

Procesadores de Lenguajes 2

# Desarrollo de editores gráficos con GMF

Curso 2013-2014

Iván Ruiz Rube  
Departamento de Ingeniería Informática  
Escuela Superior de Ingeniería  
Universidad de Cádiz

## En la clase anterior...

- Estudiamos cómo construir editores de modelos reflexivos (basados en árbol) utilizando Eclipse EMF.
- Despliegue de editores como plugins de Eclipse (PDE) y cómo productos independientes (RCP).



# Contenidos

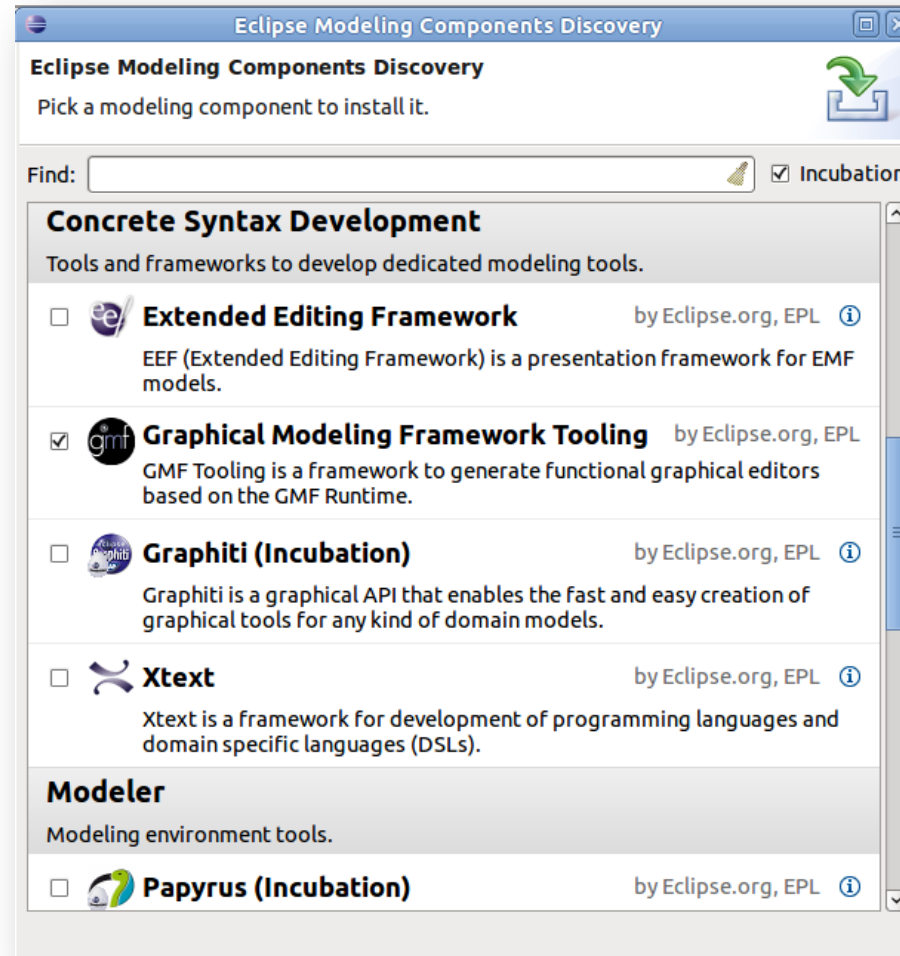
- Introducción
- Proceso de desarrollo
- Un editor de diagramas E/R

DESARROLLO DE EDITORES GRÁFICOS CON GMF



# INTRODUCCIÓN

# Instalación GMF



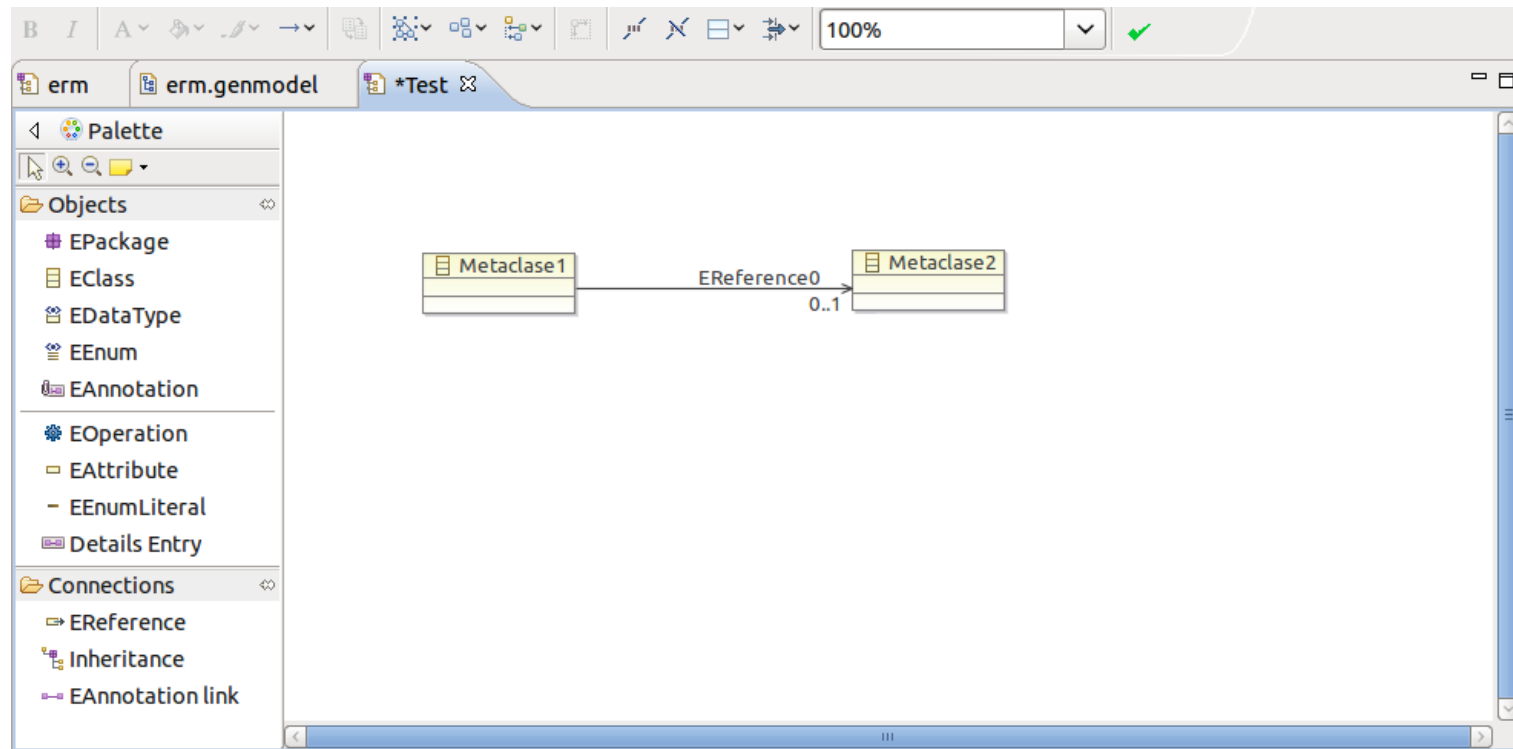
Help → Install Modeling Components



# Graphical Modeling Framework

- GMF es el proyecto de Eclipse que nos ayudará a construir editores gráficos para nuestros DSLs.
- *GMF Tooling*: componente que nos permitirá definir la funcionalidad del editor y automáticamente construir el plugin correspondiente.
- *GMF Runtime*: componente encargado de ejecutar el editor visual
- *GMF Notation*: componente que proporciona el mecanismo para persistir e intercambiar información de los diagramas visuales.

# Ejemplo editor GMF



El propio editor visual de metamodelos en EMF es un ejemplo de editor gráfico, con el cual podemos diseñar modelos de dominio basados en Ecore.

# Características

- Desarrollo de editores visuales sin escribir código.
- Edición de propiedades visuales: colores, fuentes, estilos de líneas, etc.
- Funciones adicionales como la ordenación automática de elementos, reglas, zoom...
- Edición de etiquetas de los elementos directamente sobre el área de trazado.



## Características (II)

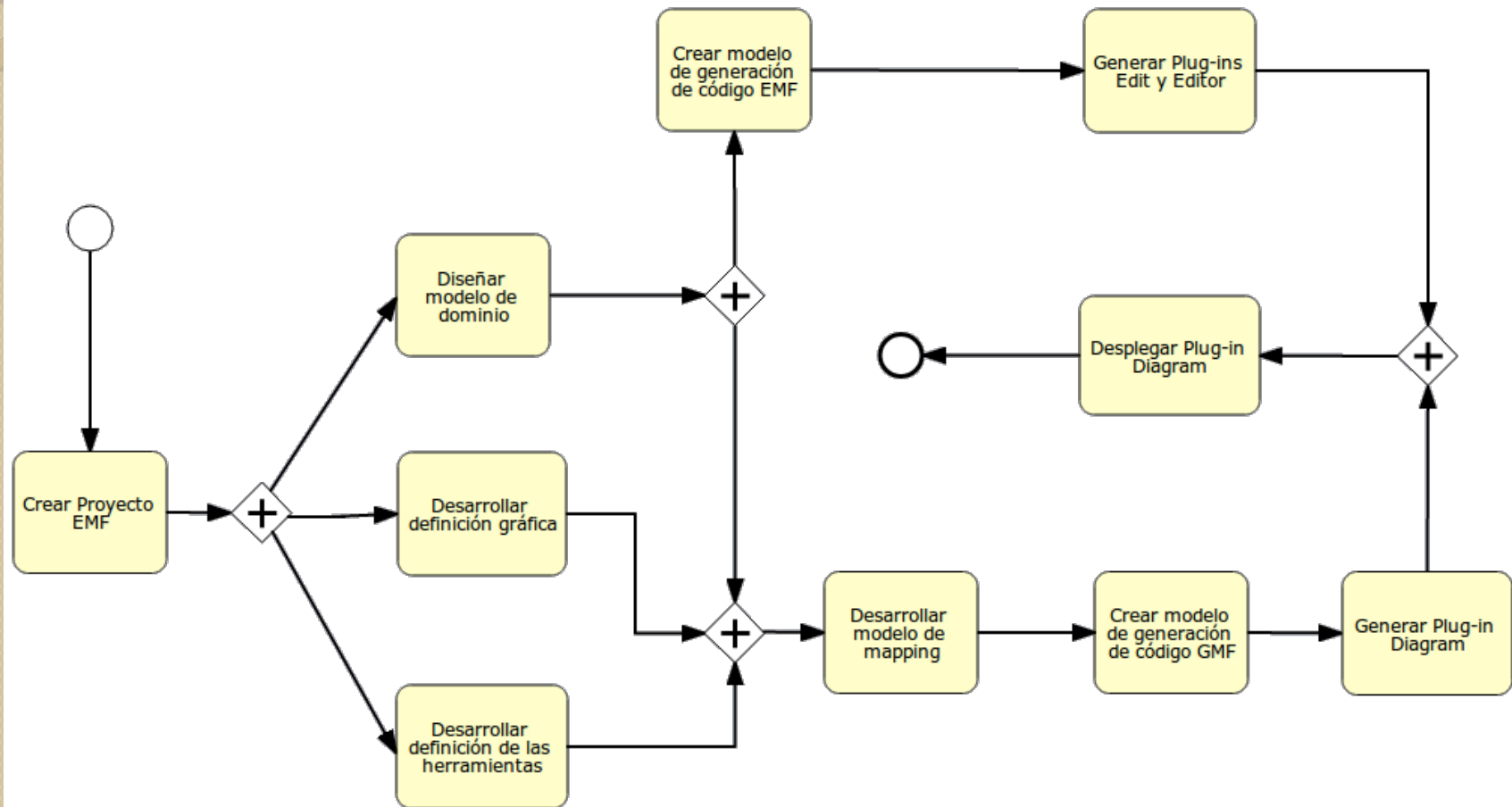
- Exportación de modelos a varios formatos de imágenes.
- Integración con el clipboard del sistema.
- Utilización de elementos visuales como contenedores de otros elementos.
- Integración de restricciones y validaciones mediante el lenguaje OCL.

DESARROLLO DE EDITORES GRÁFICOS CON GMF

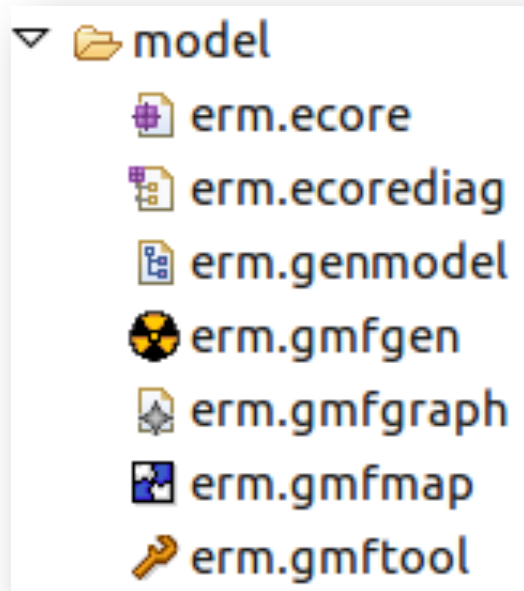


# PROCESO DE DESARROLLO

# Proceso generativo de desarrollo



# Modelos implicados



- Dominio (.ecore)
- Generación de código EMF (.genmodel)
- Definición gráfica (.gmfgraph)
- Herramientas (.gmftool)
- Mapeo entre los modelos anteriores (.gmfmap)
- Generación de código GMF (.gmfgen)

# Desarrollo con GMF

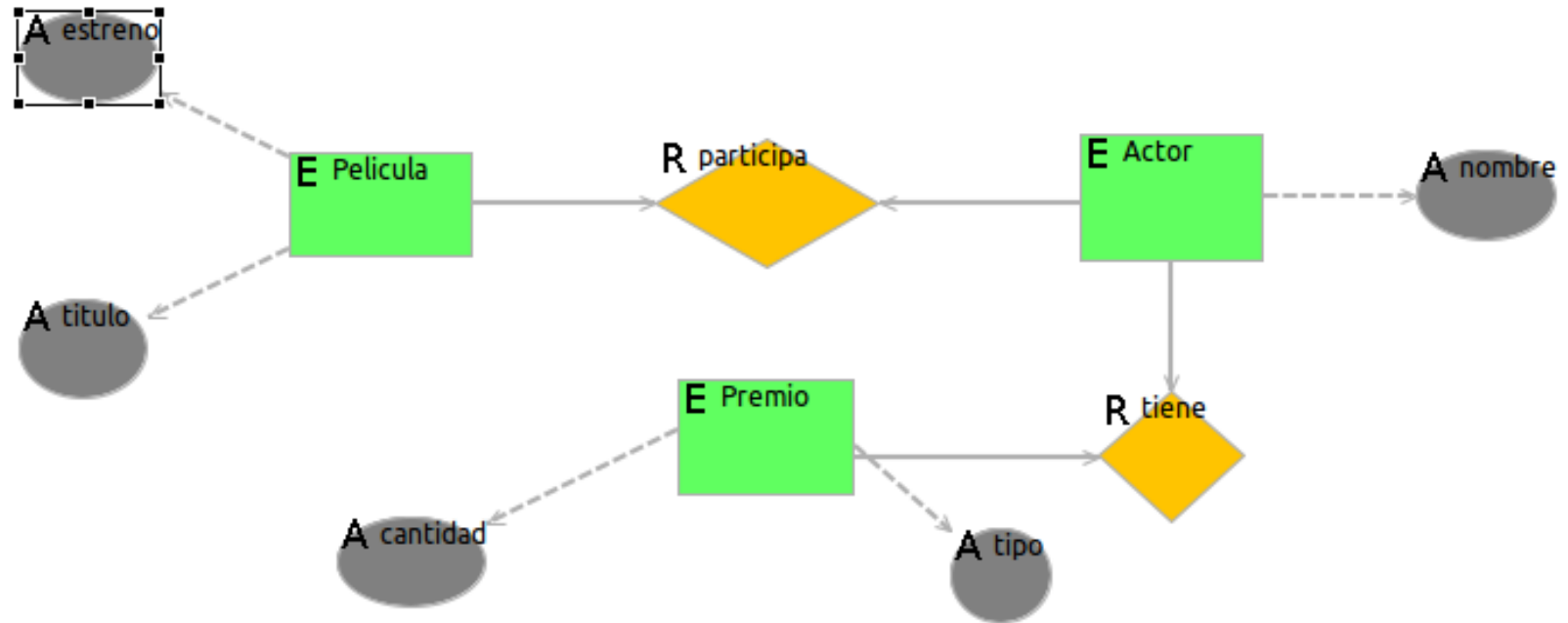
- El desarrollo de editores GMF no es del todo sencillo.
- La generación automática de modelos no es determinista 100%.
- Tenemos que **revisar, ajustar y validar** cada modelo generado.
- En caso de tener que regenerar un proyecto o un modelo, debemos previamente borrarlo del sistema de ficheros.

DESARROLLO DE EDITORES GRÁFICOS CON GMF



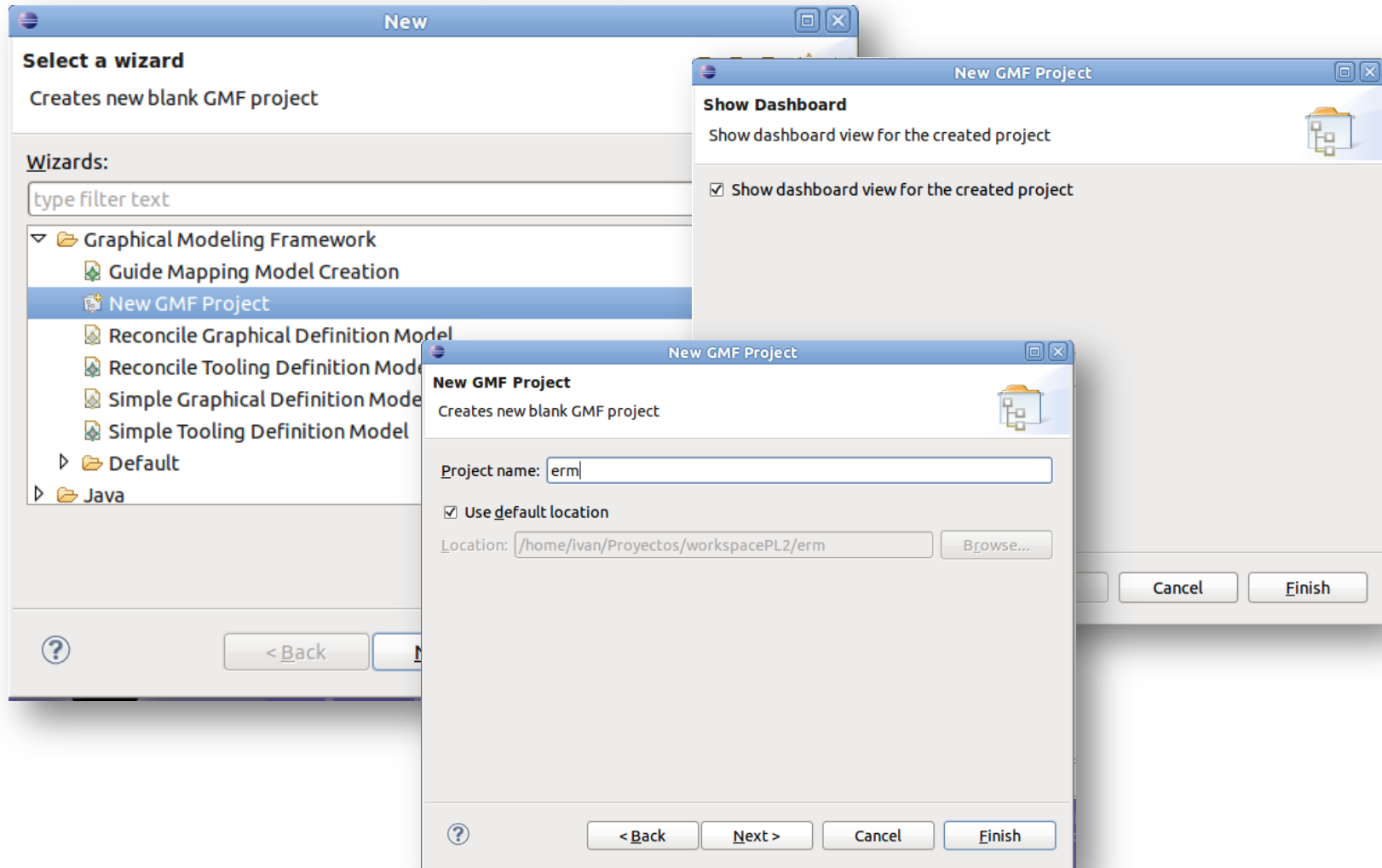
# **UN EDITOR DE DIAGRAMAS E/R**

# Diagrama entidad/relación



Vamos a construir nuestro primer DSL visual!!

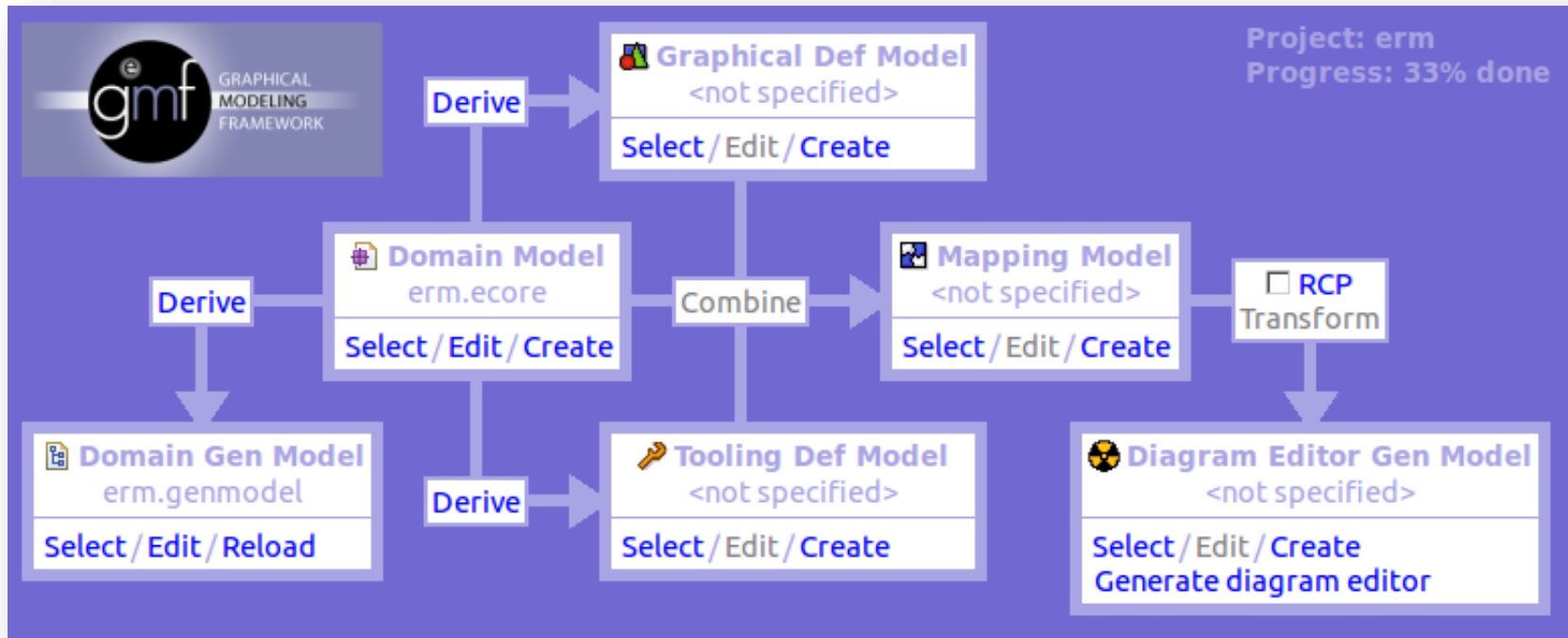
# Creación proyecto GMF



File → New → GMF Project...

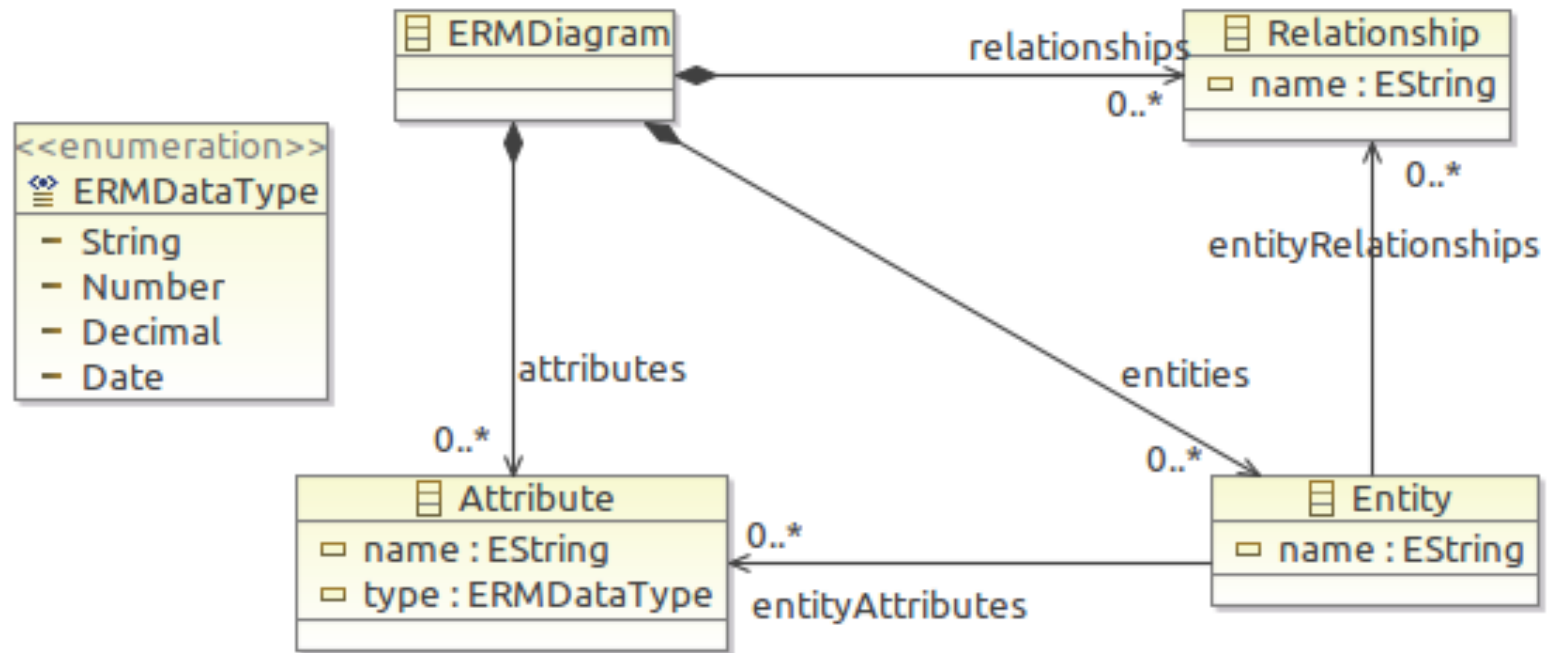


# GMF Dashboard



Esta vista nos guiará durante gran parte del proceso de construcción del editor gráfico

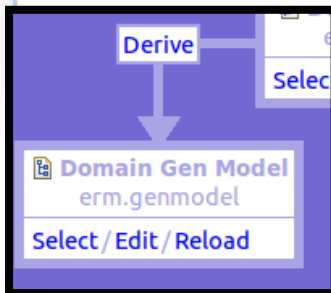
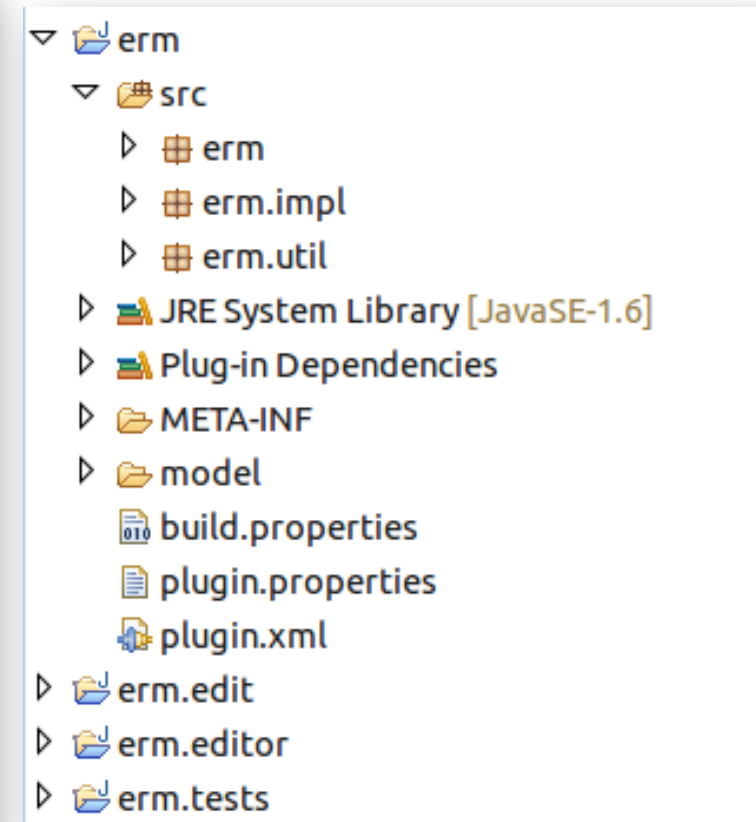
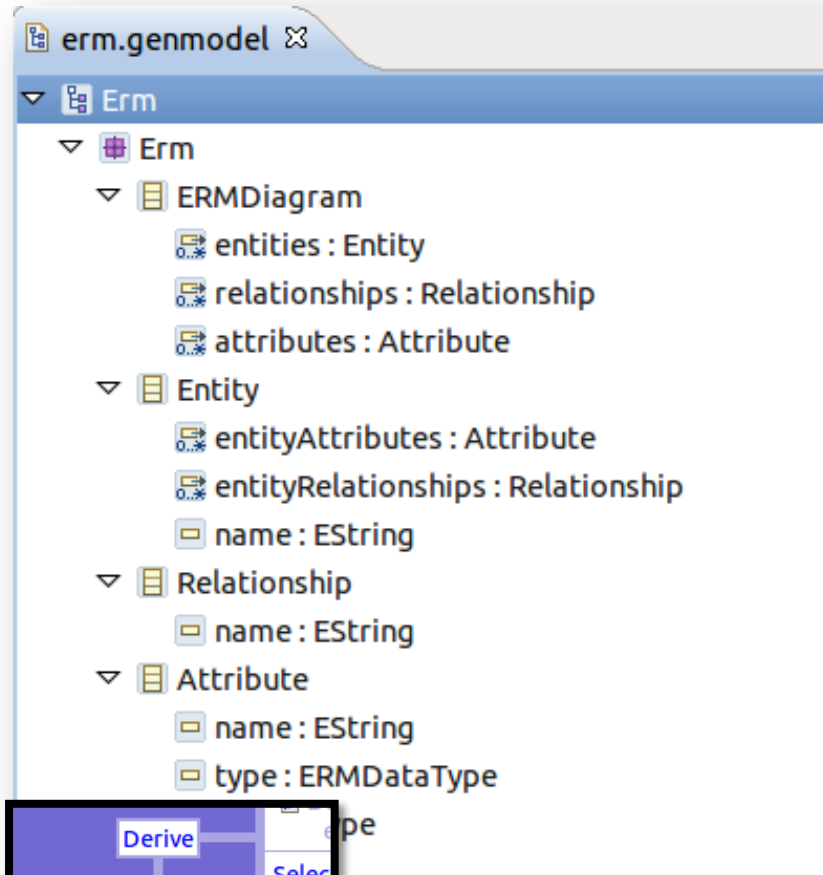
# Diseñar metamodelo (modelo de dominio)



File → New → Ecore Diagram...

**Recordatorio: establecer las propiedades Name, Lower Bound, Upper Bound, Is Containment, ID y EType**

# Derivar modelo de generación de código EMF y generar plug-ins



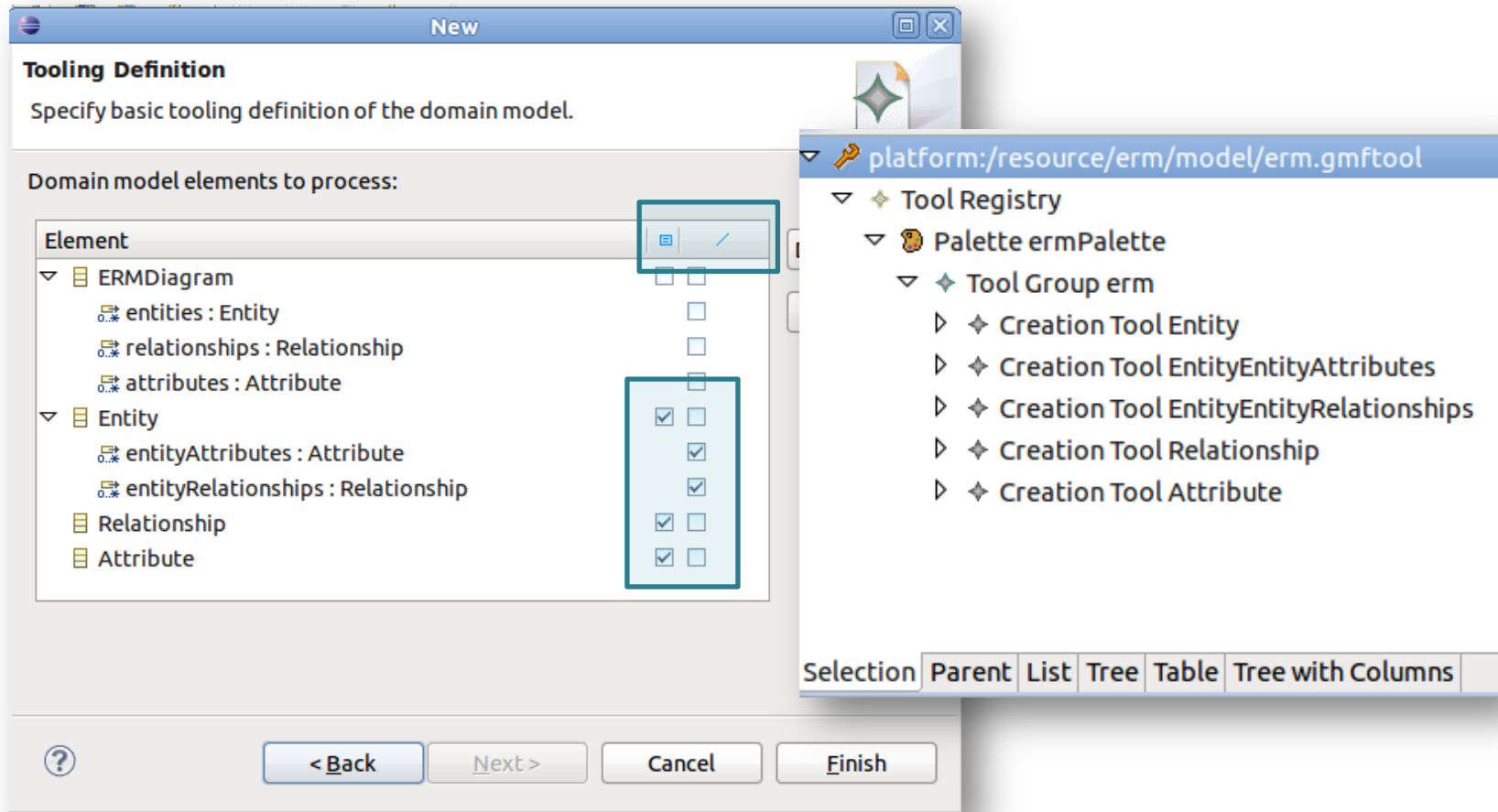
[.genmodel] Generate All...

# Derivar modelo de definición de herramientas

The image shows two overlapping dialog boxes from a software development environment. The background dialog is titled "New GMFTool Model" and prompts the user to "Create a new GMFTool model". It includes a field for the parent folder (set to "erm/model"), a file explorer showing a directory structure with folders like "META-INF" and "model", and a "File name" field containing "erm.gmftool". The foreground dialog is titled "New Domain Model" and prompts the user to "Select file with ecore domain model". It features a "Model URI" field with the value "platform:/resource/erm/model/erm.ecore" and a "Load" button. Below this, a "Diagram Element" list includes "Attribute", "ERMDiagram" (which is selected), "Entity", and "Relationship". At the bottom of the foreground dialog are buttons for "< Back", "Next >", "Cancel", and "Finish".

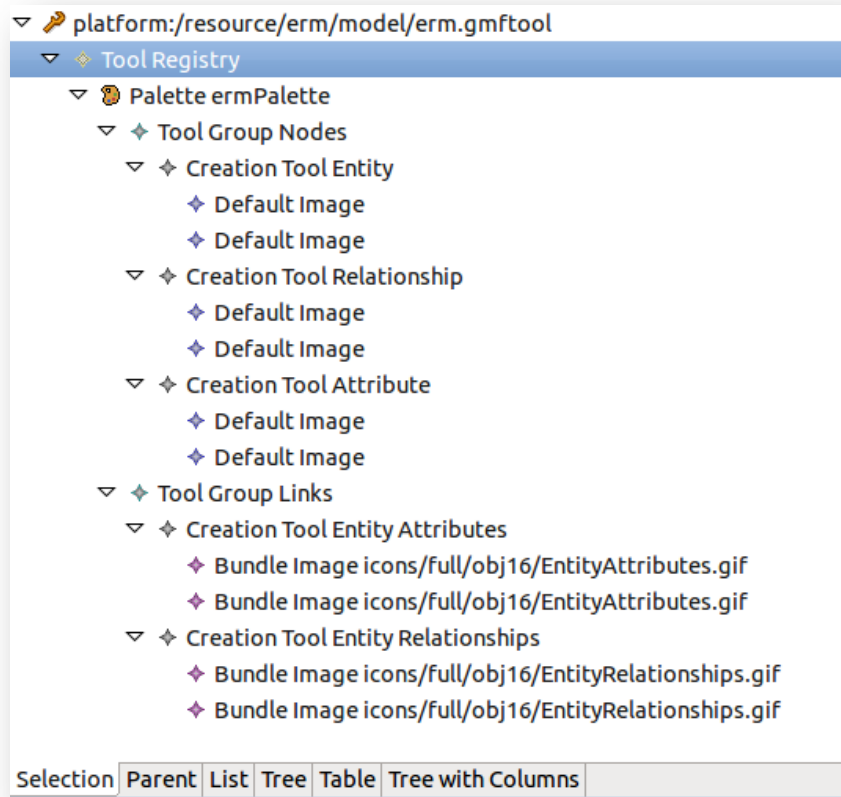
Below the dialog boxes, a diagram illustrates the derivation process. A box labeled "Derive" has an arrow pointing to a box labeled "Tooling Def Model <not specified>". The "Derive" box is connected to a larger box containing "Select / Edit / Create" and "Select".

# Generación de herramientas



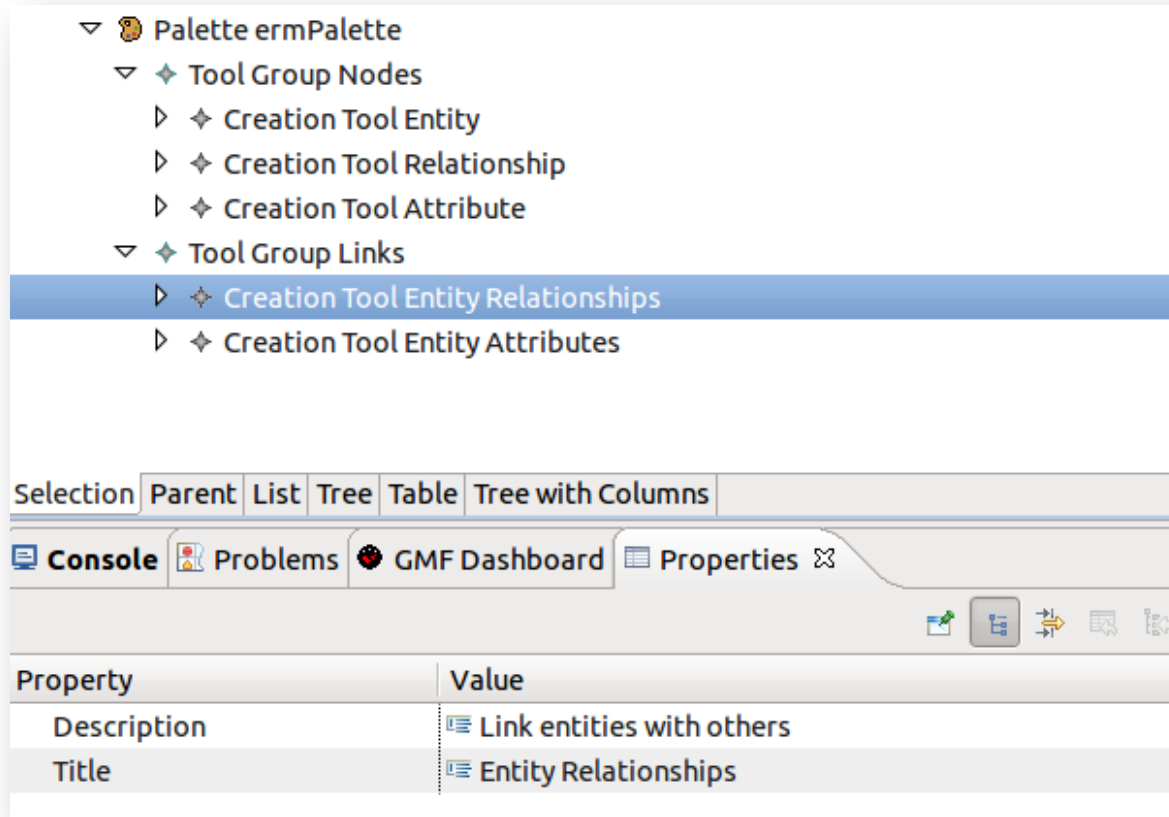
Con este asistente podemos construir herramientas (de tipo nodo o tipo conexión) para los elementos de nuestro metamodelo.

# Modelo de herramientas



- GMF permite personalizar las herramientas y cómo éstas aparecen en la paleta.
- Cada herramienta utiliza icono gráfico (metáfora) para representar el concepto del metamodelo a utilizar en nuestro editor.

# Ajuste de las herramientas (I)



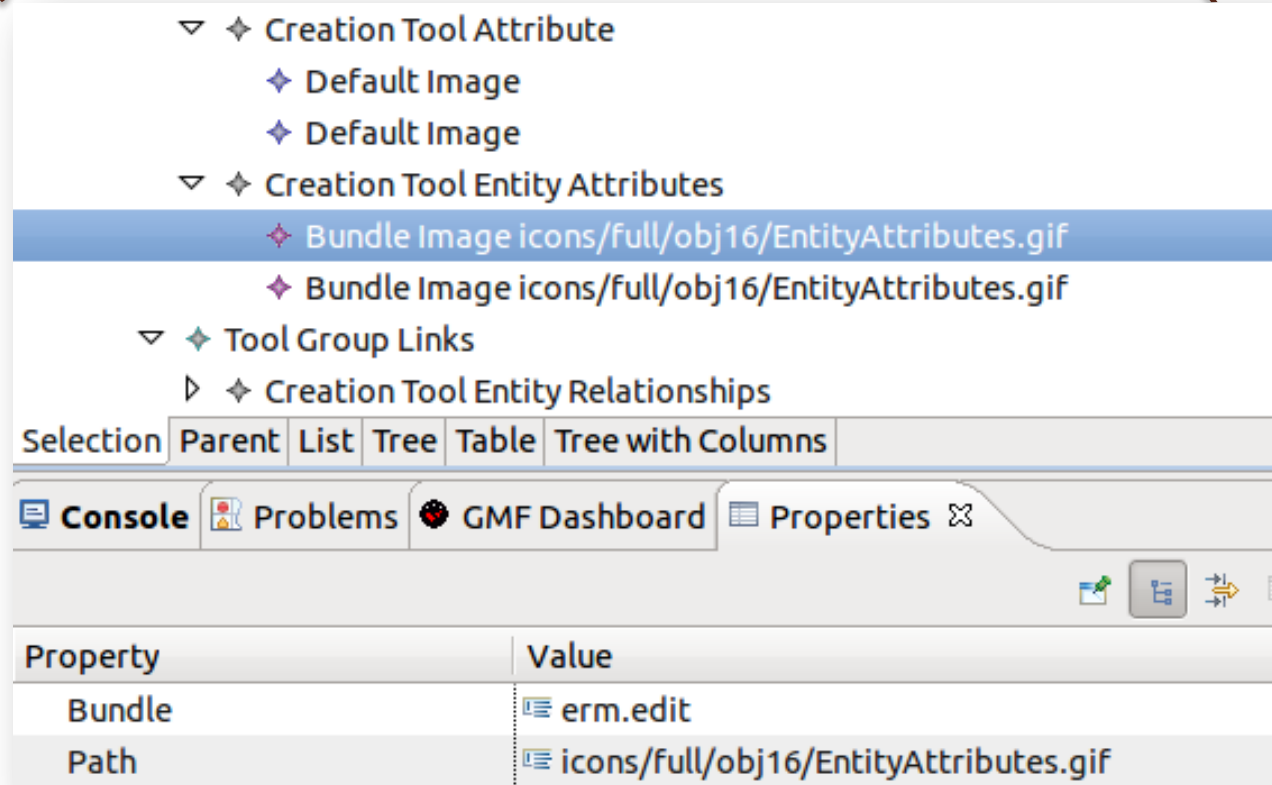
The screenshot shows the Eclipse IDE interface. The Palette view displays a tree structure of tool groups. The 'Creation Tool Entity Relationships' tool group is selected and highlighted in blue. Below the palette, the Properties view is open, showing the following properties:

Property	Value
Description	Link entities with others
Title	Entity Relationships

Personalizar propiedades *Description* y *Title* de las herramientas (*Creation Tools*) y definir grupos (*Tool Groups*), estableciendo la propiedad *Collapsible* a *true*.



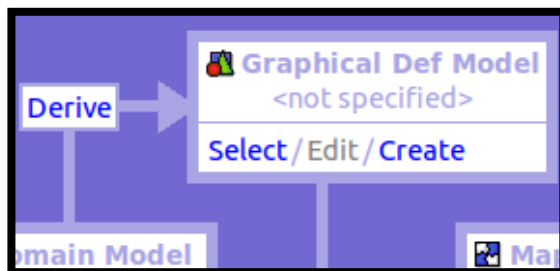
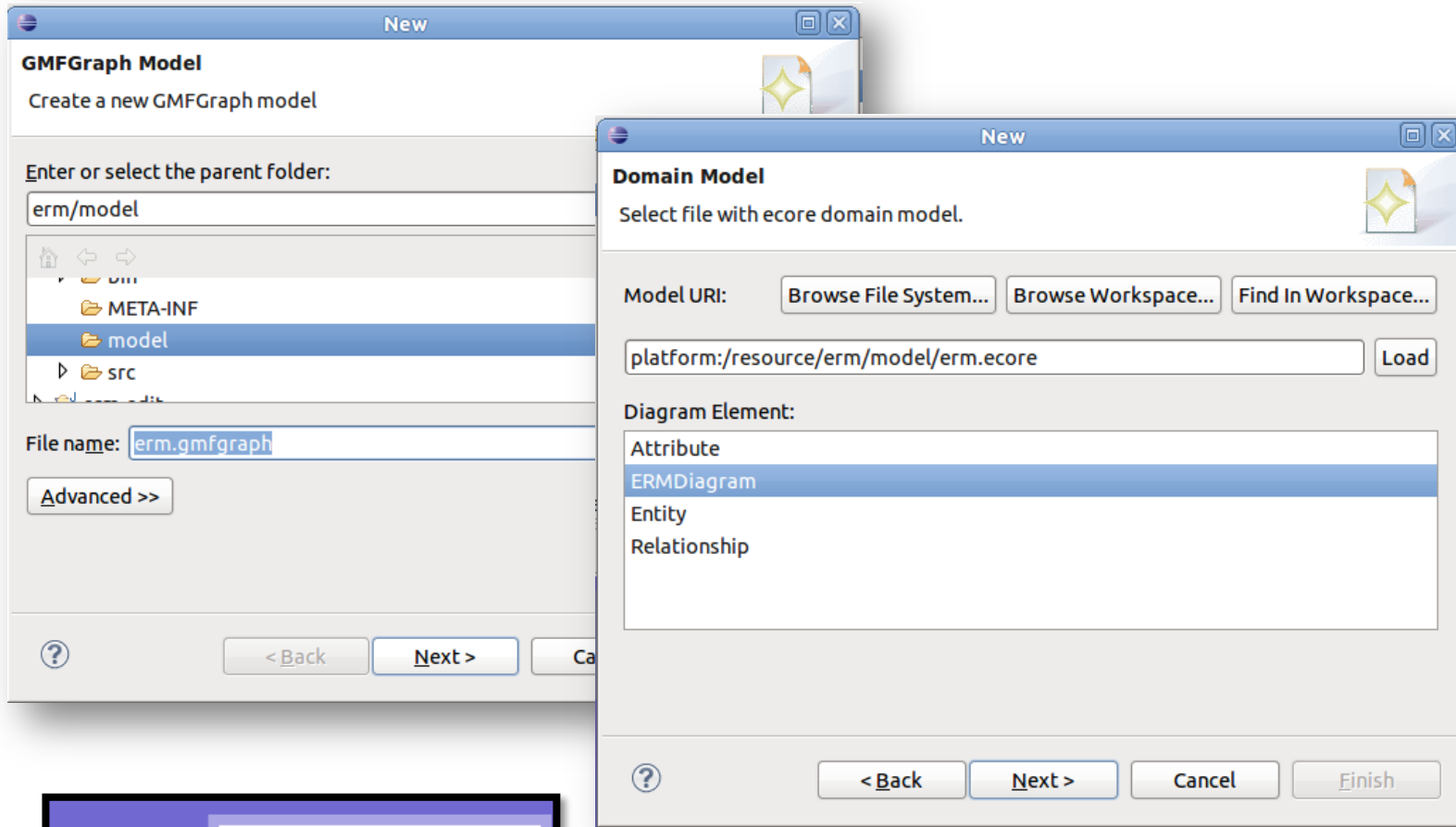
# Ajuste de las herramientas (II)



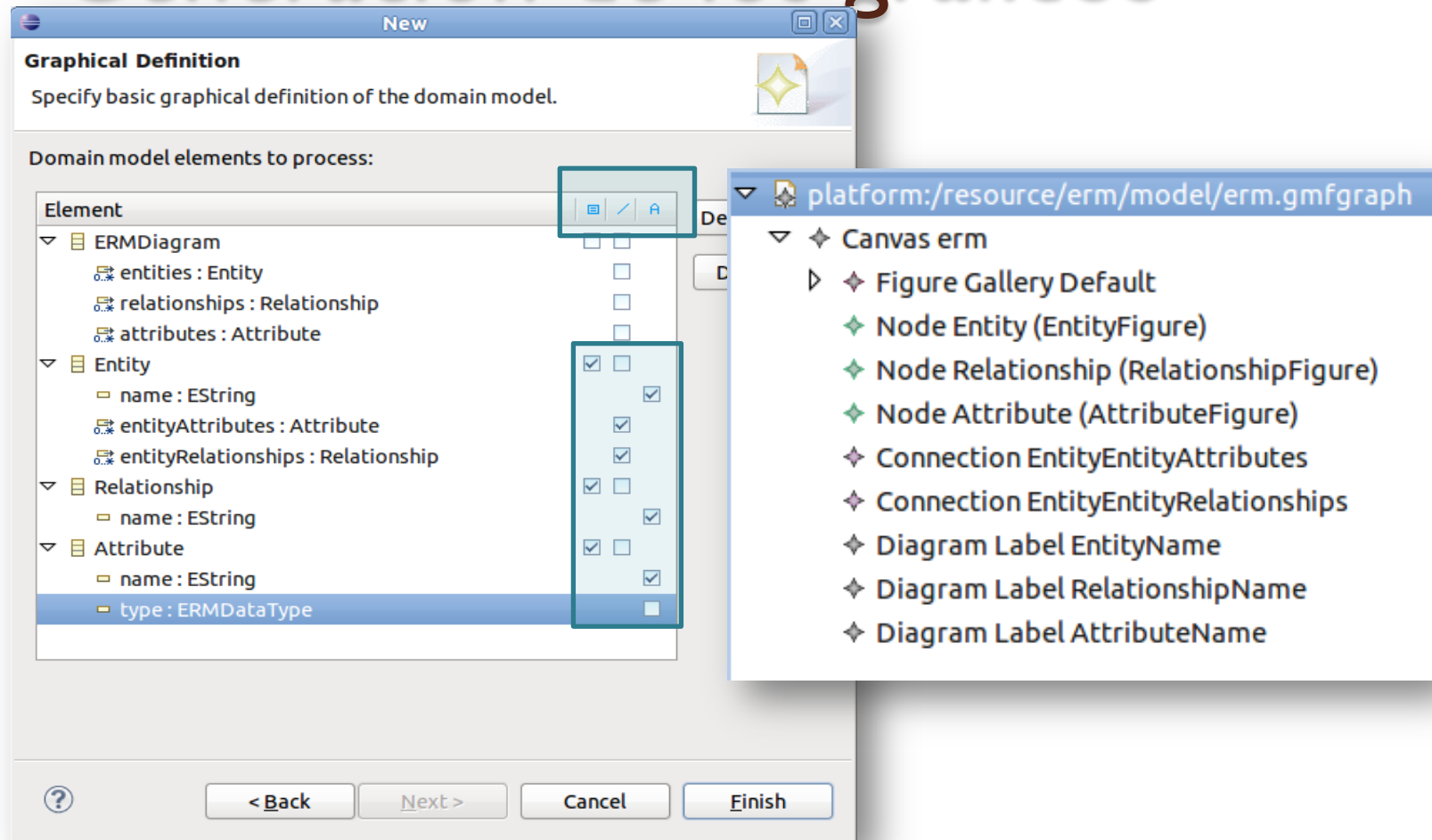
Para modificar los iconos de las herramientas de tipo nodo, tenemos que reemplazar las imágenes (*Default Images*) que se encuentran dentro el proyecto *.edit*. Para las herramientas de tipo link, debemos añadir las imágenes y luego referenciarlas (*Bundle Images*).



# Derivar modelo de definición gráfica



# Generación de los gráficos



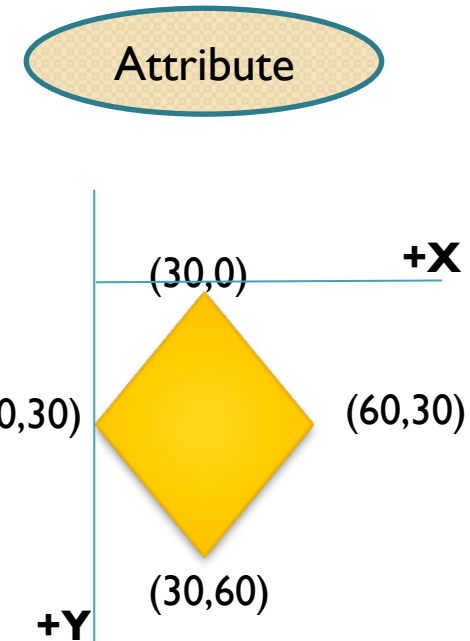
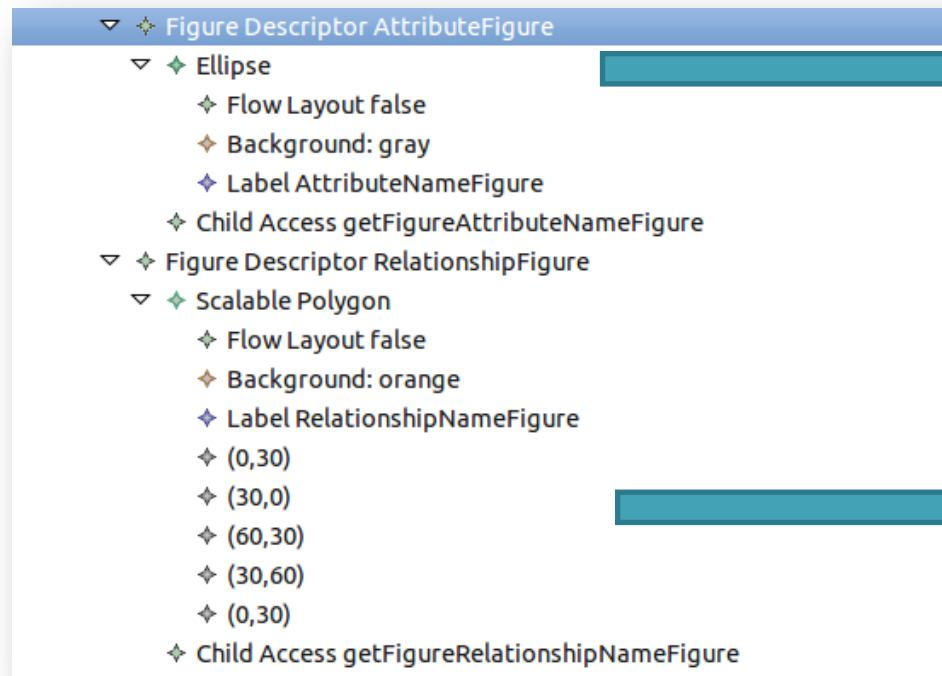
Con este asistente podemos construir los elementos gráficos (nodos, conexiones o etiquetas) para los elementos de nuestro metamodelo.

# Modelo de definición gráfica

```
platform:/resource/erm/model/erm.gmfgraph
  ✦ Canvas erm
  ▾ ✦ Figure Gallery Default
    ✦ Polyline Decoration EntityEntityAttributesTargetDecorat
    ✦ Polyline Decoration EntityEntityRelationshipsTargetDeco
    ▷ ✦ Figure Descriptor EntityFigure
    ▷ ✦ Figure Descriptor AttributeFigure
    ▷ ✦ Figure Descriptor RelationshipFigure
    ▷ ✦ Figure Descriptor EntityEntityAttributesFigure
    ▷ ✦ Figure Descriptor EntityEntityRelationshipsFigure
    ✦ Node Entity (EntityFigure)
    ✦ Node Relationship (RelationshipFigure)
    ✦ Node Attribute (AttributeFigure)
    ✦ Connection EntityEntityAttributes
    ✦ Connection EntityEntityRelationships
    ✦ Diagram Label EntityName
    ✦ Diagram Label RelationshipName
    ✦ Diagram Label AttributeName
```

- GMF permite dibujar formas geométricas escalares e imágenes personalizadas sobre el área de trazado (*canvas*).
- Cada elemento gráfico es altamente personalizable: colores, texturas, textos, etc.

# Ajuste de la definición gráfica (I)



Los elementos *Figure Descriptor* engloban los aspectos gráficos y sus propiedades visuales. Para cambiar el tipo de gráfico, es necesario crear un nuevo descriptor, añadirle el gráfico deseado y trasladar sus propiedades.

# Ajuste de la definición gráfica (II)

The screenshot shows an IDE interface. The top part is a project tree with the following structure:

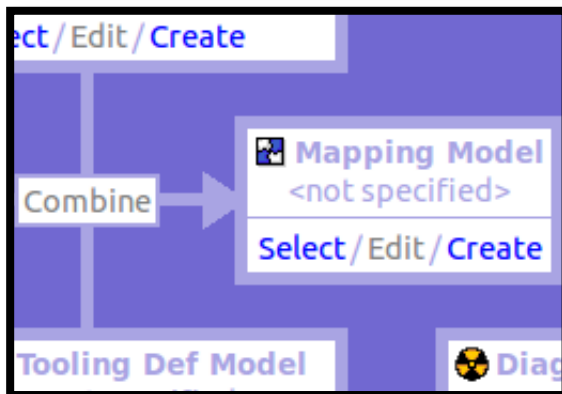
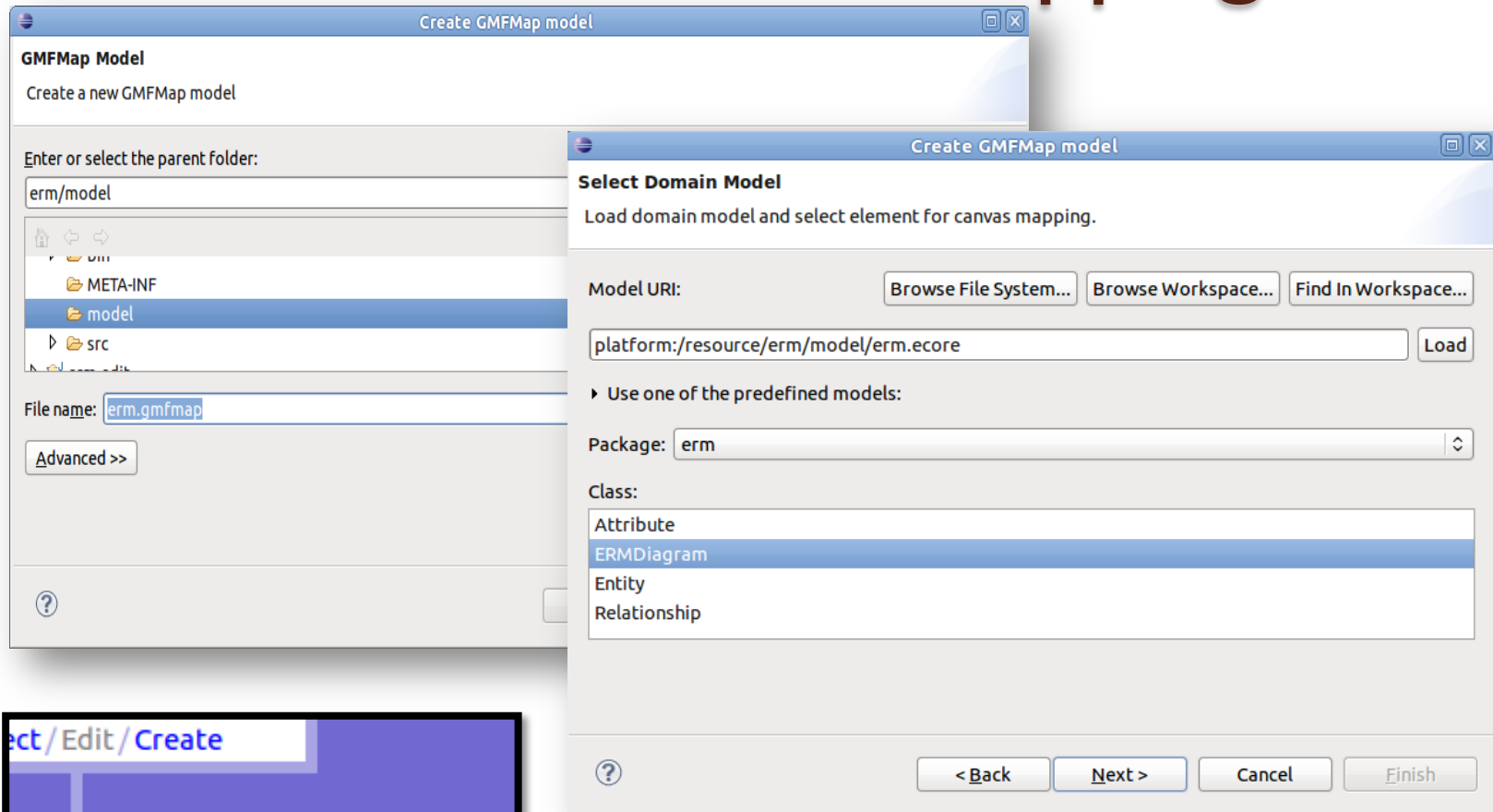
- Canvas erm
  - Figure Gallery Default
    - Node Entity (EntityFigure)**
    - Node Relationship (RelationshipFigure)
    - Node Attribute (AttributeFigure)
    - Connection EntityEntityAttributes
    - Connection EntityEntityRelationships
    - Diagram Label EntityName
    - Diagram Label RelationshipName
    - Diagram Label AttributeName

The bottom part is a properties window with the following table:

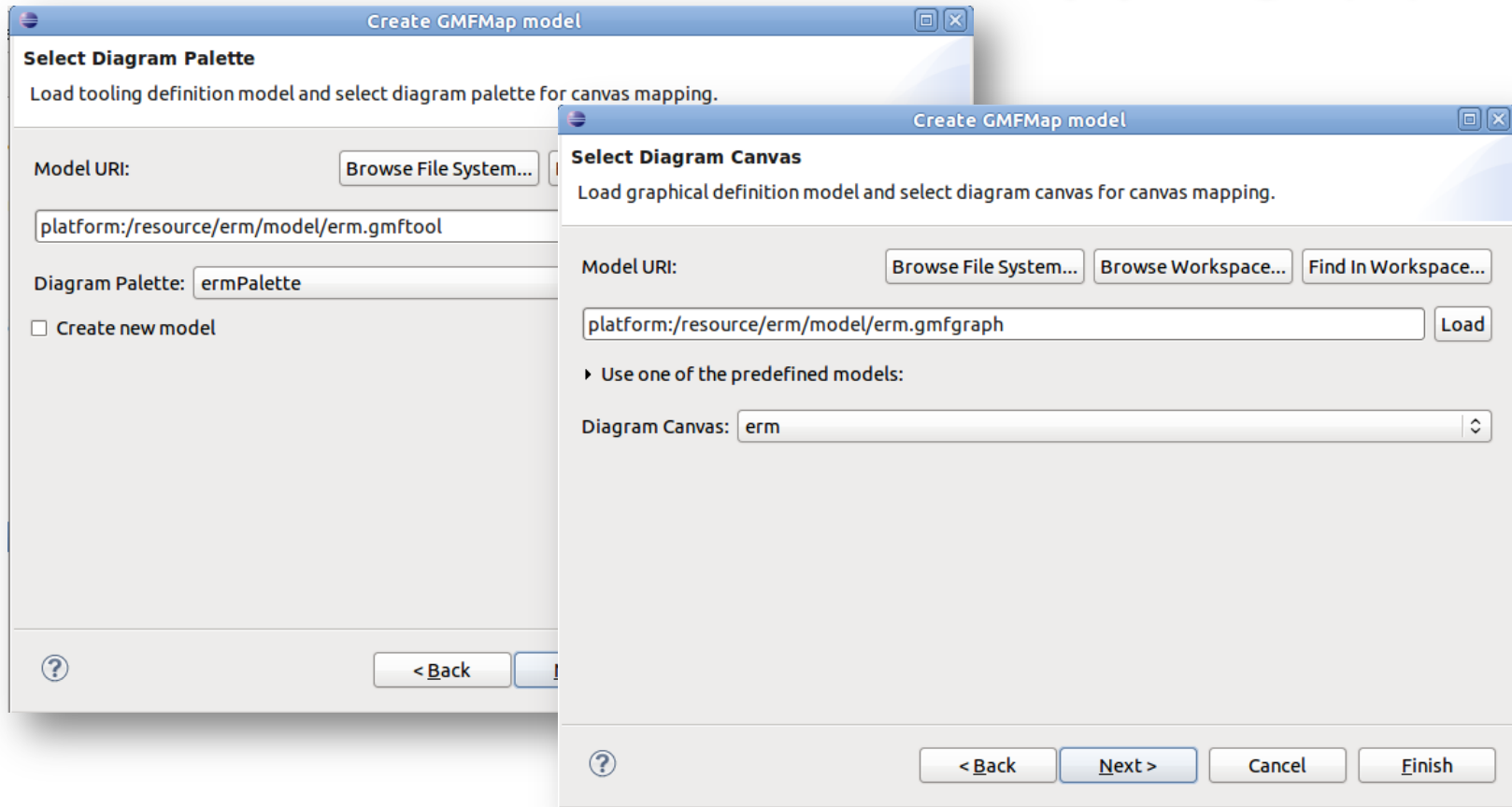
Property	Value
Affixed Parent Side	NONE
Content Pane	
Figure	Figure Descriptor EntityFigure
Name	Entity
Resize Constraint	NSEW

Tenemos que revisar que cada elemento *Node*, *Connection* o *Diagram Label* tenga asociado su *Figure Descriptor*.

# Derivar modelo de mapping



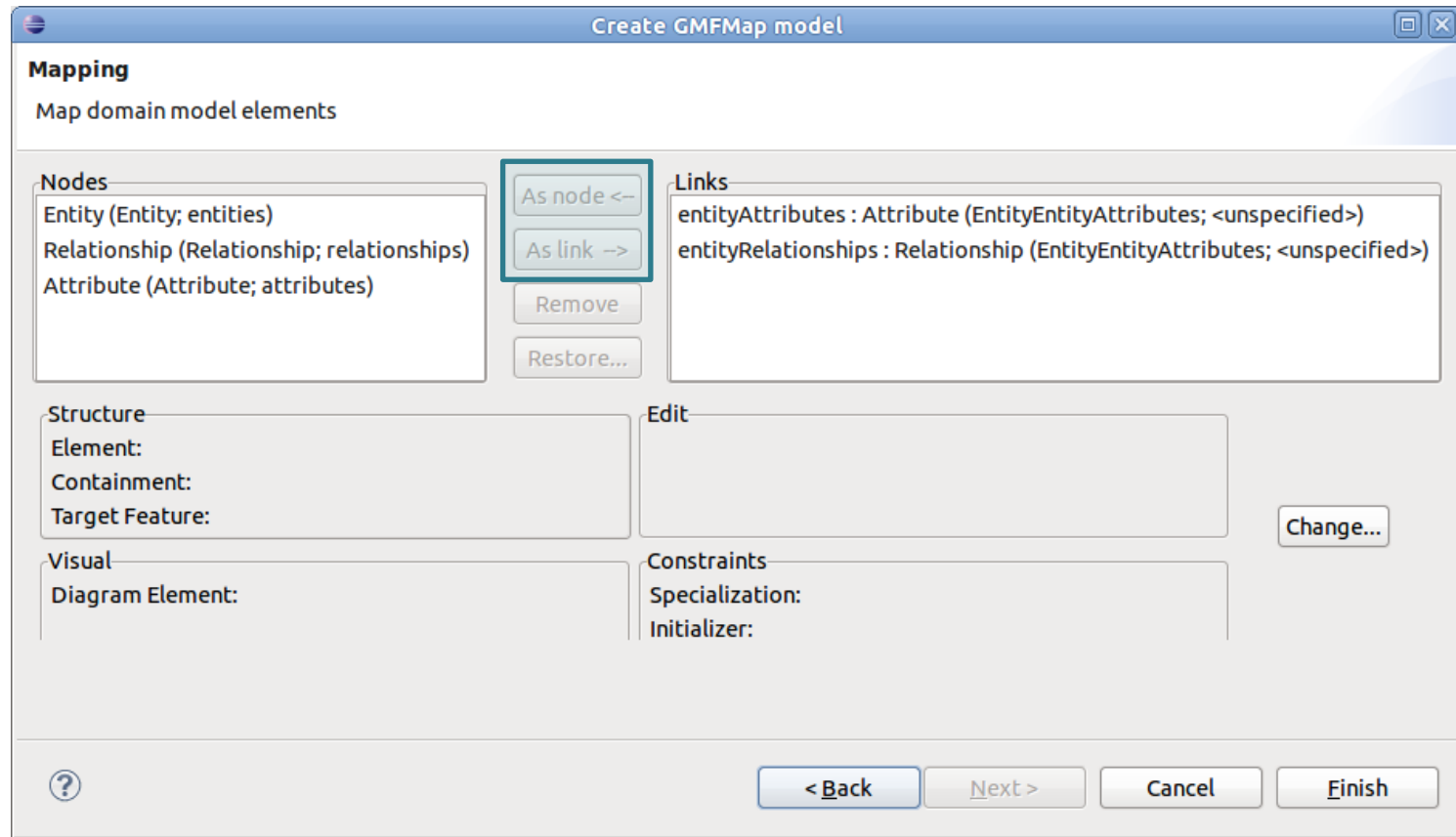
# Derivar modelo de mapping (II)



Nuestro mapping combinará los elementos del modelo de dominio, los elementos gráficos y las herramientas.



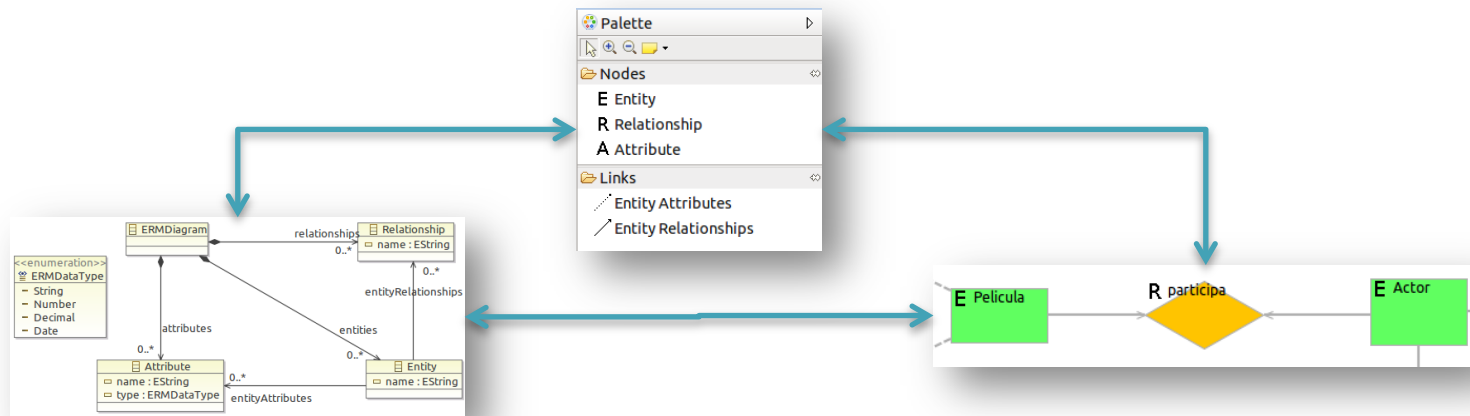
# Generación del mapping



Tenemos que confirmar qué elementos del dominio son nodos (*as node* ←) y cuáles son links (*as link* →)

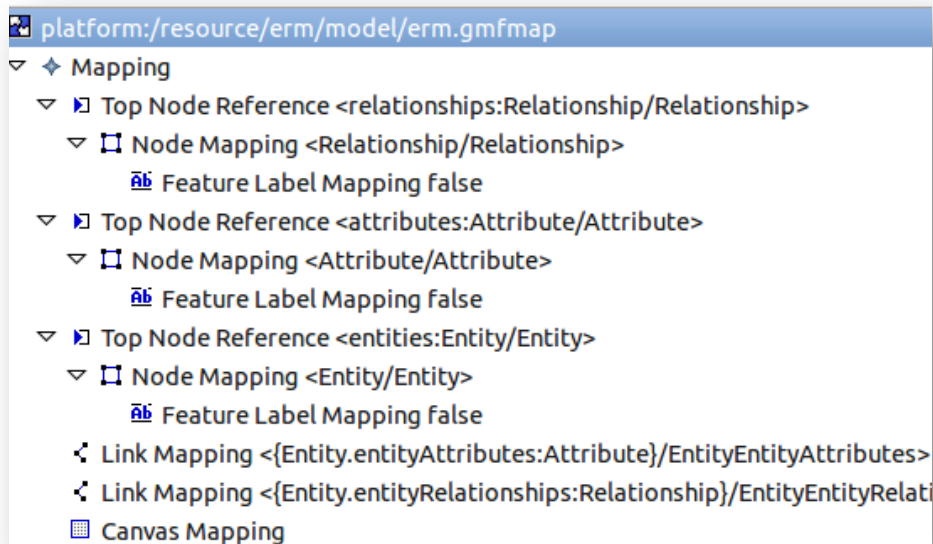


# Modelo de mapping (I)



Gmfmap Element	Ecore Element	Gmfgraph Element	Gmftool Element
Canvas Mapping	Metaclase raiz	Canvas	Palette
Node Mapping	Metaclase	Node	Creation Tool
Feature Label Mapping	Metaclase.atributoID	Label	-
Link Mapping	Metaclase.asociación	Connection	Creation Tool

# Modelo de mapping (II)



- Este modelo permite vincular los elementos del metamodelo, con su definición gráfica y su herramienta.
- El modelo de mapping generado de forma automática por GMF necesita ser completado y validado.

# Ajuste del mapping (I)

The screenshot shows the GMF Properties window for a Node Mapping. The 'Domain meta information' property is set to 'Entity'. The 'Diagram Node' property is set to 'Node Entity (EntityFigure)'. The 'Creation Tool Entity' property is also visible.

Property	Value
Domain meta information	Entity
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Node	Node Entity (EntityFigure)
Tool	Creation Tool Entity

Debemos revisar las propiedades de cada *Node Mapping*, para asegurar que se vinculen los elementos correctos.

# Ajuste del mapping (II)

The screenshot shows the GMF Properties view for a Node Mapping. The 'Feature Label Mapping' property is set to 'false'. The 'Features to edit' property is expanded, showing two entries: 'Entity.name:EString' and 'Entity.name:EString', both of which are highlighted with a blue selection bar. The 'Diagram Label' property is set to 'Diagram Label EntityName', and the 'Read Only' property is set to 'false'. The 'Visual representation' section shows 'Edit Method' and 'View Method' set to 'MESSAGE\_FORMAT'.

Property	Value
Domain meta information	
Features to display	Entity.name:EString
Features to edit	Entity.name:EString
Misc	
Diagram Label	Diagram Label EntityName
Read Only	false
Visual representation	
Edit Method	MESSAGE_FORMAT
Editor Pattern	
Edit Pattern	
View Method	MESSAGE_FORMAT
View Pattern	

Debemos incluir los *Feature Label Mapping* que necesitemos y verificar que se vinculen los elementos correctos.

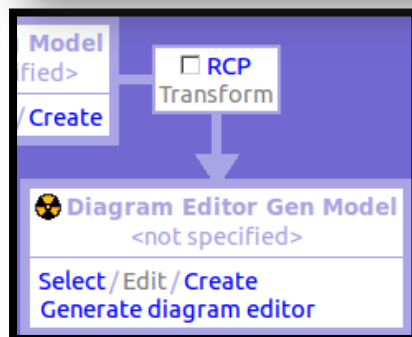
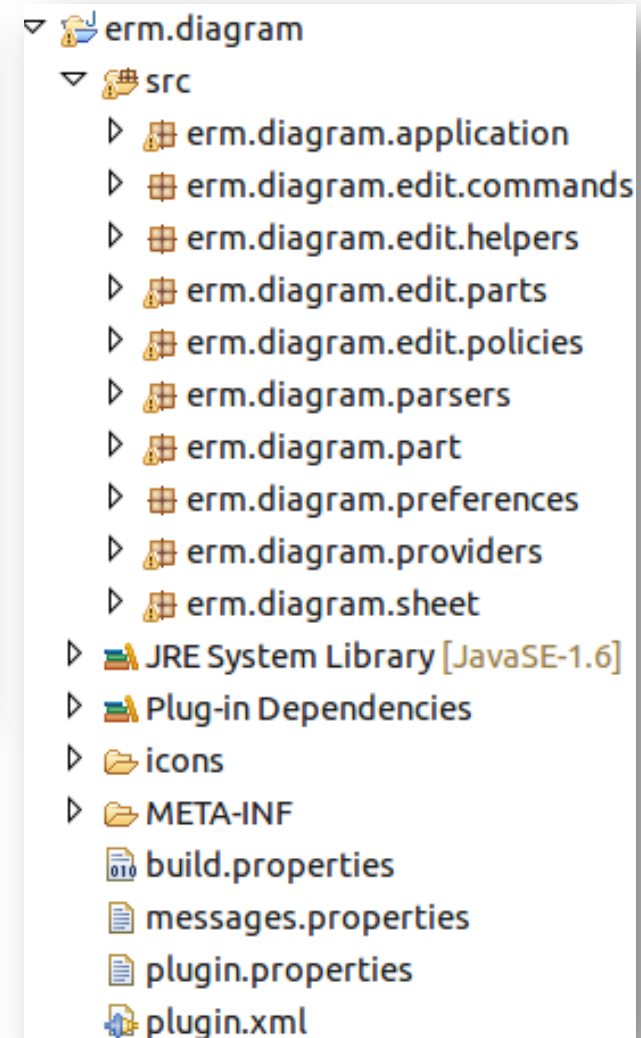
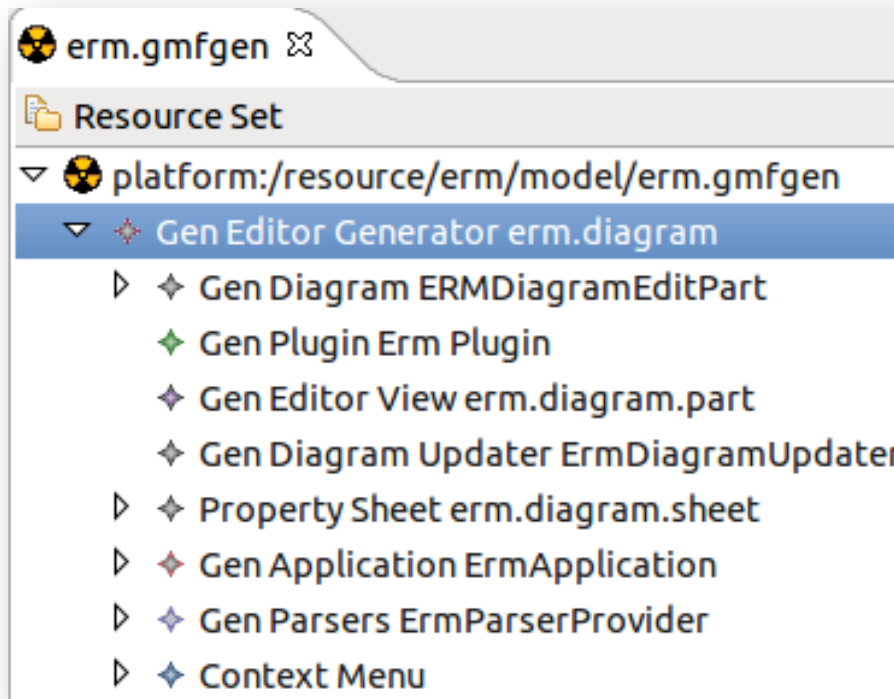
# Ajuste del mapping (III)

The screenshot shows the GMF Properties window for a Link Mapping. The selected element is 'Link Mapping <{Entity.entityRelationships:Relationship}/EntityEntityRelationships>'. The 'Target Feature' property is set to 'Entity.entityRelationships:Relationship'. The 'Diagram Link' property is set to 'Connection EntityEntityRelationships' and 'Creation Tool Entity Relationships'.

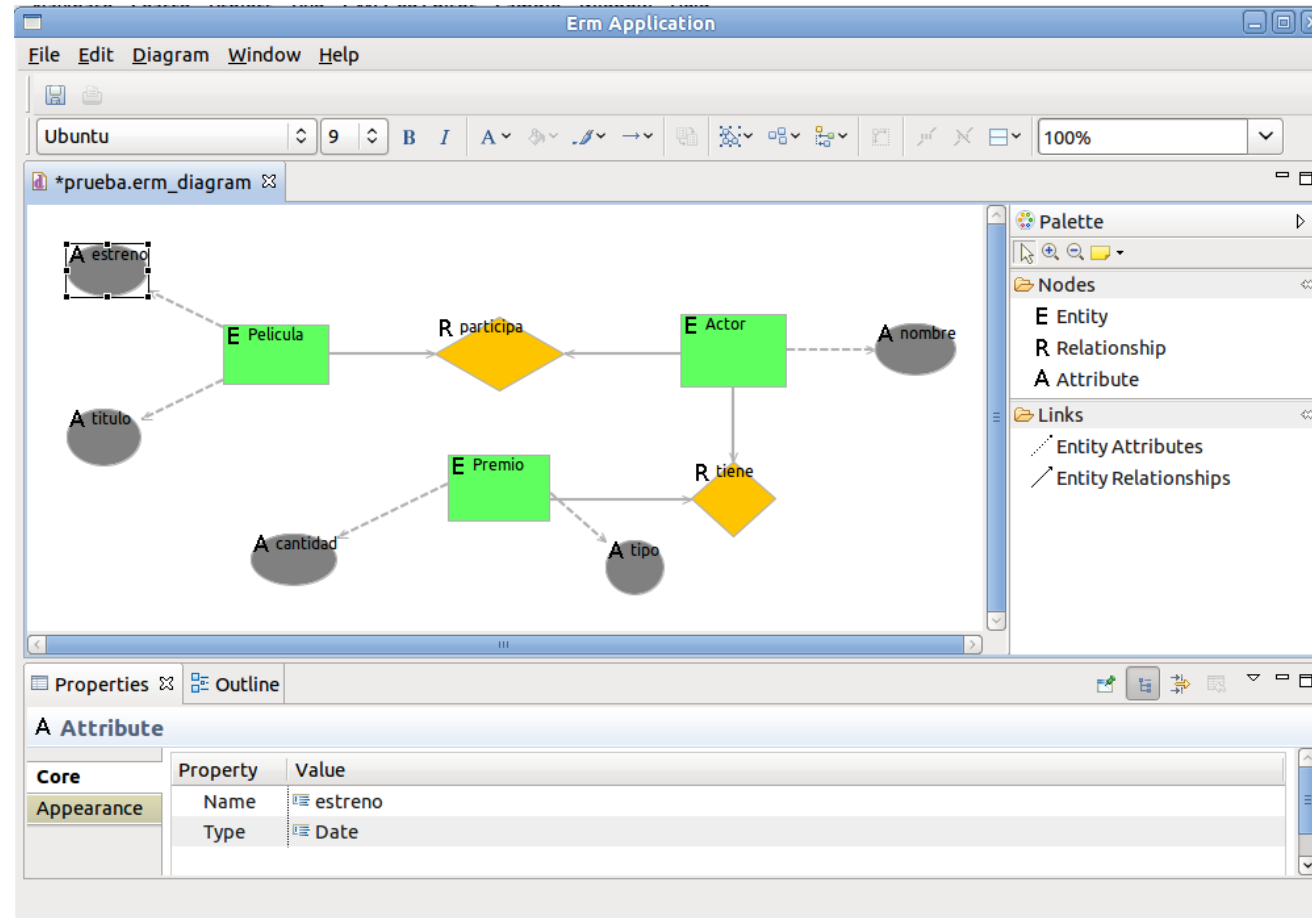
Property	Value
Domain meta information	
Containment Feature	
Element	
Source Feature	
Target Feature	Entity.entityRelationships:Relationship
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Link	Connection EntityEntityRelationships Creation Tool Entity Relationships
Tool	

Debemos revisar las propiedades de cada *Link Mapping*, para asegurar que se vinculen los elementos correctos.

# Derivar modelo de generación de código GMF y generar plug-in



# Test del editor visual



[Project *Diagram*] Run as → Eclipse Application

DESARROLLO DE EDITORES GRÁFICOS CON GMF



# RESUMEN



# ¿Qué hemos aprendido hoy?

- Framework GMF para el desarrollo de editores visuales de modelos.
- Desarrollo basado en la sucesiva transformación de modelos.
- Requiere un desarrollo muy ordenado, ya que es común cometer errores.
- No siempre debemos confiar en los modelos intermedios generados automáticamente por Eclipse.
- Existe una alternativa más cómoda: EuGENia



Procesadores de Lenguajes 2

# Desarrollo de editores gráficos con GMF

**Iván Ruiz Rube**

ivan.ruiz@uca.es