

EuGENia

José Joaquín González Maline

Universidad de Cádiz

15 de diciembre de 2011

1 Instalación

2 EuGENia

Epsilon
Emfatic
EuGENia

3 GMF

4 Ejemplo

Proyecto GMF
Fichero Emfatic

5 Conclusiones

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

GMF

Ejemplo

Proyecto GMF

Fichero Emfatic

Conclusiones

1 Instalación

2 EuGENia

Epsilon
Emfatic
EuGENia

3 GMF

4 Ejemplo

Proyecto GMF
Fichero Emfatic

5 Conclusiones

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

GMF

Ejemplo

Proyecto GMF

Fichero Emfatic

Conclusiones

Para hacer uso de EuGENia, debemos descargarnos la distribución Epsilon de Eclipse.

Podemos encontrarla en la siguiente dirección:
<http://eclipse.org/gmt/epsilon/download/>



1 Instalación

2 EuGENia

Epsilon
Emfatic
EuGENia

3 GMF

4 Ejemplo

Proyecto GMF
Fichero Emfatic

5 Conclusiones

- Epsilon es un conjunto de lenguajes de programación de tareas específicas consistente e interoperable.



- Es usado para interactuar con los modelos EMF para realizar tareas comunes de Model Driven Engineering como generación de código, transformaciones M2M, validación de modelos, comparación, migración, fusión y refactorización.

- Emfatic es un editor de texto que soporta navegación, edición y conversión de modelos Ecore, usando un sintaxis similar a la de Java.
- Está diseñado para representar modelos EMF Ecore de forma textual.
- Permite la opción de convertir un diagrama Ecore en código Emfatic automáticamente, así como lo inverso.

Documentación:

<http://eclipse.org/gmt/epsilon/doc/articles/emfatic/>

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

GMF

Ejemplo

Proyecto GMF

Fichero Emfatic

Conclusiones

EuGENia es una herramienta que simplifica el desarrollo de los editores GMF.

Permite la creación automática de los modelos *gmfgraph*, *gmftool* y *gmfmap* a partir de un metamodelo creado con Ecore tal y como hemos visto en clase.

No permite todas las opciones de *gmfgraph* y *gmfmap*. Solo un subconjunto con las características más usadas.

En la documentación de EuGENia podemos encontrar un primer ejemplo funcional que nos permitirá crear un editor visual donde podremos emular un sistema de ficheros. El código es el siguiente:

```
@namespace(uri="filesystem", prefix="filesystem")
@gmf(foo="bar")
package filesystem;

@gmf.diagram(foo="bar")
class Filesystem {
    val Drive[*] drives;
    val Sync[*] syncs;
}

class Drive extends Folder {

}

class Folder extends File {
    @gmf.compartment(foo="bar")
    val File[*] contents;
}

class Shortcut extends File {
    @gmf.link(target.decoration="arrow", style="dash")
    ref File target;
}
```

Continuación.

```
@gmf.link(source="source", target="target", style="dot", width="2")
class Sync {
    ref File source;
    ref File target;
}

@gmf.node(label = "name")
class File {
    attr String name;
}
```

Fuente: <http://eclipse.org/gmt/epsilon/doc/eugenia/>

Resultado hipotético.

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

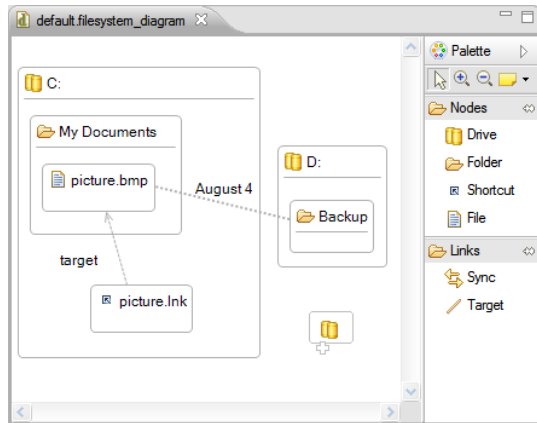
GMF

Ejemplo

Proyecto GMF

Fichero Emfatic

Conclusiones



Resultado real.

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

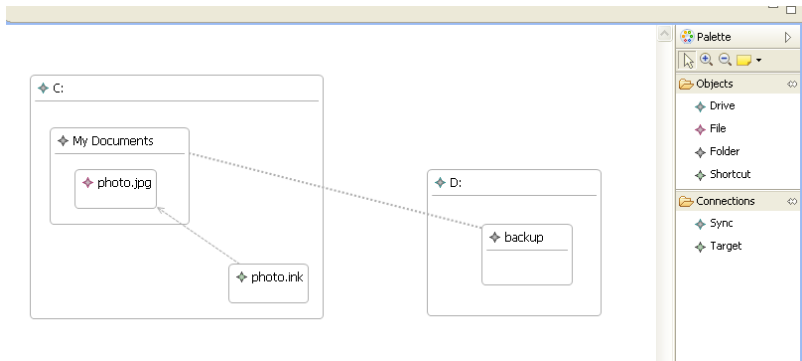
GMF

Ejemplo

Proyecto GMF

Fichero Emfatic

Conclusiones



1 Instalación

2 EuGENia

Epsilon
Emfatic
EuGENia

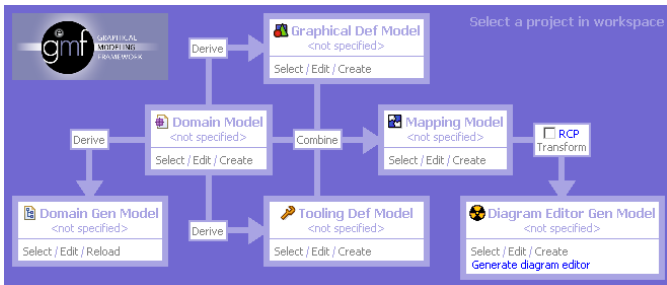
3 GMF

4 Ejemplo

Proyecto GMF
Fichero Emfatic

5 Conclusiones

- En clase desarrollamos un editor gráfico a través del proyecto GMF.
- Seguimos todas las transformaciones, y el proceso fue un poco tedioso.
- Hicimos uso del dashboard.



1 Instalación

2 EuGENia

Epsilon
Emfatic
EuGENia

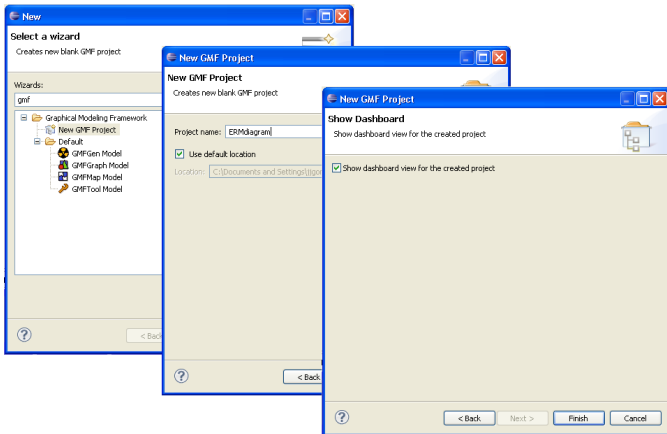
3 GMF

4 Ejemplo

Proyecto GMF
Fichero Emfatic

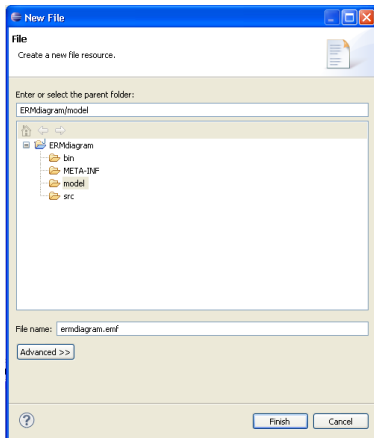
5 Conclusiones

Lo primero que necesitamos hacer es crear un proyecto GMF



File > New > Other... GMF Project

A continuación creamos el fichero Emfatic donde escribiremos el código de nuestro metamodelo con las correspondientes anotaciones. El fichero debe tener extensión `emf`.



New > File...

Código (I)

```
@namespace(uri="http://ermdiagram/1.0",
           prefix="ermdiagram")

package ermdiagram;

enum ERMDDataType {
    String = 0;
    Number = 1;
    Decimal = 2;
    Date = 3;
}

@gmf.diagram(foo="bar")
class ERMDiagram {
    val Attribute[*] attributes;
    val Relationship[*] relationship;
    val Entity[*] entities;
}
```

Código (II)

```
@gmf.node(label="name", figure="ellipse",  
          color="135,135,135")
```

```
class Attribute {  
    attr String name;  
    attr ERMDDataType type;  
}
```

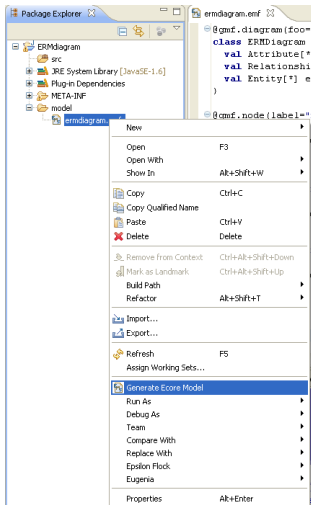
```
@gmf.node(label="name", color="255,153,0")  
class Relationship {  
    attr String name;  
}
```

<http://www.colorpicker.com>

Código (III)

```
@gmf.node(label="name", figure="rectangle",  
          color="0,255,0")  
  
class Entity {  
    attr String name;  
    @gmf.link(target.decoration="arrow",  
             style="solid")  
    ref Relationship[*] entityRelationship;  
    @gmf.link(target.decoration="arrow",  
             style="dash")  
    ref Attribute[*] entityAttribute;  
}
```

A partir del anterior código en Emfatic, podemos generar el metamodelo Ecore, así como el diagrama.



Paso a diagrama (I)

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

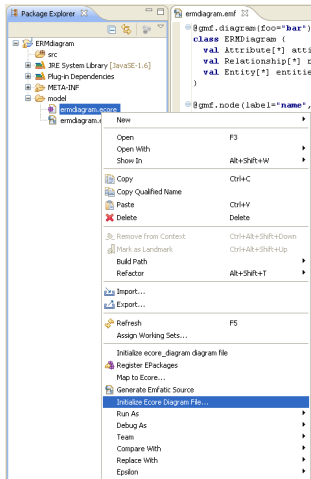
GMF

Ejemplo

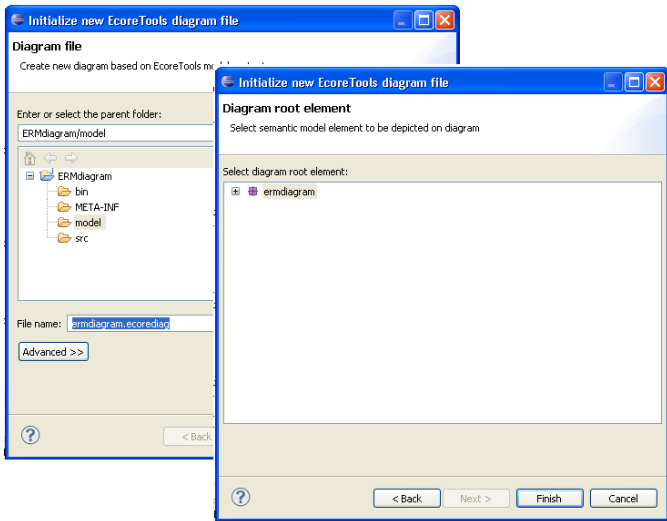
Proyecto GMF

Fichero Emfatic

Conclusiones



Paso a diagrama (II)



Diagrama

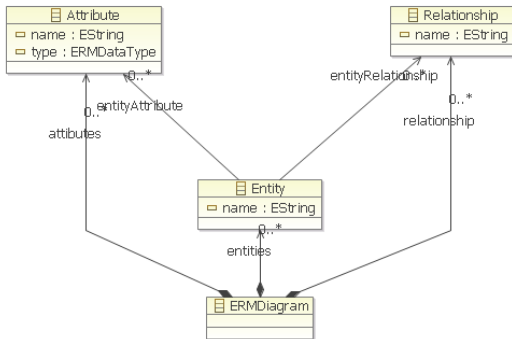
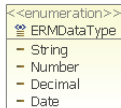
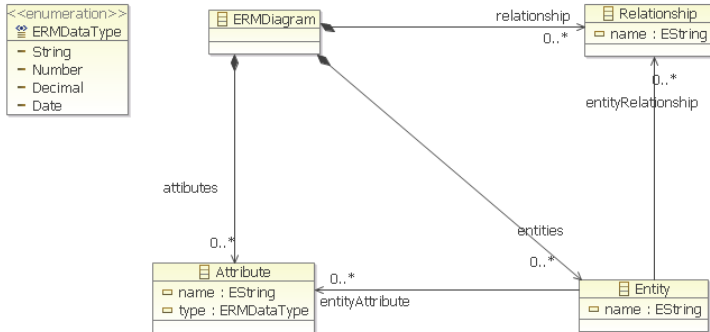


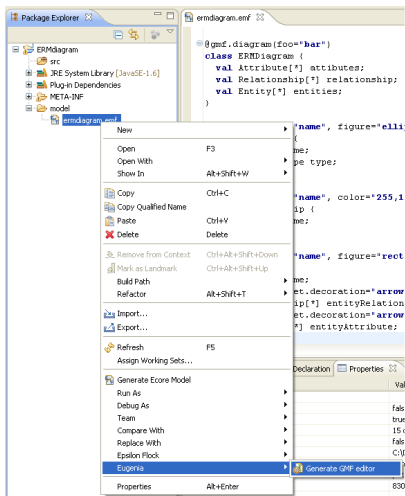
Diagrama modificado.



Una vez generado el metamodelo Ecore, podríamos generar el editor en varios pasos:

- Right-click your Ecore metamodel and select Eugenia > Generate GMF graph, tool and map models
- Create a .genmodel from your Ecore metamodel by selecting the .ecore in the "Domain Model" box of the dashboard and then follow the wizard to "derive" the "Domain Gen Model".
- Right click on the root of the Gen Model editor and "Generate All" to create the java code from the .genmodel
- Right click the generated .gmfmap model and select Create generator model
- Right click the generated .gmfgen model and select Eugenia > Synchronize GMF gen model
- Right click the .gmfgen model again and select Generate diagram code

Pero esto no es necesario. Podemos crear nuestro editor a partir de un fichero con el código en Emfatic correctamente anotado.



Se generan automáticamente todos los modelos necesarios, así como el código final.

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

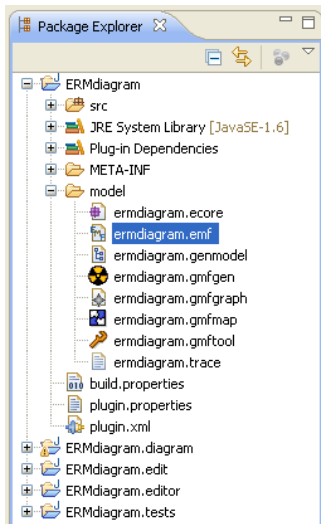
GMF

Ejemplo

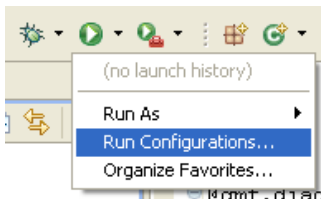
Proyecto GMF

Fichero Emfatic

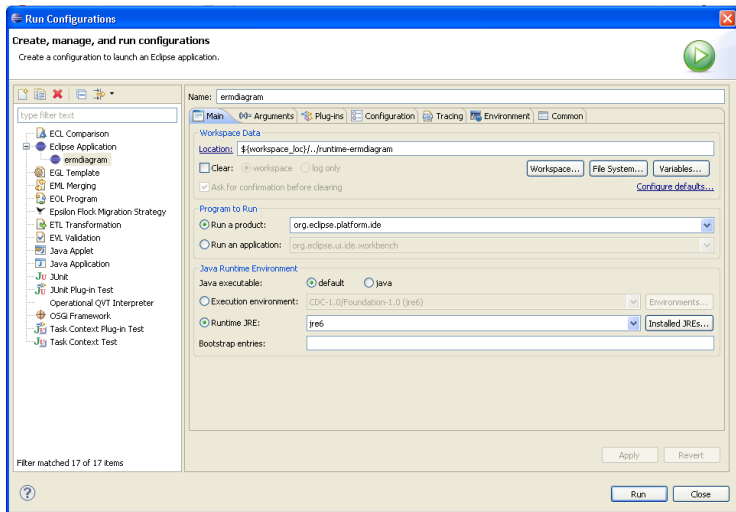
Conclusiones



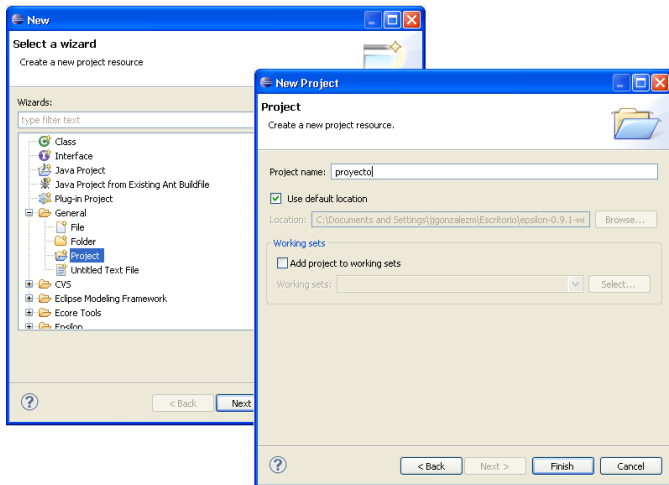
Es el momento de ejecutar nuestro editor gráfico.



Creamos una nueva configuración como *Eclipse Application*.



Ya tenemos la instancia de Eclipse donde podremos hacer de nuestro editor gráfico. Para ello, creamos un nuevo proyecto



Creamos una instancia del tipo de nuestro modelo. En este caso ERMDiagram.

Instalación

EuGENia

Epsilon

Emfatic

EuGENia

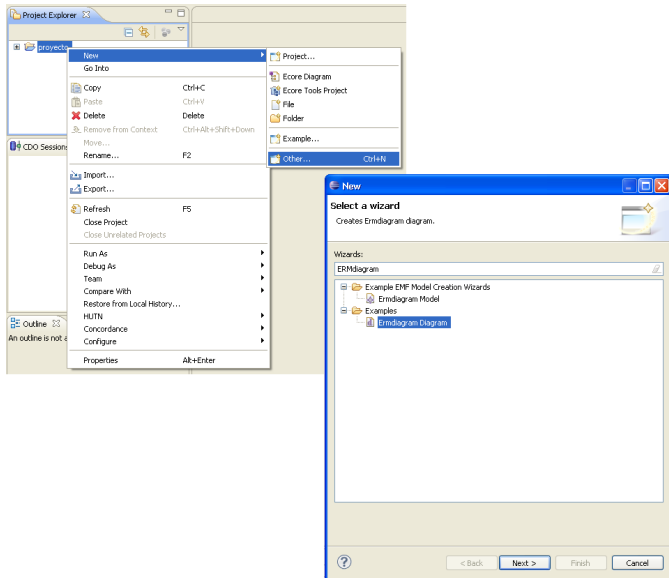
GMF

Ejemplo

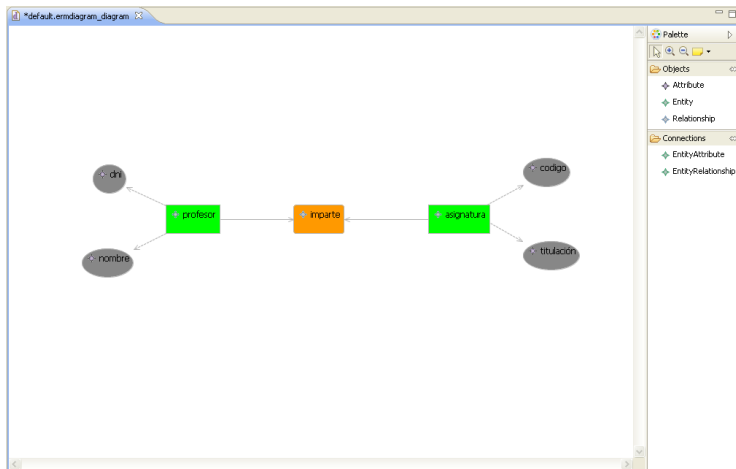
Proyecto GMF

Fichero Emfatic

Conclusiones



Ya podemos usar nuestro editor.



Ejemplo de la documentación (I)

```
@namespace(uri="filesystem", prefix="filesystem")
@gmf(foo="bar")
package filesystem;

@gmf.diagram(foo="bar")
class Filesystem {
    val Drive[*] drives;
    val Sync[*] syncs;
}

class Drive extends Folder {

}

class Folder extends File {
    @gmf.compartment(foo="bar")
    val File[*] contents;
}
```

Ejemplo de la documentación (II)

```
class Shortcut extends File {
    @gmf.link(target.decoration="arrow", style="das
    ref File target;
}

@gmf.link(source="source", target="target",
          style="dot", width="2")
class Sync {
    ref File source;
    ref File target;
}

@gmf.node(label = "name")
class File {
    attr String name;
}
```

1 Instalación

2 EuGENia

Epsilon
Emfatic
EuGENia

3 GMF

4 Ejemplo

Proyecto GMF
Fichero Emfatic

5 Conclusiones

- Herramienta muy útil para principantes en la materia, pero con una base de programación.
- Más controlable por el método con más pasos.
- En algunos proyectos, si queremos hacer algo grande y correcto desde un principio, suele ser mejor recorrer el camino largo sin intermediarios.