

Procesadores de Lenguajes 2

# Lenguajes Específicos de Dominio

Curso 2013-2014

Iván Ruiz Rube

Departamento de Ingeniería Informática

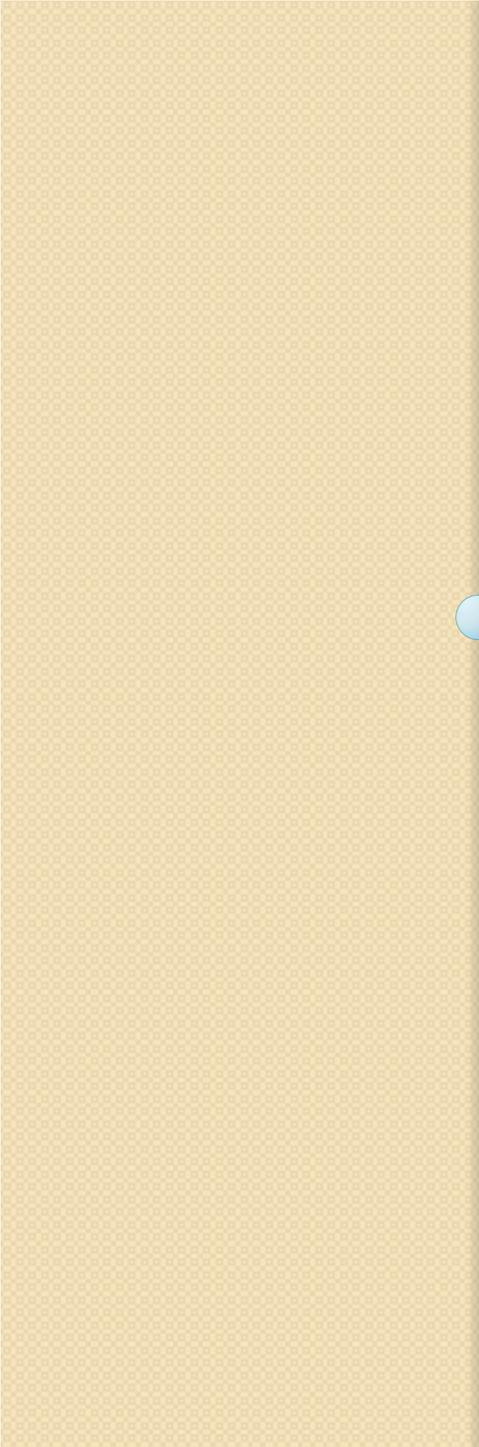
Escuela Superior de Ingeniería

Universidad de Cádiz



# Contenidos

- Introducción
- Clasificación
- Herramientas



LENGUAJES ESPECÍFICOS DE DOMINIO



# INTRODUCCIÓN



# Domain Specific Languages (DSL)

- Lenguaje de programación orientado a un problema específico.
- Son opuestos a los lenguajes de propósito general.
- Diseñado para ser utilizado para un conjunto específico de tareas.
- Herramienta específica para un trabajo concreto.

# Herramienta de Propósito General



**Peter Friese, itemis**

# Herramienta de Propósito Específico

A specific tool  
for a specific job



**Peter Friese, itemis**



# Domain Specific Languages (DSL)

- La semántica del lenguaje está muy cercana al dominio de problema para el cual se diseña.
- Los DSL ofrecen un alto nivel de abstracción al usuario. Por tanto, están dirigidos a “expertos en el dominio”.
- Ofrecen un vocabulario controlado para su ámbito.

# Ejemplos DSL

```
1<project xmlns="http://maven.apache.org/POM/4.0.0"
2  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4  http://maven.apache.org/maven-v4_0_0.xsd">
5
6  <modelVersion>4.0.0</modelVersion>
7  <groupId>org.sonatype.mavenbook.simple</groupId>
8  <artifactId>simple</artifactId>
9  <packaging>jar</packaging>
10 <version>1.0-SNAPSHOT</version>
11 <name>simple</name>
12 <url>http://maven.apache.org</url>
13
14 <dependencies>
15   <dependency>
16     <groupId>junit</groupId>
17     <artifactId>junit</artifactId>
18     <version>3.8.1</version>
19     <scope>test</scope>
20   </dependency>
21 </dependencies>
```

## Apache Maven

Automatización de la gestión y construcción de proyectos Java

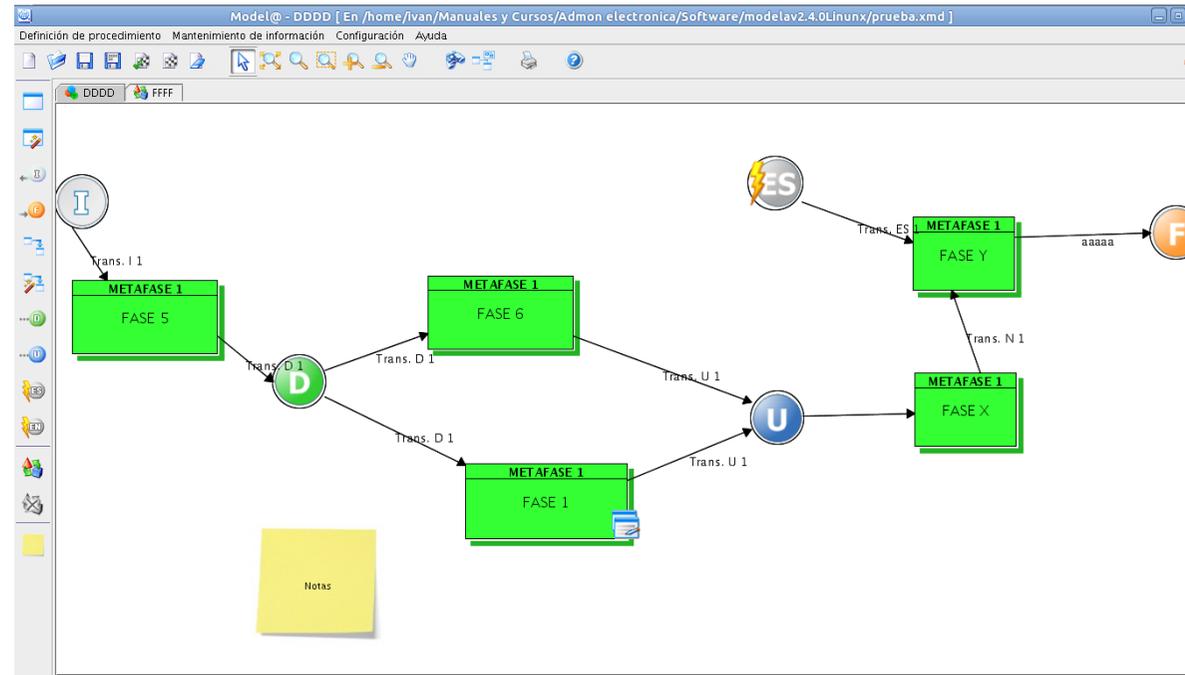
# Ejemplos DSL (II)

```
rule "Driver in unsafe area for marginal age"  
  when  
    Policy type is 'COMPREHENSIVE'  
    Driver is less than 25 years old  
    Driver has a location risk profile of  
  
  then  
    <> Driver has a location risk profile of '{risk}'  
    <> Driver has an age of at least {age}  
  end  
rule "Dr  
  when  
    <> Driver has had more than {prior} prior claims  
    <> Driver has had {number} prior claims  
    <> Driver is between {lower} and {upper} years old  
    <> Driver is greater than {age} years old  
    <> Driver is less than {age} years old  
    <> Policy has not been rejected  
    <> Policy type is '{type}'
```

## **JBoss Drools**

Reglas de negocio definidas en alto nivel, dentro de un motor de procesos

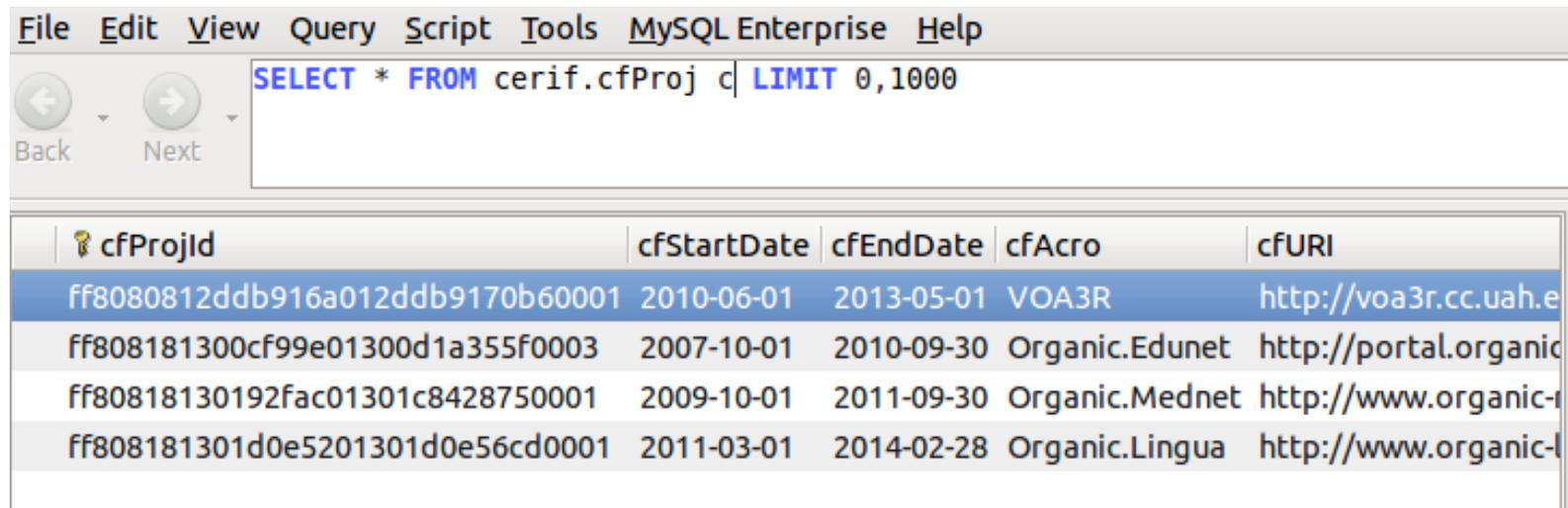
# Ejemplos DSL (III)



**TREW@**

Modelado de procedimientos administrativos telemáticos  
en el marco de la administración pública en Andalucía

# Ejemplos DSL (IV)



The screenshot shows the MySQL Enterprise interface. The menu bar includes File, Edit, View, Query, Script, Tools, MySQL Enterprise, and Help. The query editor contains the SQL statement: `SELECT * FROM certif.cfProj c LIMIT 0,1000`. Below the query editor, a table displays the results of the query. The table has five columns: cfProjId, cfStartDate, cfEndDate, cfAcro, and cfURI. The first row is highlighted in blue.

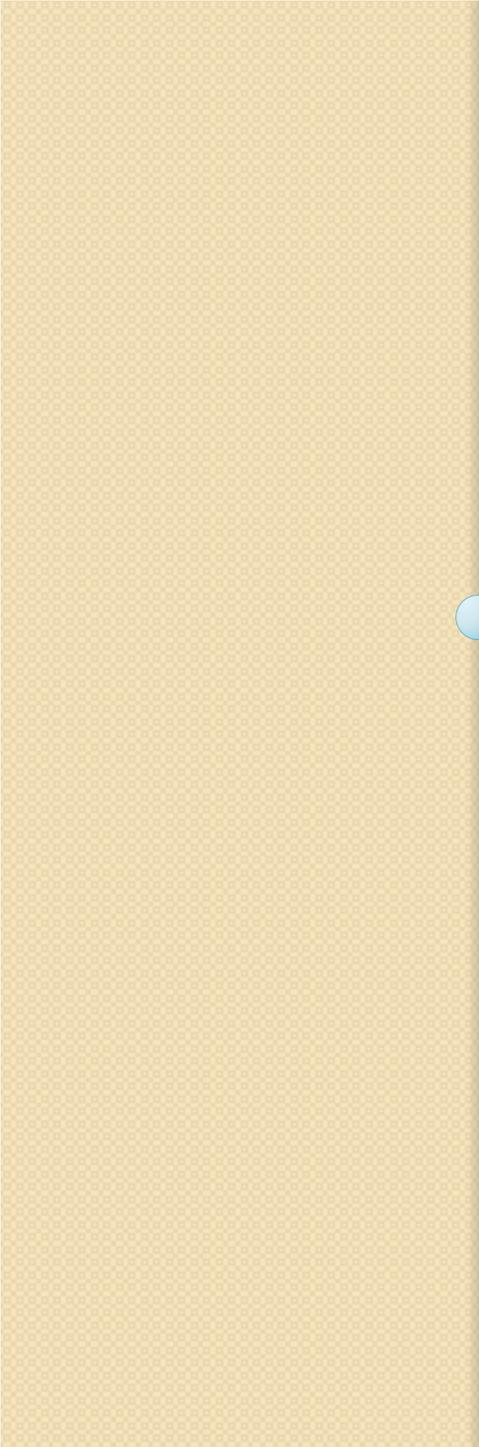
cfProjId	cfStartDate	cfEndDate	cfAcro	cfURI
ff8080812ddb916a012ddb9170b60001	2010-06-01	2013-05-01	VOA3R	http://voa3r.cc.uah.e
ff808181300cf99e01300d1a355f0003	2007-10-01	2010-09-30	Organic.Edunet	http://portal.organic
ff80818130192fac01301c8428750001	2009-10-01	2011-09-30	Organic.Mednet	http://www.organic-r
ff808181301d0e5201301d0e56cd0001	2011-03-01	2014-02-28	Organic.Lingua	http://www.organic-l

## Structured Query Language (SQL)

Manipulación de bases de datos relacionales

# Ejemplos DSL (V)

- HTML
  - Utilizado para escribir páginas web
- CSS
  - Usado para describir hojas de estilos en la web
- Flex, Bison o ANTLR:
  - Para generar compiladores
- AWK, plantillas Wikipedia, shell scripts, etc.



LENGUAJES ESPECÍFICOS DE DOMINIO

 **CLASIFICACIÓN**



# Clasificación de DSL

- Representación
  - Visuales
  - Textuales
- Implementación
  - Internos
  - Externos

# DSL textuales internos

- Utilizan la infraestructura de un lenguaje de programación existente (host).
- En la mayoría de las ocasiones, se implementan como librerías en el lenguaje de programación.
- Ejemplo: Rails
  - Construido sobre la base de Ruby
  - Orientado a construir aplicaciones web

# DSL textuales externos

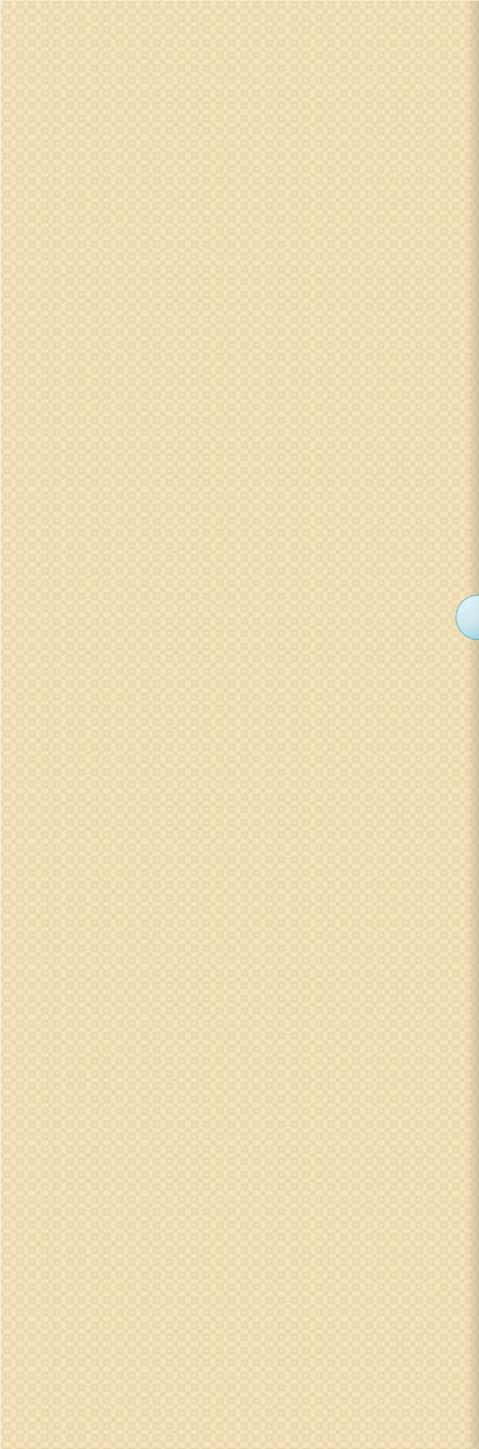
- No se hospedan bajo ningún lenguaje.
- Desarrollados desde cero.
- Requiere la construcción de las etapas clásicas de un compilador.
- Sin embargo, no necesitan todas las complejidades que tienen los lenguajes de propósito general.

# DSL visuales internos

- Muchos problemas de Ingeniería del Software se analizan e interpretan mejor utilizando modelos gráficos.
- UML proporciona un mecanismo de extensión (*profiles*) para personalizar modelos para dominios y plataformas particulares.
- Los perfiles no modifican la semántica de UML, sino que la amplían.
- Se implementan mediante estereotipos, definiciones de etiquetas y restricciones.

# DSL visuales externos

- Los lenguajes visuales son más fáciles de analizar que los textuales.
- Sin embargo, el lenguaje UML y los perfiles basados en él, no son del todo sencillos para un usuario no técnico.
- Por ello, se desarrollan lenguajes gráficos con las semánticas apropiadas para la clase de problemas a modelar.
- El desarrollo de este tipo de lenguajes suele ser más complejo que uno textual.



LENGUAJES ESPECÍFICOS DE DOMINIO

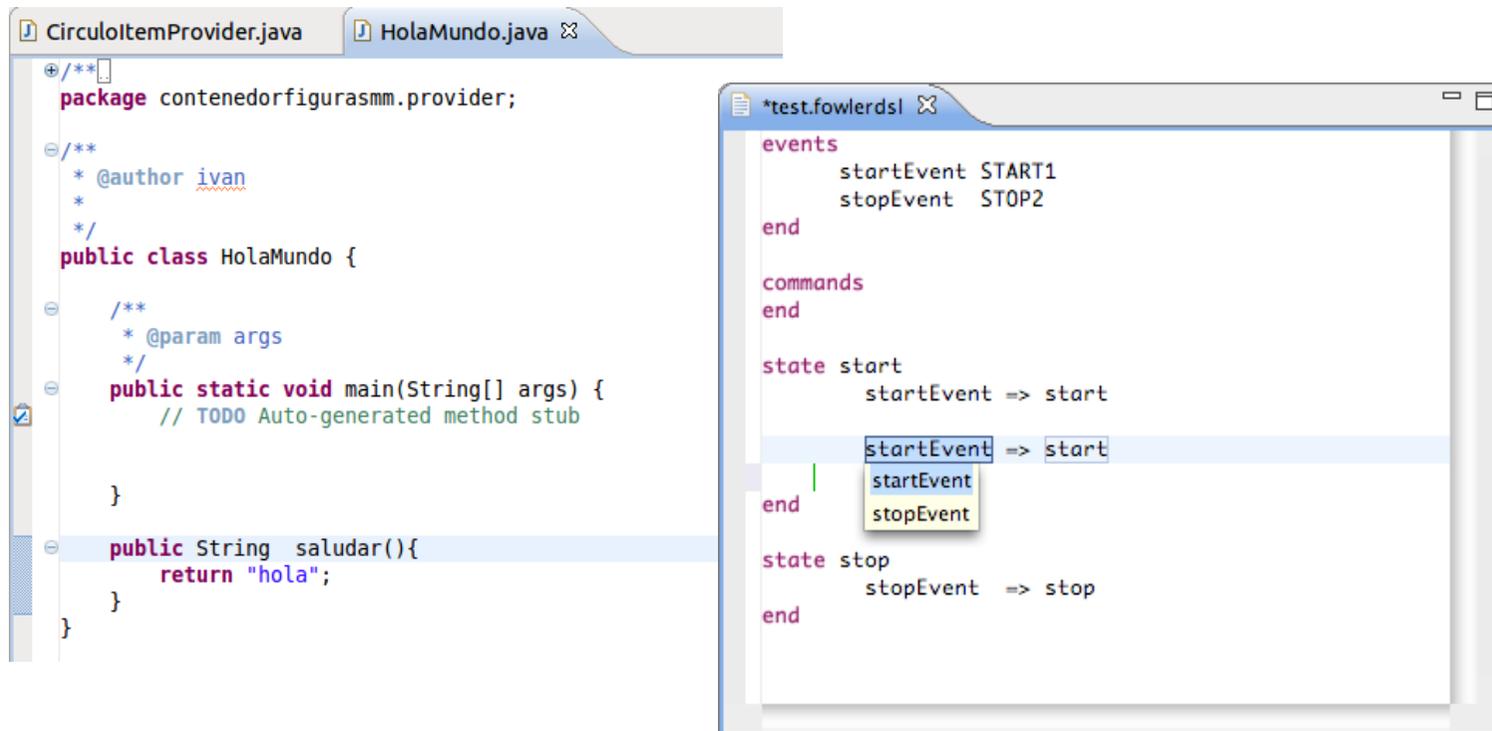


# **HERRAMIENTAS**

# Herramientas de desarrollo DSL

- Para dar soporte a las fases del desarrollo de un DSL, necesitamos herramientas que nos asistan en dicho proceso.
- Enfoque tradicional
  - FLEX, BISON
  - ANTLR,
  - etc.
- Enfoque *model-driven*
  - **Eclipse (Modeling Project)**
  - MS Visual Studio (DSL Tools)
  - MetaEdit+
  - etc.

# Eclipse: Edición textual



The screenshot displays the Eclipse IDE interface. On the left, the 'CirculoItemProvider.java' editor shows the following Java code:

```
/**
 * @author ivan
 */
package contenedorfigurasm. provider;

public class HolaMundo {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }

    public String saludar(){
        return "hola";
    }
}
```

On the right, the '\*test.fowlerdsl' editor shows a DSL (Domain Specific Language) code snippet:

```
events
  startEvent START1
  stopEvent STOP2
end

commands
end

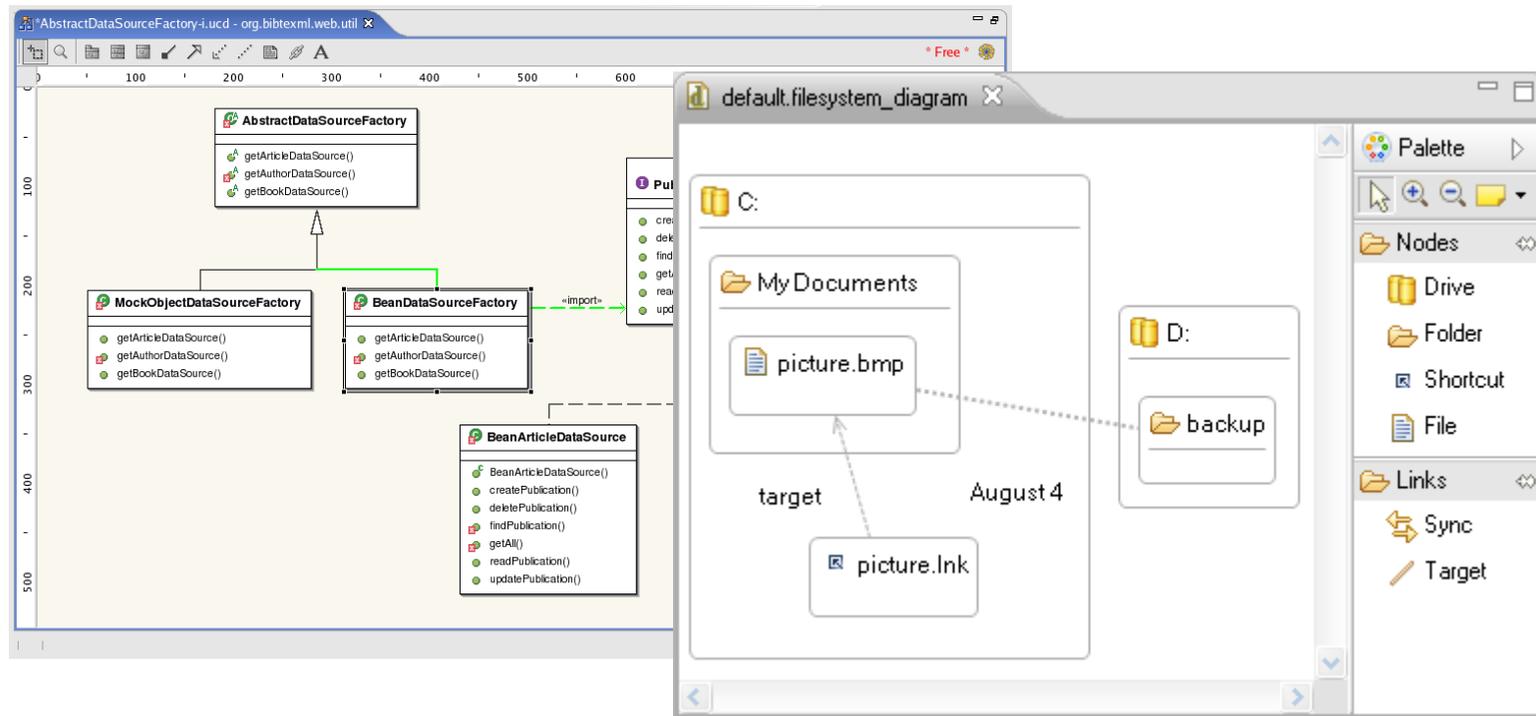
state start
  startEvent => start
end

state stop
  stopEvent => stop
end
```

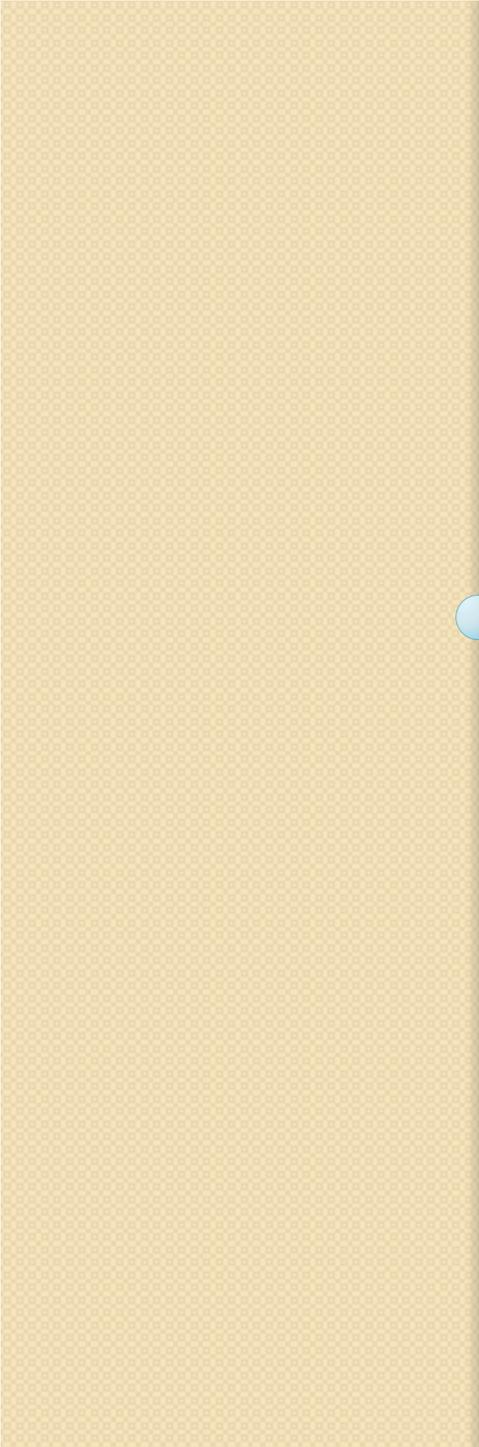
The DSL code is being edited, with a dropdown menu visible over the 'startEvent => start' line, showing options: 'startEvent', 'stopEvent', and 'startEvent'.

Eclipse es un entorno de programación para lenguaje Java, aunque dispone de extensiones para trabajar con otros lenguajes como PHP, XML, etc. Así mismo, permite trabajar con DSLs creados por el usuario

# Eclipse: Edición visual



Eclipse permite trabajar con modelos visuales basados en UML o en DSLs creados por el usuario



LENGUAJES ESPECÍFICOS DE DOMINIO



# RESUMEN

# Resumen

- Lenguaje de programación orientado a un problema específico.
- Pueden ser visuales o textuales.
- Se pueden implementar interna o externamente.
- En esta asignatura, construiremos DSLs siguiendo un enfoque *model-driven* utilizando las facilidades que ofrece el proyecto Eclipse.



Procesadores de Lenguajes 2

# Lenguajes Específicos de Dominio

Curso 2013-2014

**Iván Ruiz Rube**

ivan.ruiz@uca.es