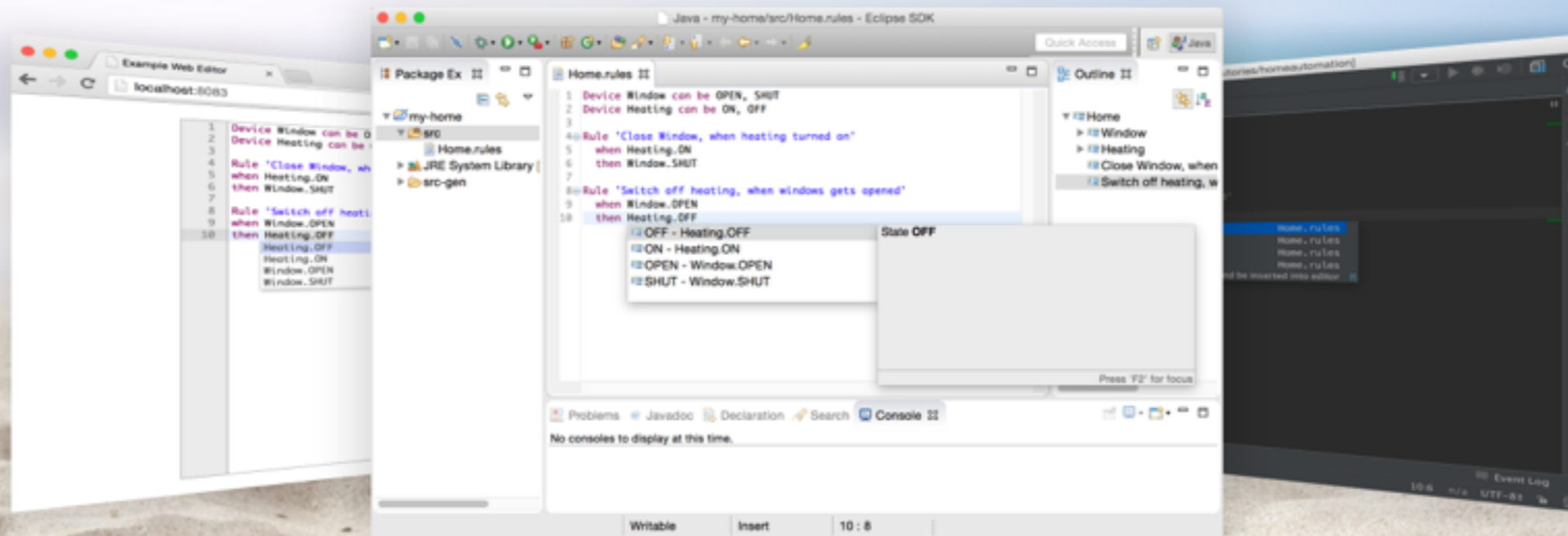
Trainings & Support

Fork me on GitHub

LANGUAGE ENGINEERING FOR EVERYONE!

Xtext is a framework for development of programming languages and domain-specific languages. With Xtext you define your language using a powerful grammar language. As a result you get a full infrastructure, including parser, linker, typechecker, compiler as well as editing support for Eclipse, IntelliJ IDEA and your favorite web browser.

[Learn more...](#)



Download

Professional Support

Tutorial: [Get started with Xtext and IntelliJ IDEA in 5 Minutes](#)