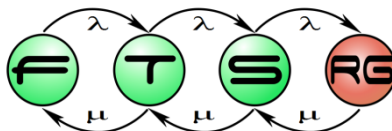
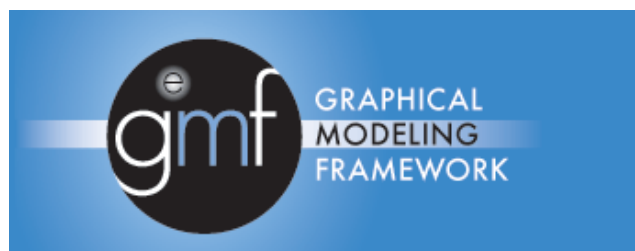


Eclipse Graphical Modeling Framework (GMF)



- Graphical Modeling Framework
- Cél
 - Domain-specifikus nyelvek grafikus szerkesztése
 - Modellalapon, kevés kódolással (kódgenerálás segítségével)
 - Egységes keretrendszerben (Eclipse alapokon)
 - Funkciók gyors, inkrementális megvalósítása
- Fejlesztők: IBM, Borland

Tartalom

- GMF bevezető
 - GMF runtime és kódgenerálás
- GMF fejlesztés lépései
 - Konkrét szintaxis specifikációja
 - EMF metamodel
 - Grafikus modell
 - Tooling modell
 - Mapping modell
 - Generátor testreszabása
 - Kódgenerálás és tesztelés
 - Extrák, bővítések
 - GMF runtime APIs
 - OCL kényszerek
- Összefoglaló

GMF áttekintés

- A GMF két fő részből áll:
 - Runtime
 - Kapocs az EMF és a GEF között
 - Modell és diagram szintű szolgáltatások
 - Bővíthető
 - Generálás (tooling)
 - Modellvezérelt
 - Diagram megjelenelés (graphical), eszközök (tooling), leképezés a modellre (mapping)
 - Generált kód a runtime-ot célozza meg

GMF Runtime

- Grafikus editorok írását támogató keretrendszer
 - Újrahasznosítható komponensek
 - Standardizált diagram metamodel
 - A logikai és diagram metamodellek különválnak
 - Bővíthető grafikus szerkesztőket készíthetünk az Eclipse Workbenchbe integrálva

GMF Runtime

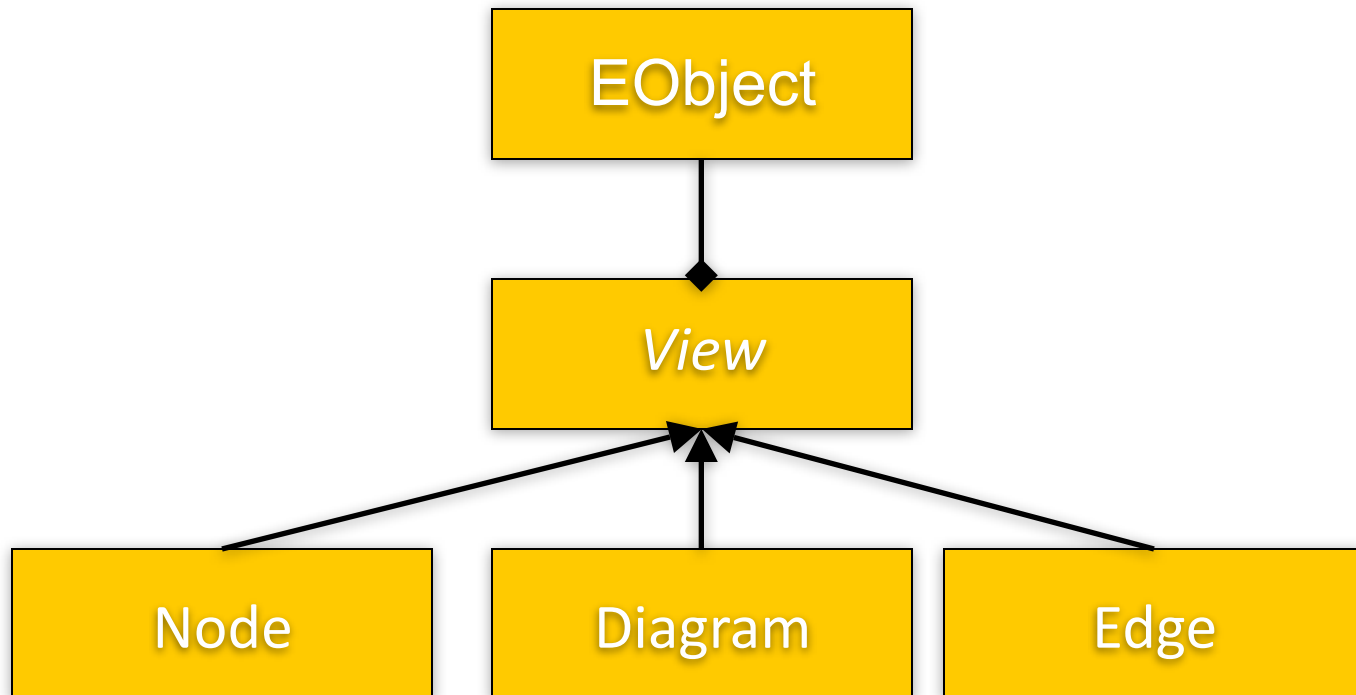
- Grafikus editorok írását támogató keretrendszer
 - Command-jellegű működés az EMF és GEF együttes vezérléséhez
 - Perzisztencia az EMF runtime segítségével
 - MVC szerkesztés GEF alapokon
 - További felhasznált technológiák:
 - EMF Model Transaction
 - EMF Validation Framework
 - MDT Object Constraint Language (OCL)
 - Apache Batik (SVG)

Notation metamodel

- A Runtime fontos része
- Nézeti infók tárolása:
 - Szín, betűtípus stb.
 - Node esetében: pozíció, méret stb.
 - Edge esetében: töréspontok, dekorációk stb.
- Nézeti modell a domain modell fölé
 - GMF nyújtja, nincs szükség a domain modell megváltoztatására!

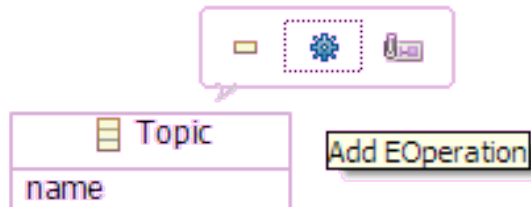
Notation metamodel

- Fő osztálya a View
 - Wrappeli a domain modell objektumot
 - Domain modell objektum elérése: get/setElement()

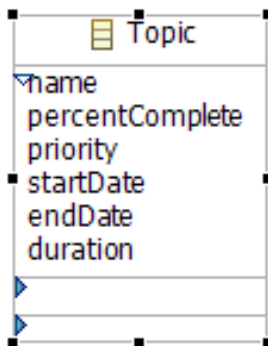


Standard komponensek

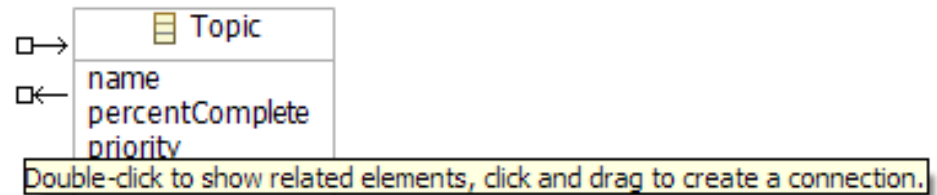
Popup Action Bar:



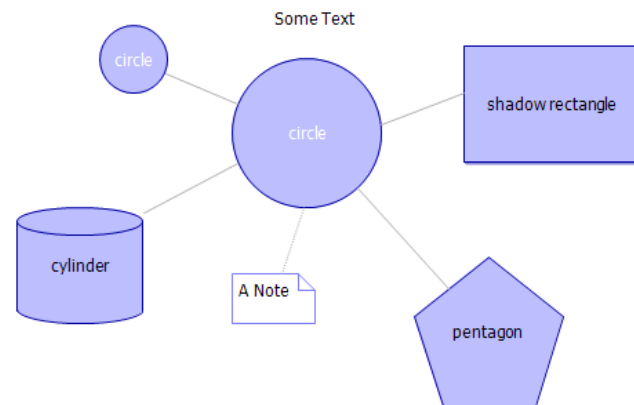
Compartment (collapsible):



Connection Handle:



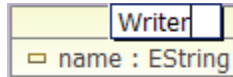
Geometrical Shape:



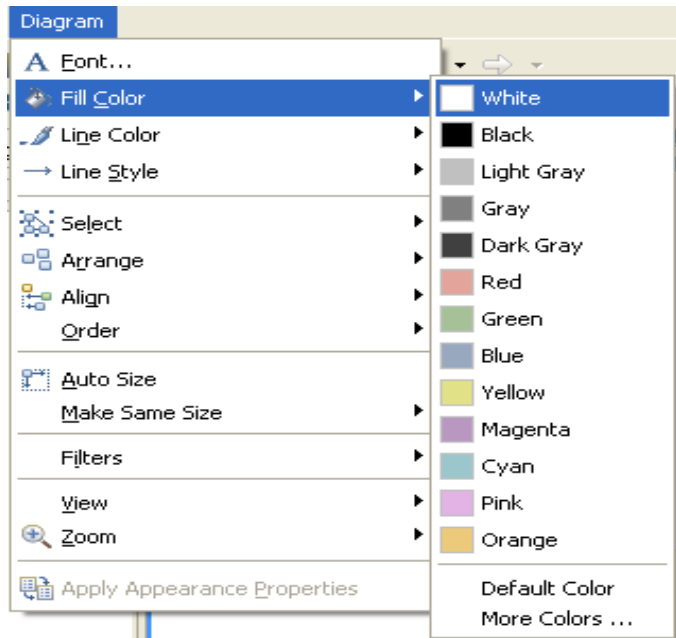
Standard komponensek

Action:

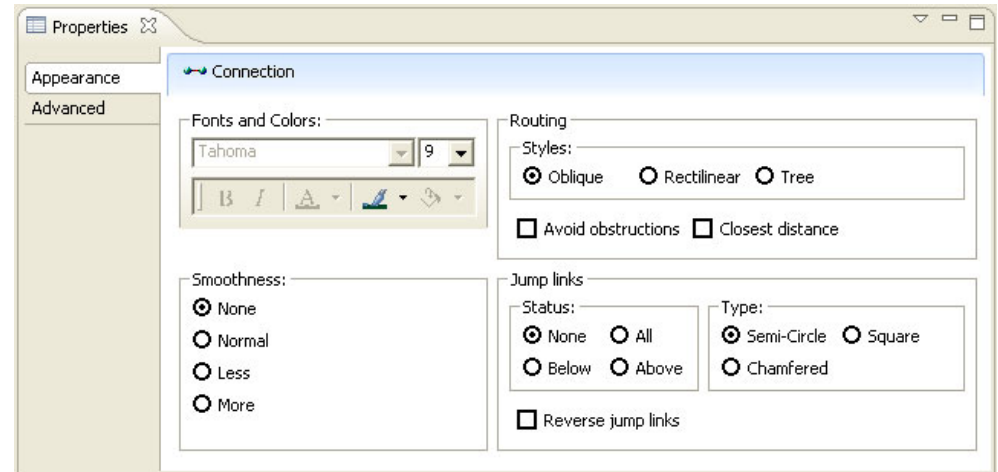
Direct Edit:



Toolbar:

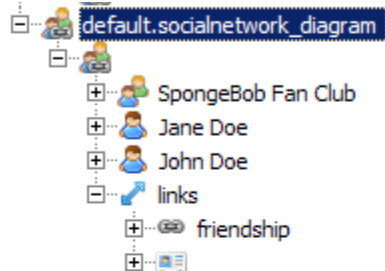


Properties View:

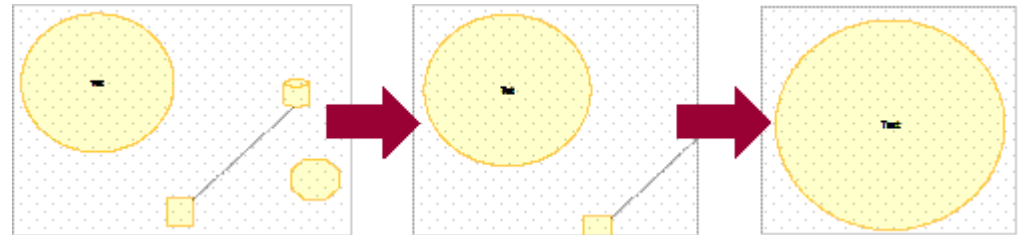


Standard komponensek

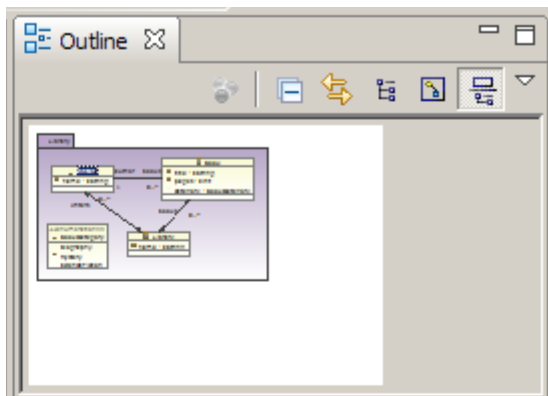
Navigátor a modellen belül:



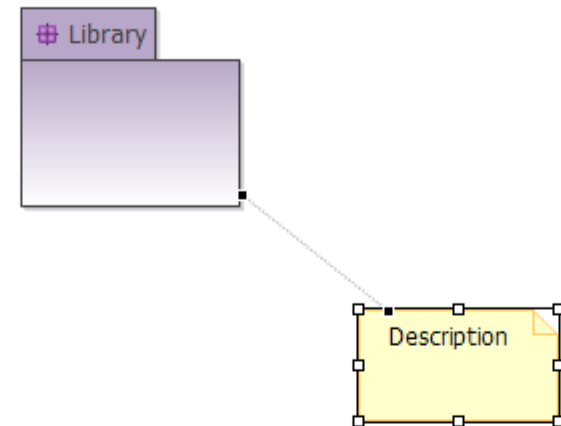
Animált zoom:



Áttekintő nézet:

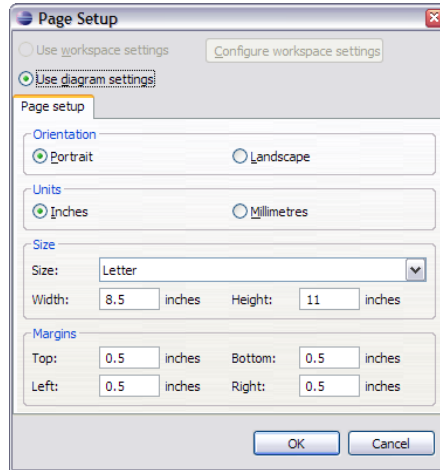


Note Attachment:



Standard komponensek

Oldalbeállítás:



Nyomtató
előnézet:

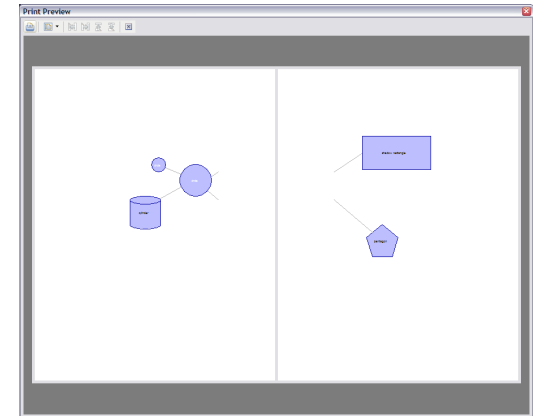


Diagram export képfájlba:

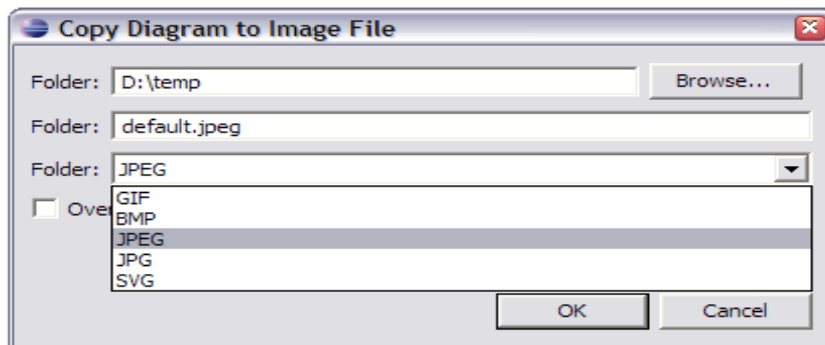
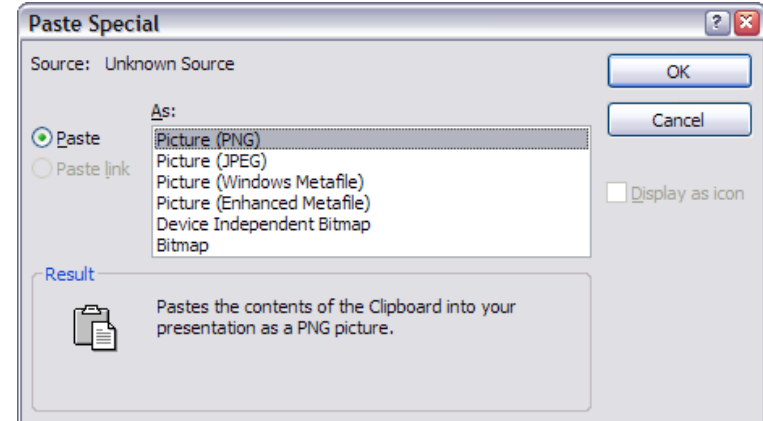


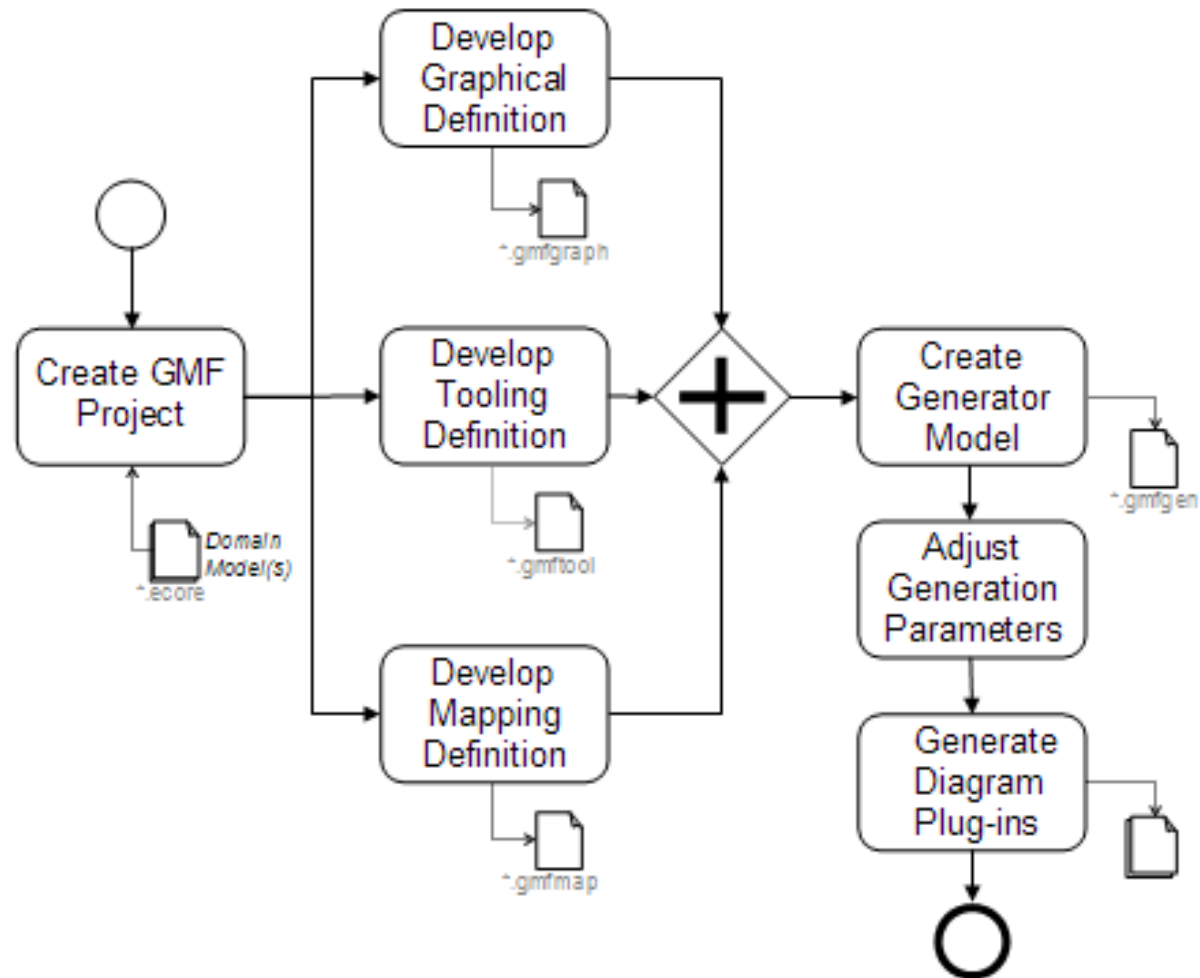
Diagram másolás a vágólapra:



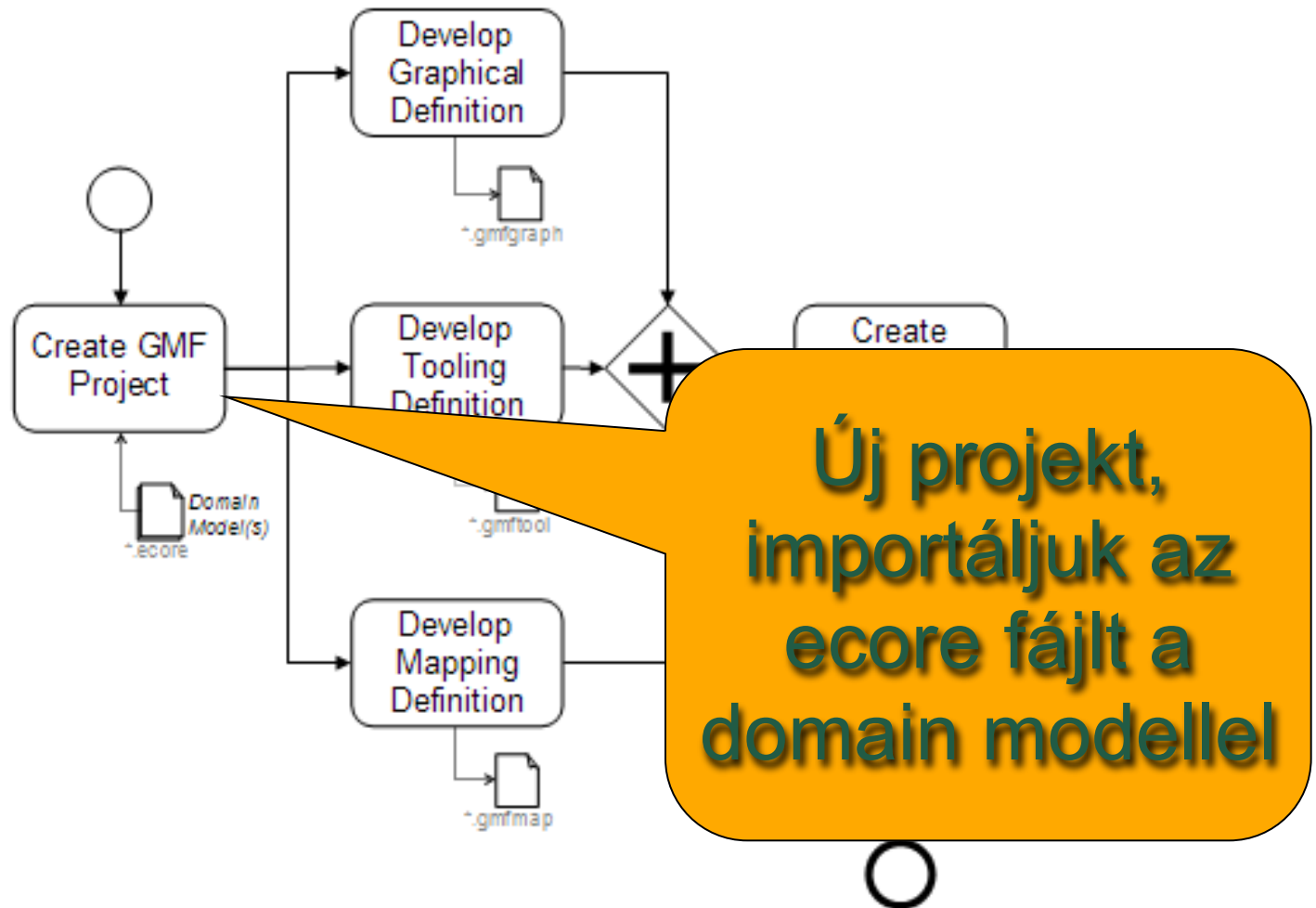
GMF Tooling

- Célja:
 - Modellvezérelt szerkesztő generálás
 - A diagram és logikai modellek elkülönülnek
 - Testreszabott domain-specifikus szerkesztők gyors „összedobása”
 - Az eredmény legyen bővíthető és rugalmasan átalakítható
- Runtime-hoz hasonlóan sok mindenre képes

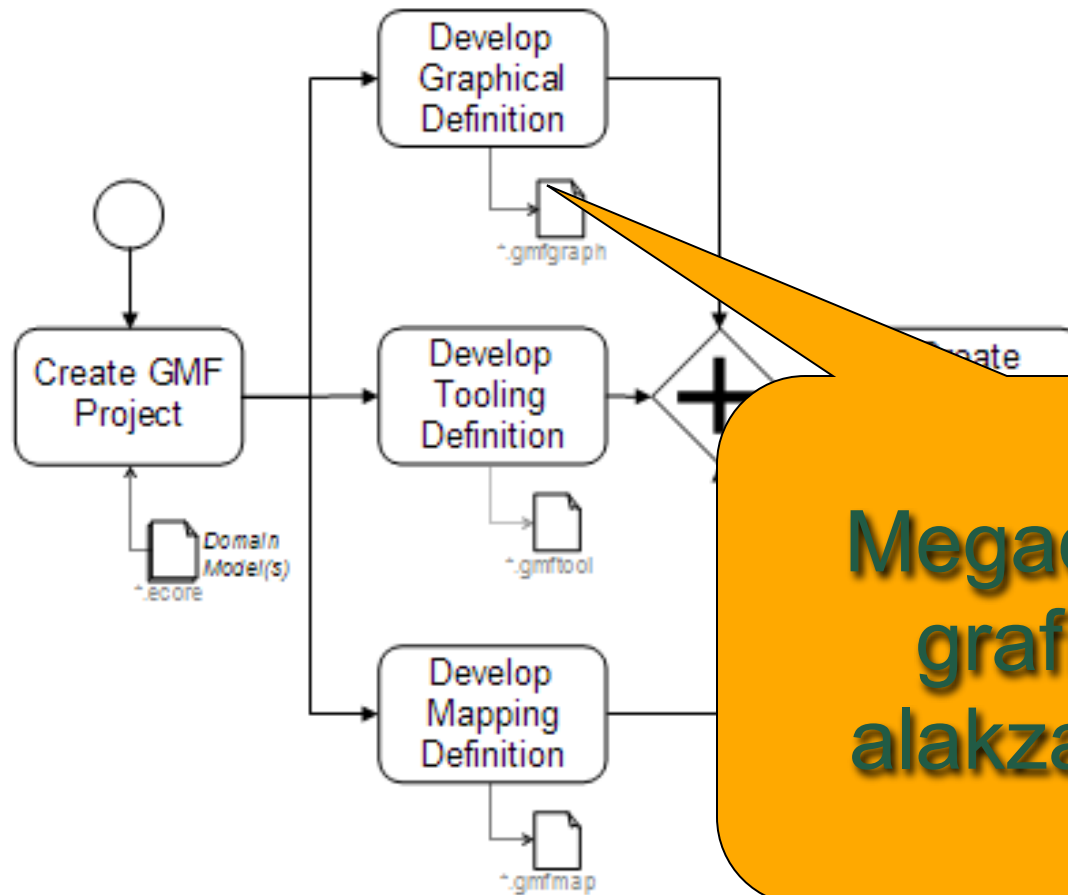
GMF szerkesztő generálása



GMF szerkesztő generálása

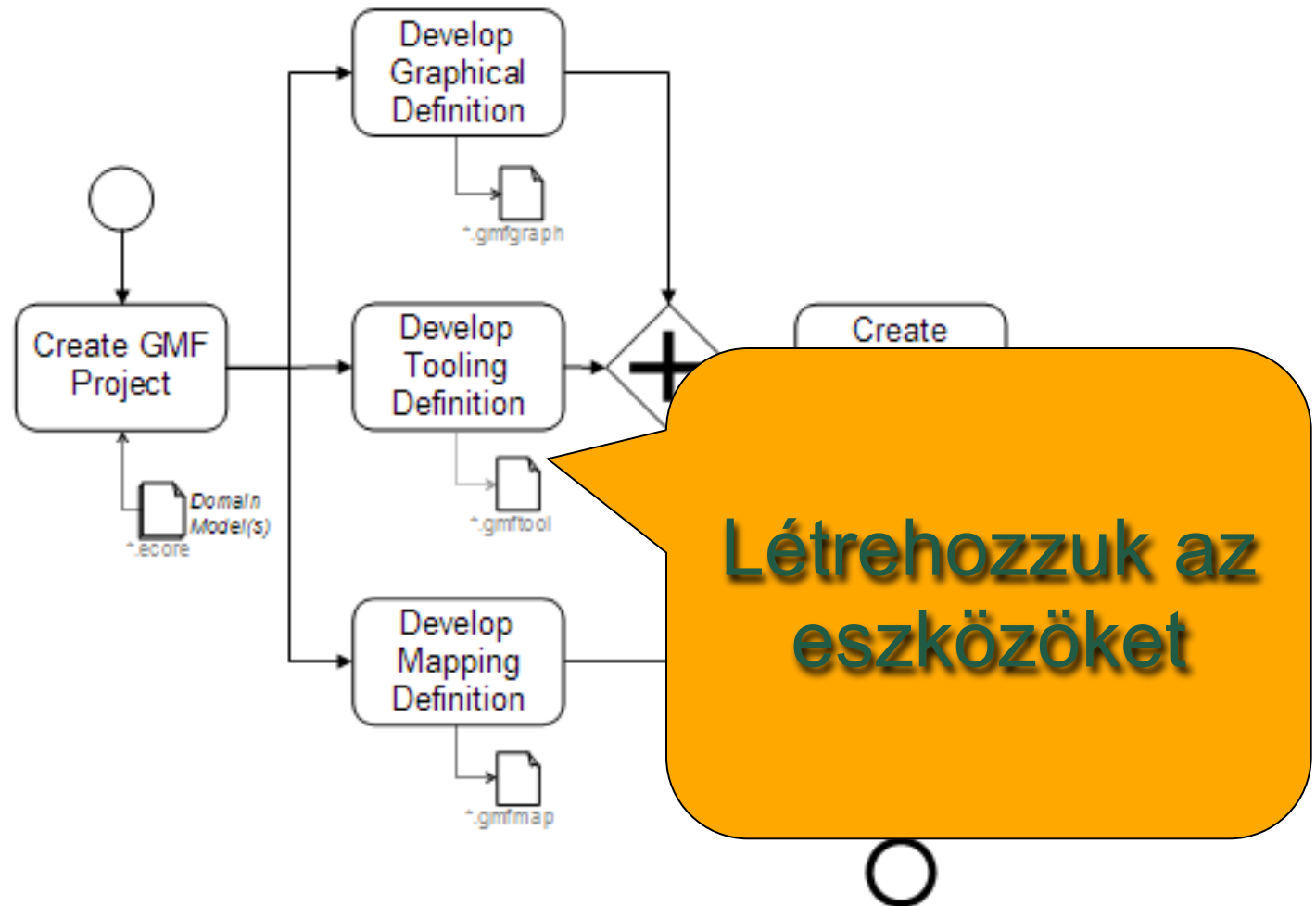


GMF szerkesztő generálása

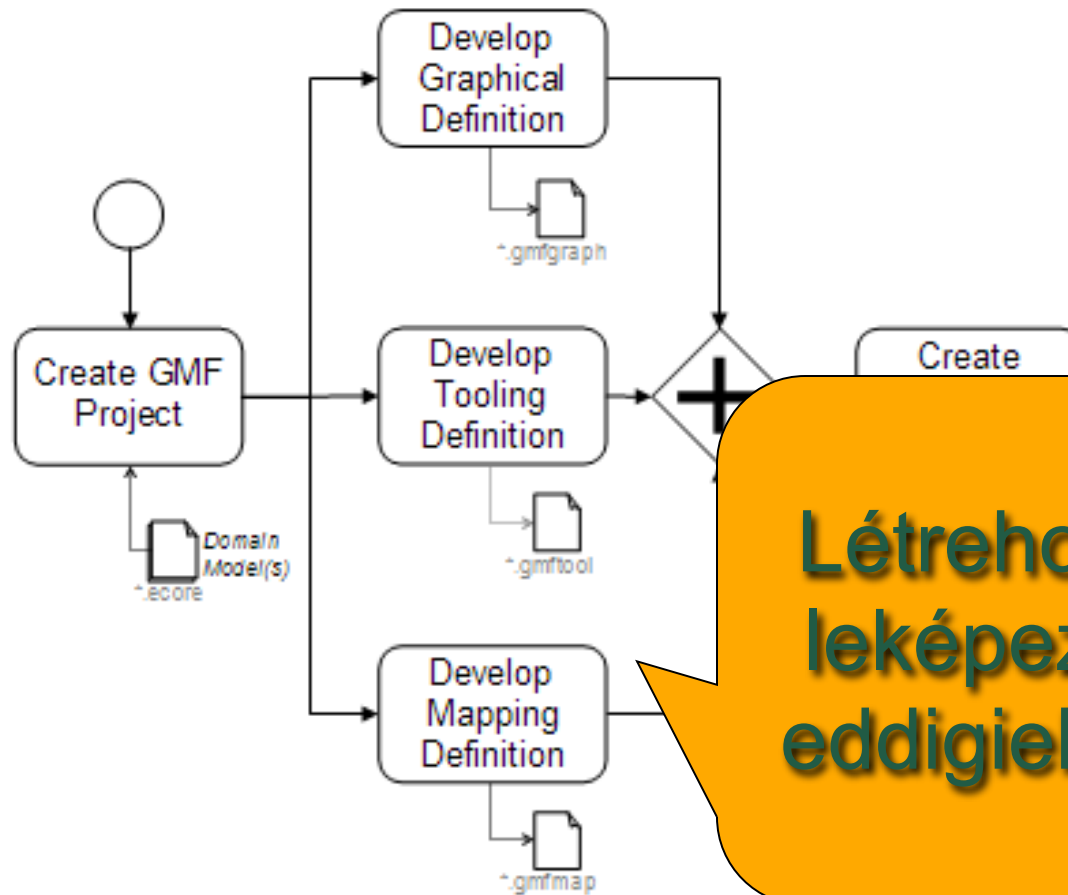


**Megadjuk a
grafikus
alakzatokat**

GMF szerkesztő generálása



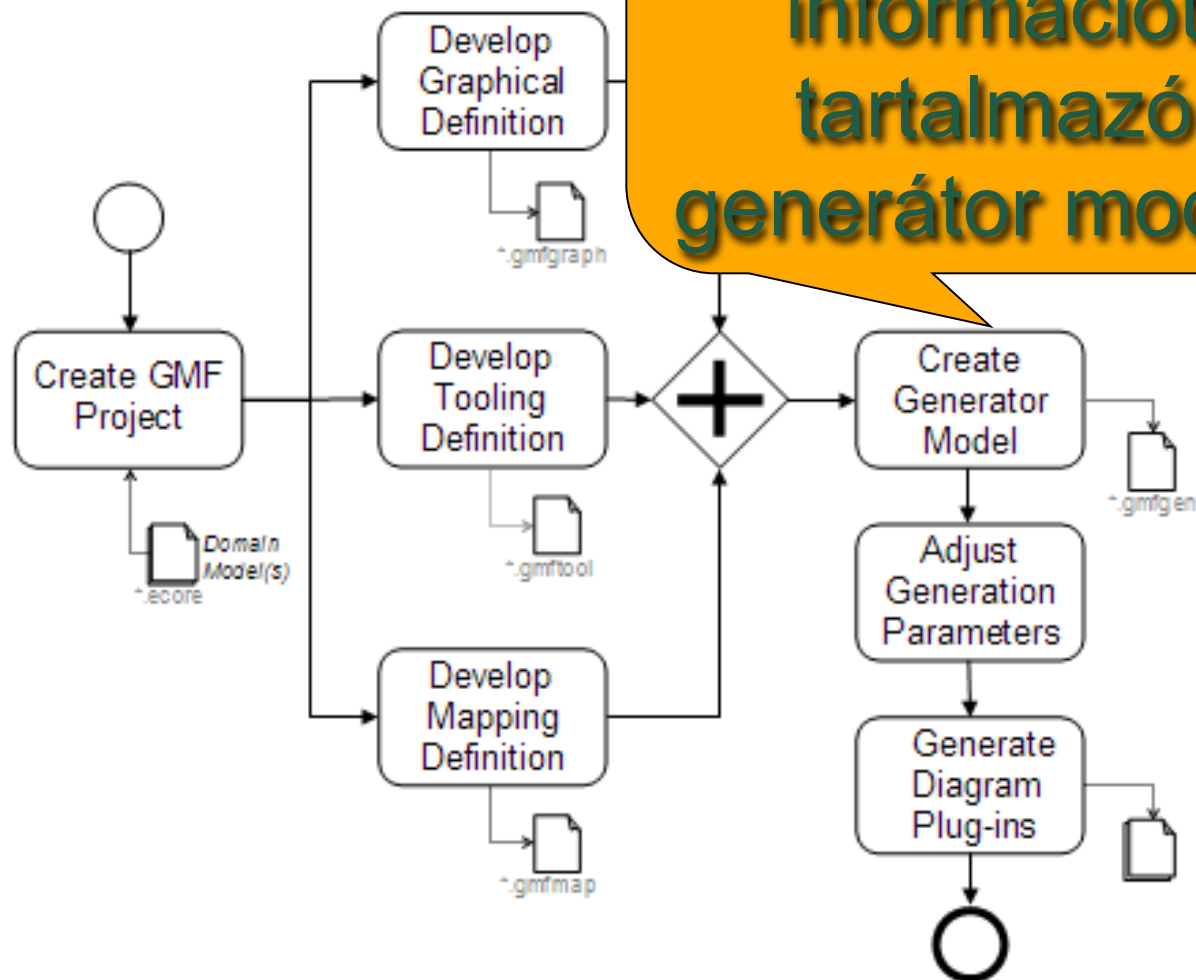
GMF szerkesztő generálása



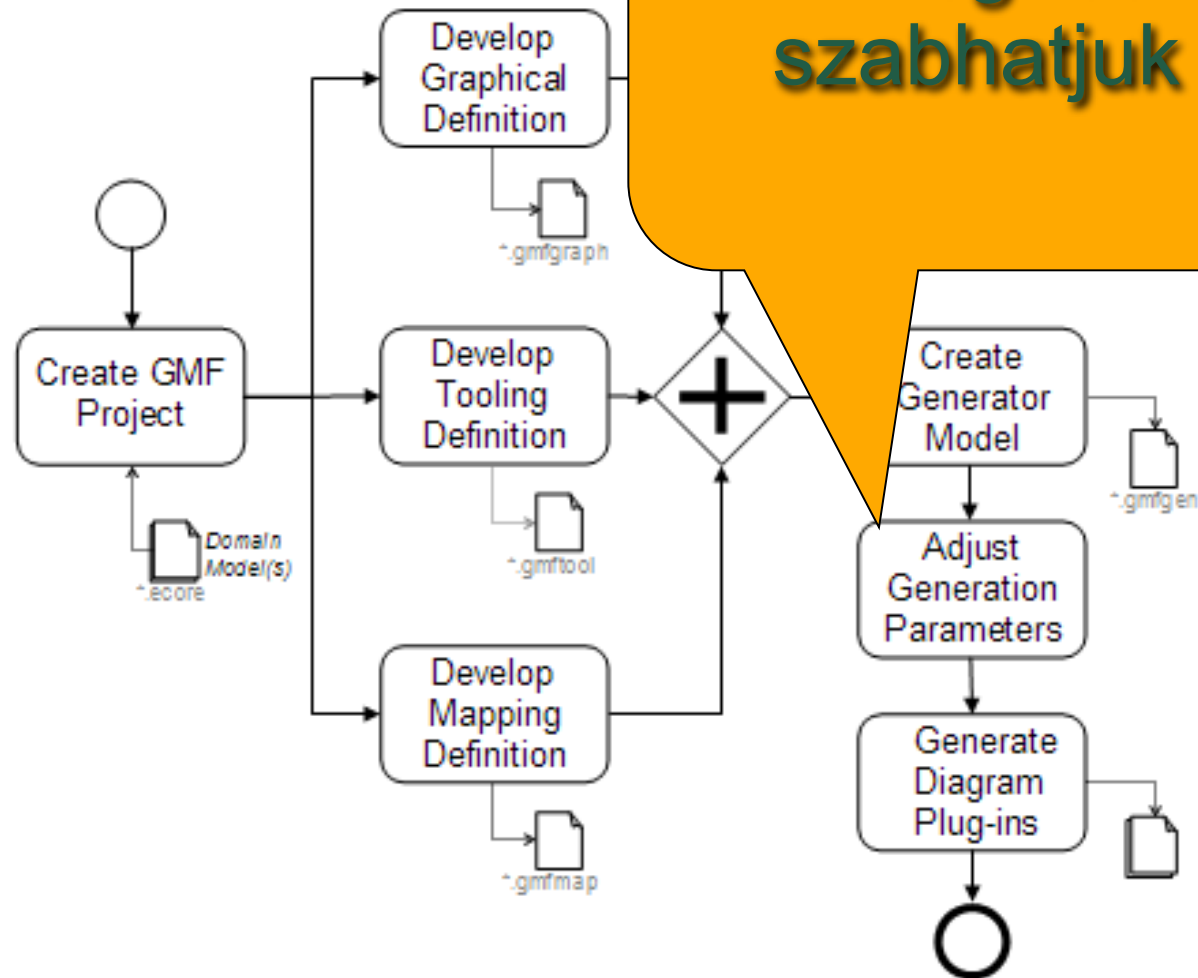
**Létrehozzuk a
leképezést az
eddigiek között**

GMF szerkesztő generálása

**Legenerálódik a
közös, minden
információt
tartalmazó
generátor modell**



GMF szerkesztő generálása

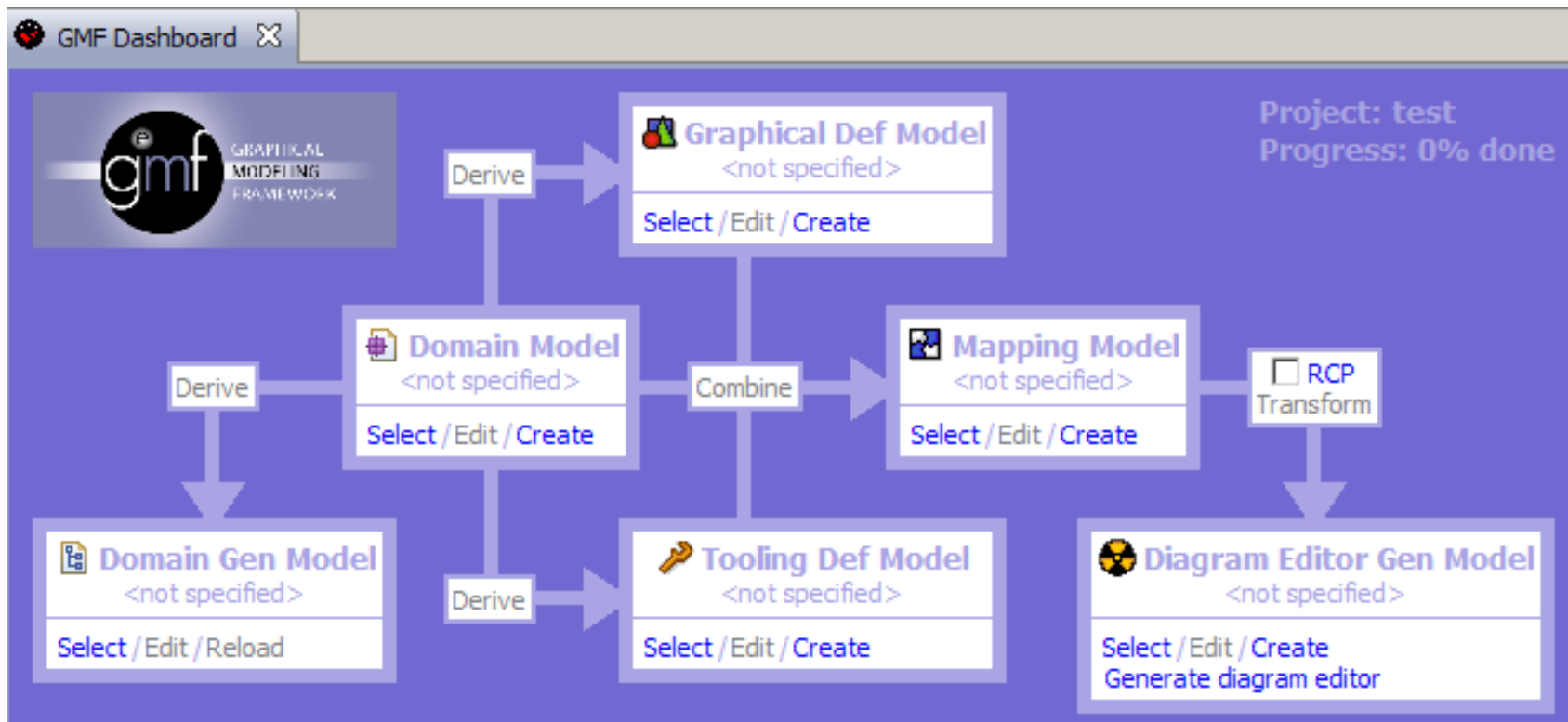


GMF szerkesztő generálása



GMF Dashboard View

- Grafikusan követhető a fejlesztés menete



GMF és EMF projektek

- A diagramszerkesztő működéséhez szükség van az EMF projekt genmodeljéből generált projektekre:
 - Model
 - Edit
- Konvencionális projektszerkezet:
 - library – EMF projekt
 - model – Modellfájlok könyvtára
 - library.ecore
 - library.genmodel
 - library.gmfgraph
 - library.gmftool
 - library.gmfmap
 - library.gmfgen
 - library.diagram – Generált GMF projekt
 - library.edit – Generált EMF edit projekt



Domain Model

- A logikai metamodel
- Normál Ecore (EMF) modell
- Megadása az EMF-ben megszokott módon
- Pl. a GMF grafikus Ecore editorával

Graphical Definition Model

- Diagram metamodel (GMFGraph)
- Cél
 - A használni kívánt grafikus elemek megadása
- A Figure-ök programozása helyett, magas szintű modellben
- Tree editorral...
- Van hozzá automatikus támogatás (varázsló)
 - A domain modelből következtet, testreszabhatóan

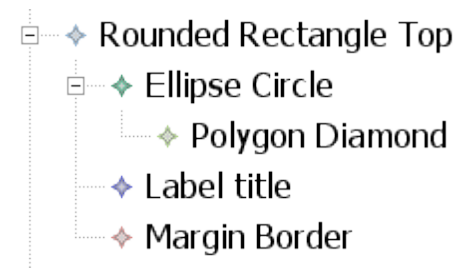
Graphical Definition Model

- Java kódolás helyett modellezés

Graphical Definition Model

- Java kódolás helyett modellezés

Előre megadott elemekből
összeállított figure



Graphical Definition Model

- Java kódolás helyett modellezés

Előre megadott elemekből
összeállított figure

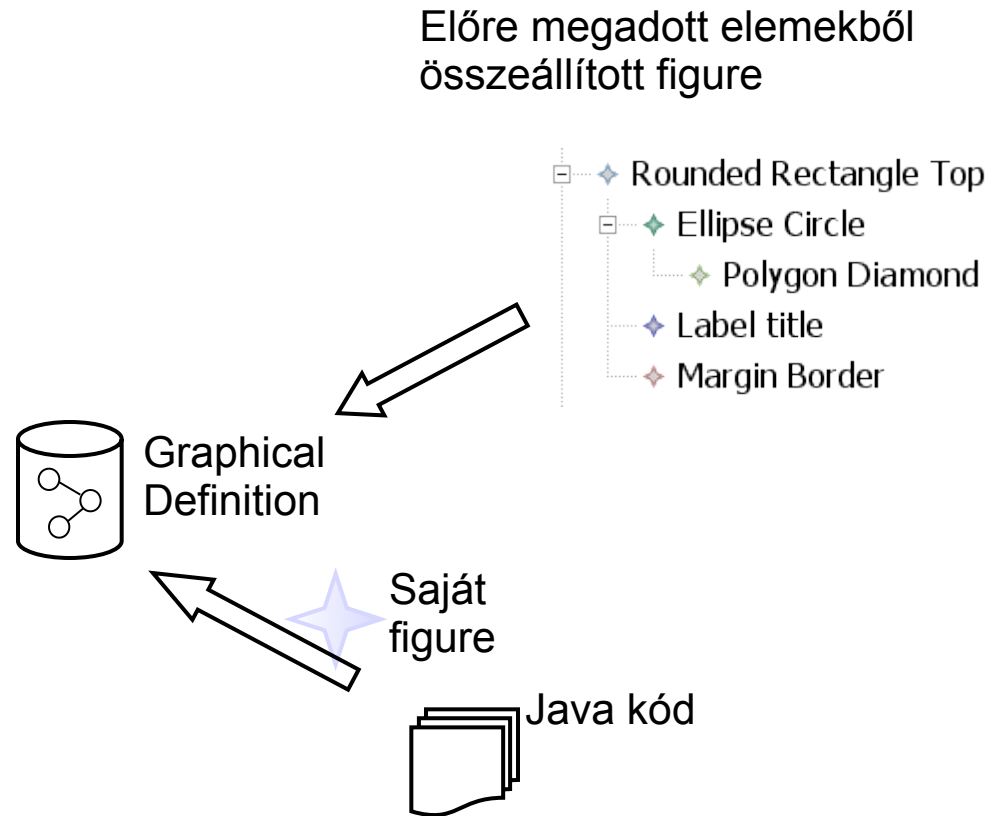


 Saját
figure

 Java kód

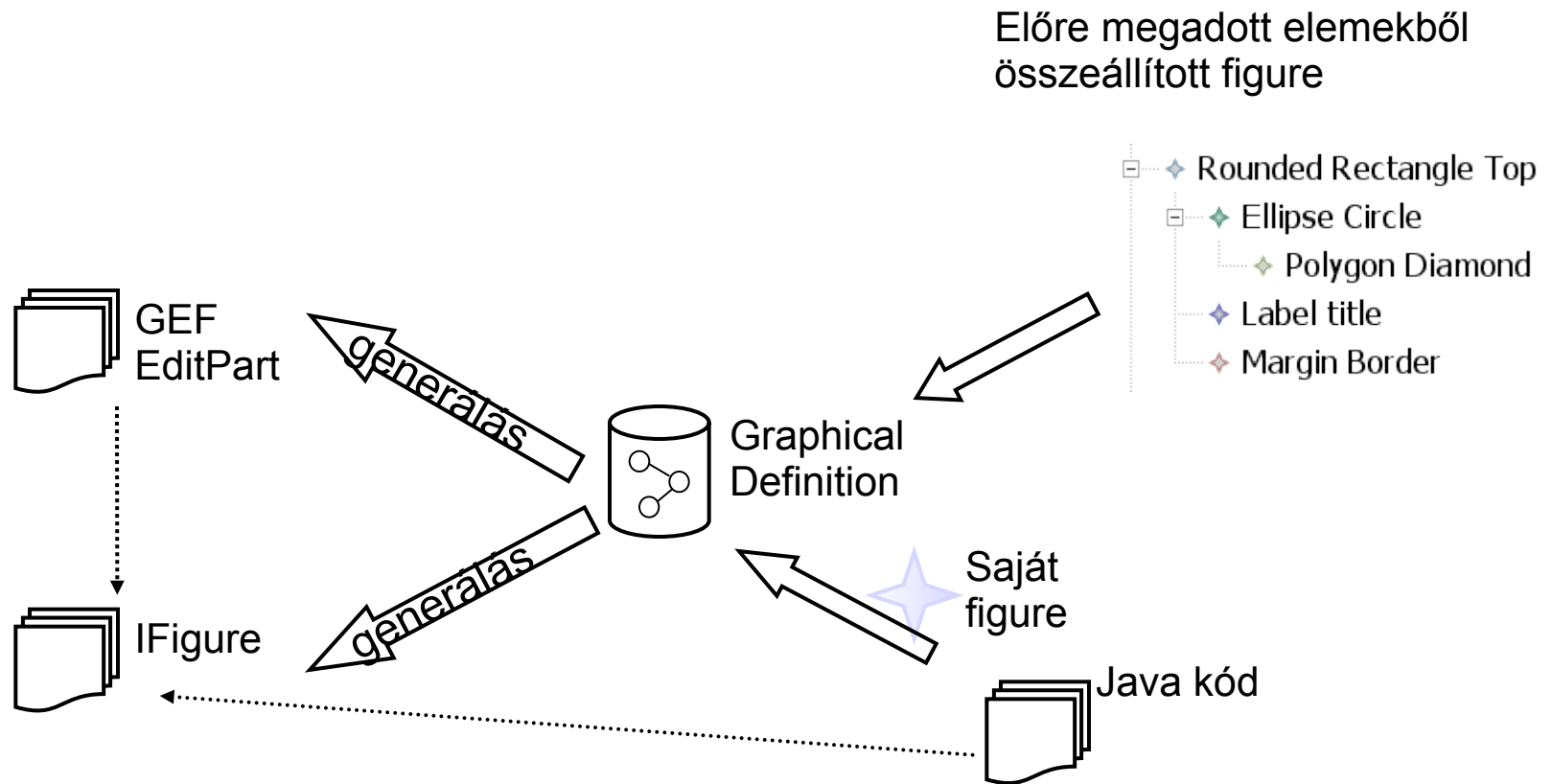
Graphical Definition Model

- Java kódolás helyett modellezés



Graphical Definition Model

- Java kódolás helyett modellezés



Graphical Definition Model

- Platformfüggetlen metamodel
- Felépítése:
 - Elemkönyvtárak (Figure Galleries)
 - Rajzelemek hierarchiája (Figures)
 - Csomópontok (Nodes)
 - Kapcsolatok (Links)
 - Tartalmazók (Compartments)
 - Címkék (Diagram Labels)
- A wizard generál egy alap változatot bármely domain metamodelhez

Figure Gallery

- Figure descriptorok
- Rajzelemek hierarchikusan
 - Label, Rectangle, Ellipse, Polygon, Polyline, Custom Figure stb.
 - Borders: Line, Margin, Compound, Custom
- Elrendezésük
 - Layouts: Flow, Border, Grid, XY, Stack, Custom
- Tulajdonságaik
 - Color, Font, Dimension, Insets
- Child Access: accessorok

Node

- Diagram csomóponttípus
- A diagramok alapeleme
- Figure descriptorra hivatkozik
- Kitöltés és körvonal tulajdonságai megadhatóak

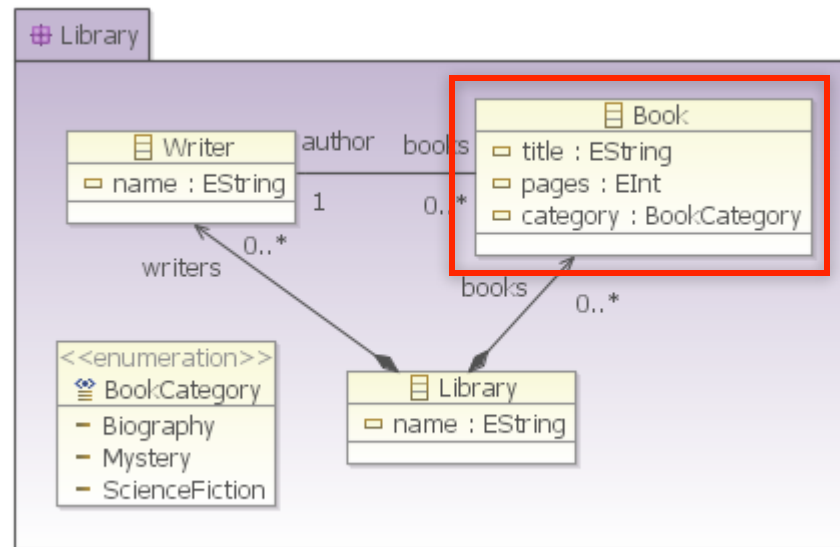
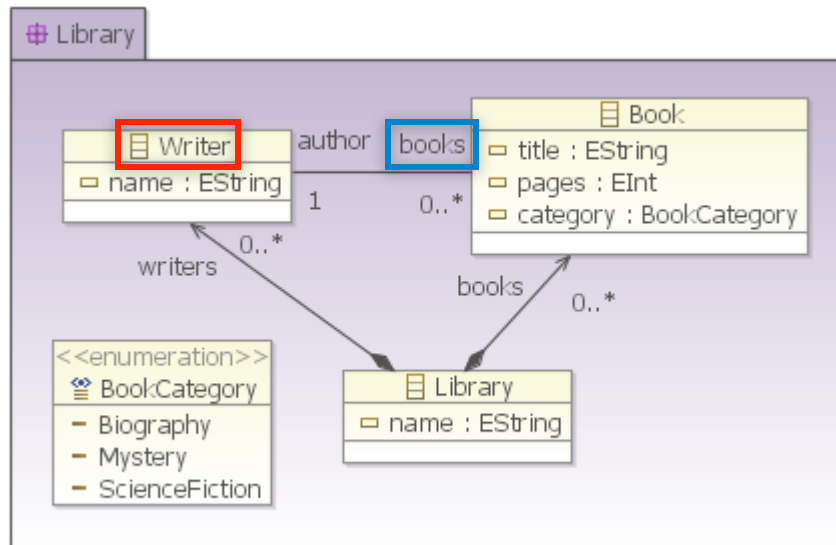


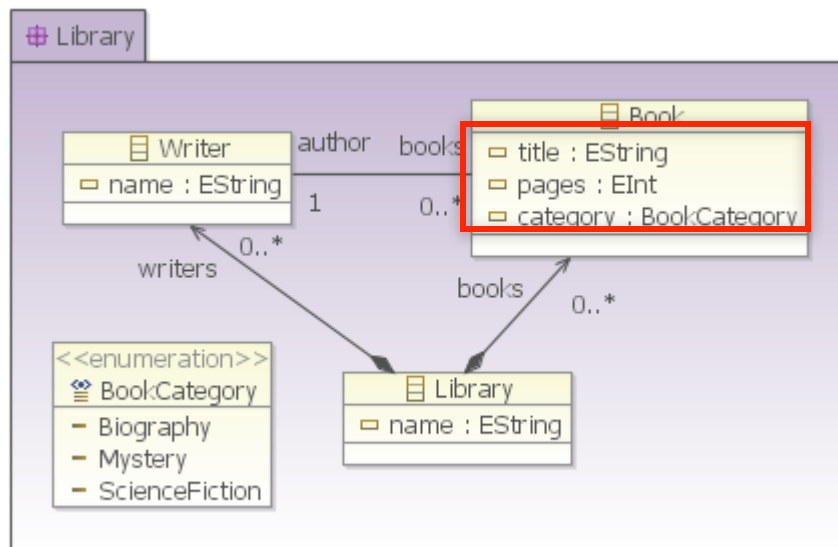
Diagram Label

- Címke a diagramon
- Két típus
 - Belső: Figure descriptor egy gyerekére (child access) hivatkozik
 - Külső: Figure descriptorra hivatkozik



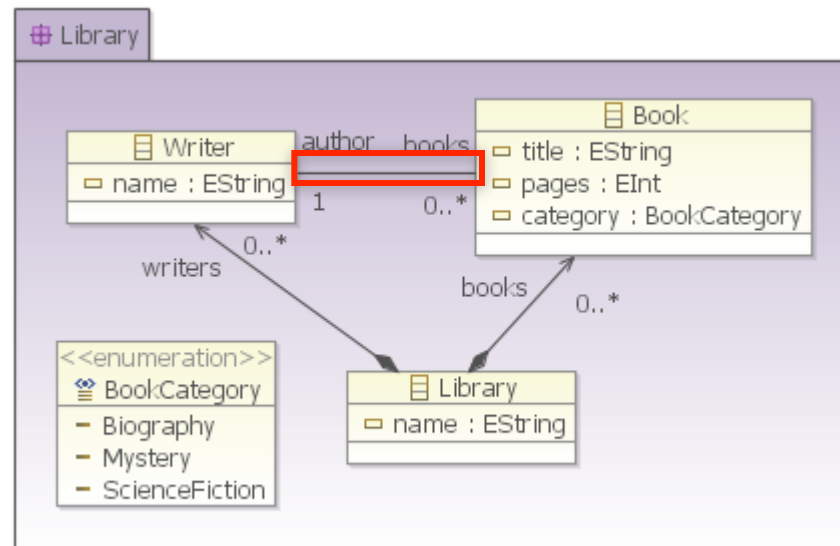
Compartment

- Tartalmazást kifejező „rekesz”
- Figure descriptor egy gyerekére (child access) hivatkozik
- Lehet összecsuksukható



Connection

- Összekötő
- Figure descriptorra hivatkozik



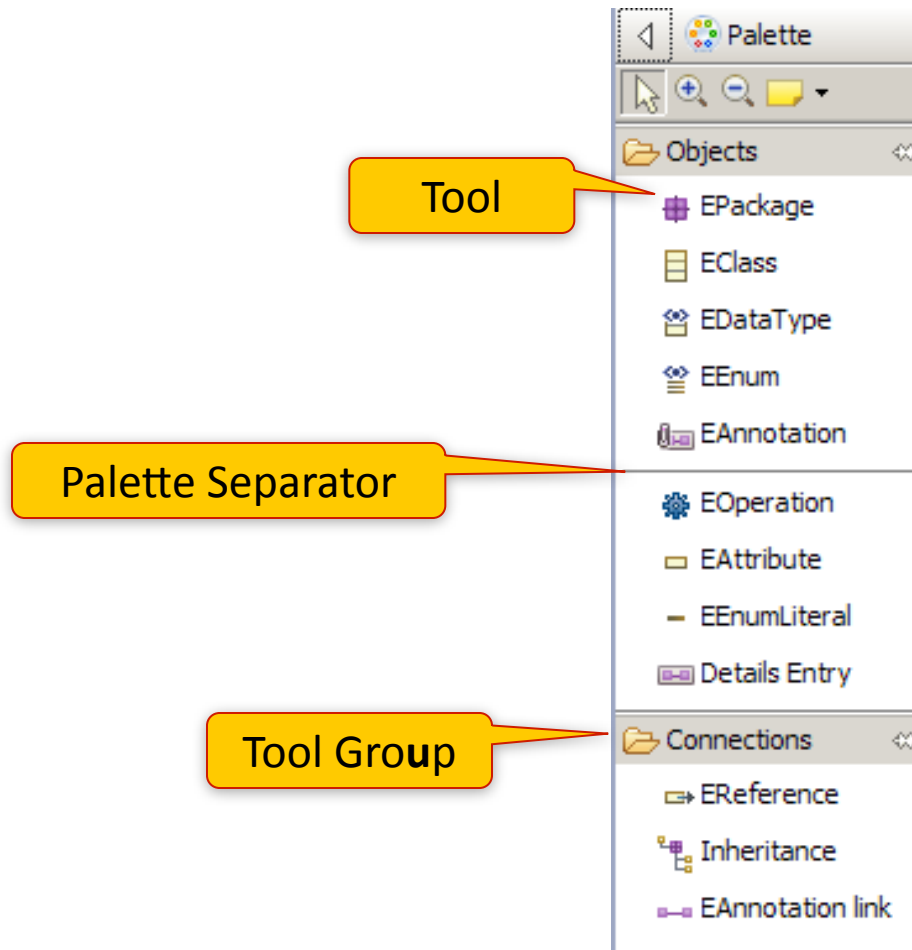
Tooling Definition Model

- Tooling metamodel (GMFTool)
- A szükséges eszköztár-elemek megadása
- Szintén varázslós támogatás
- A generálás során implementálásra is kerülnek az itt megadott parancsok

Tooling Definition Model

- A szerkesztéshez használt eszközöket definiálja:
 - Palette
 - Tool (tipikusan creation)
 - Csoportokba (tool group) rendezve
 - Menü
 - Main/Popup
 - Action
 - Toolbar
- A wizard generál egy alap változatot bármely domain metamodelhez

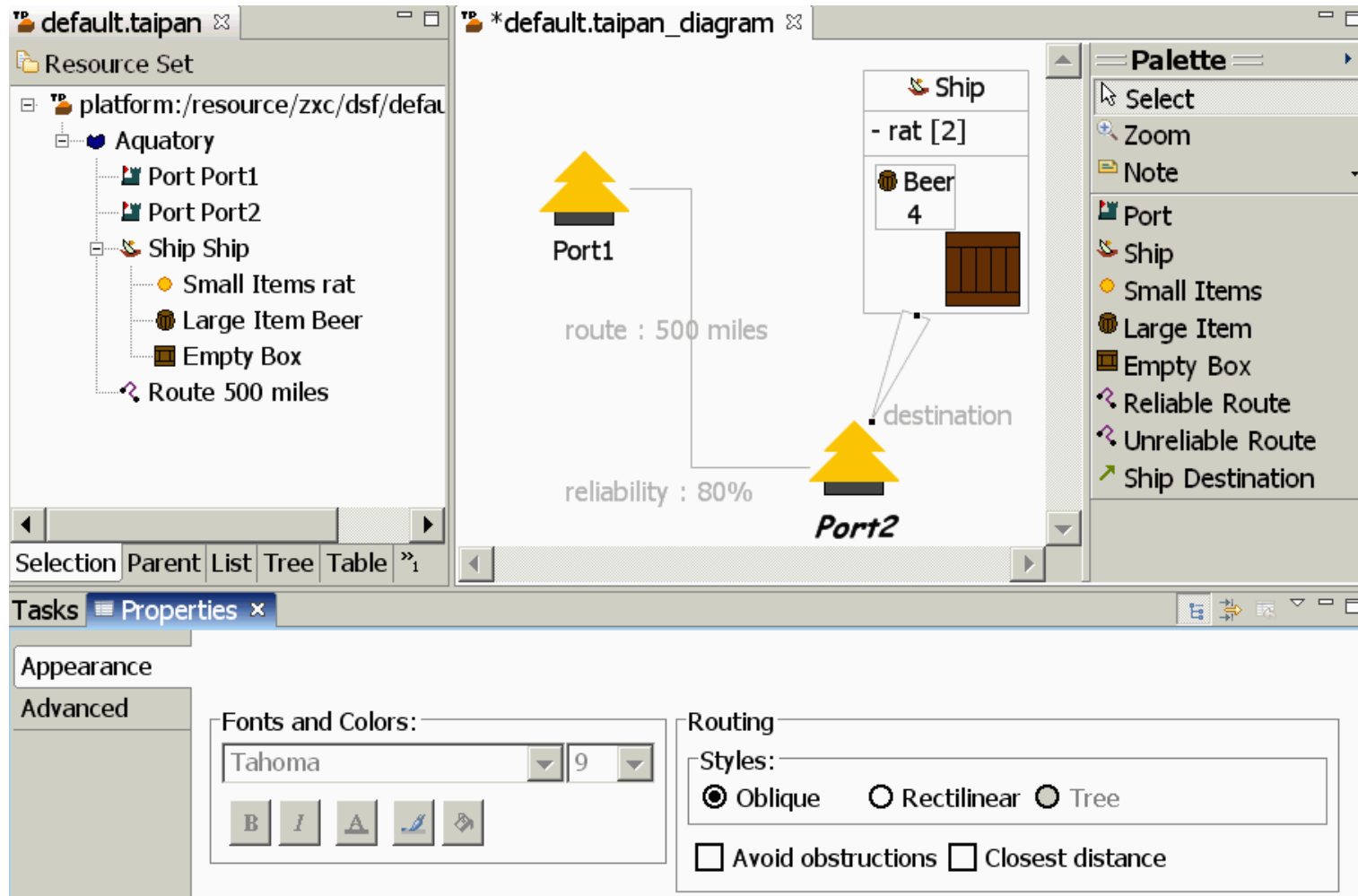
Paletta



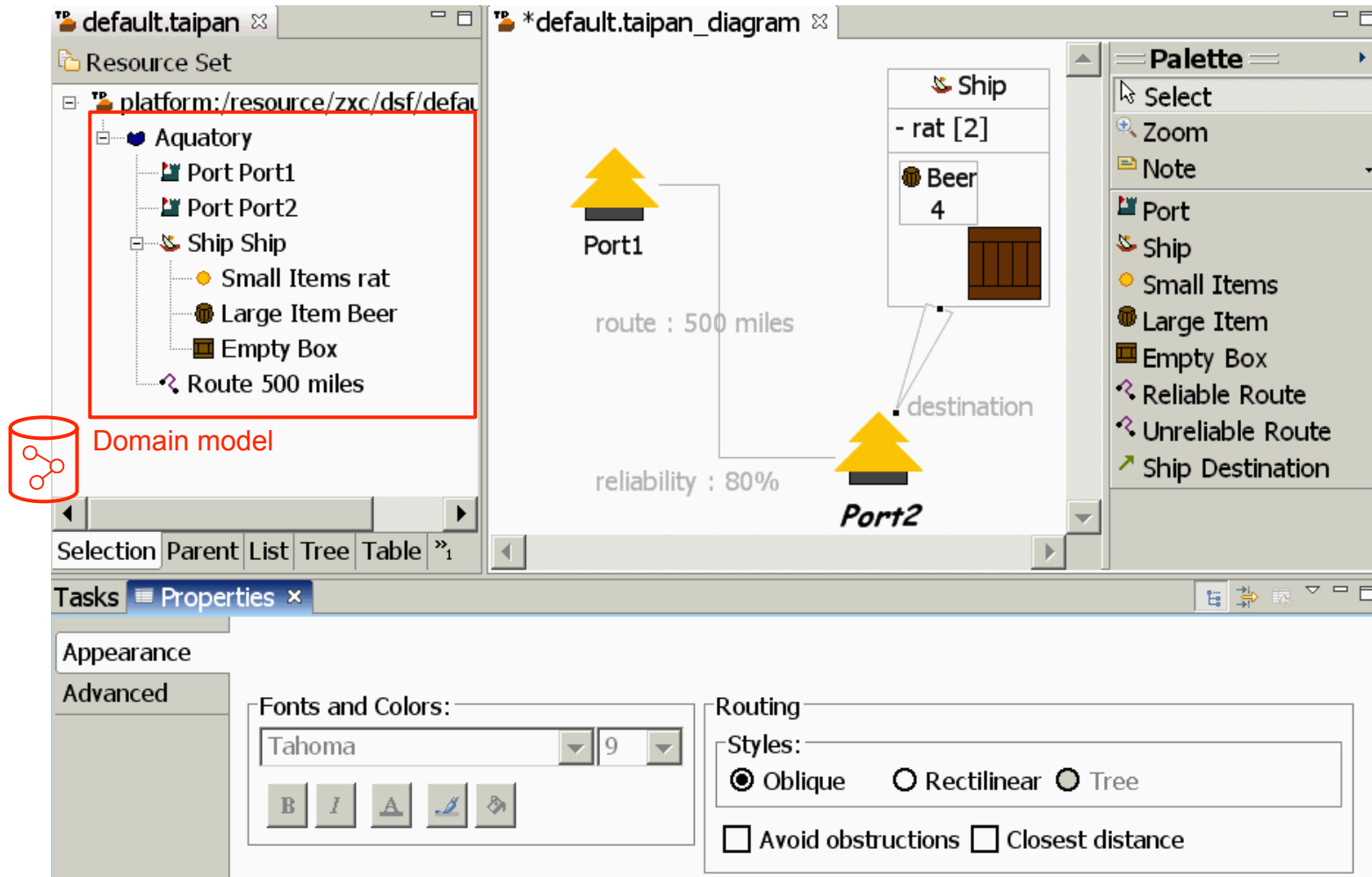
Mapping Model

- Leképezési metamodel (GMFMap)
- Az eddigi modellek összekötése
- Megfeleltetések megadása
- Domain – graphical – tooling

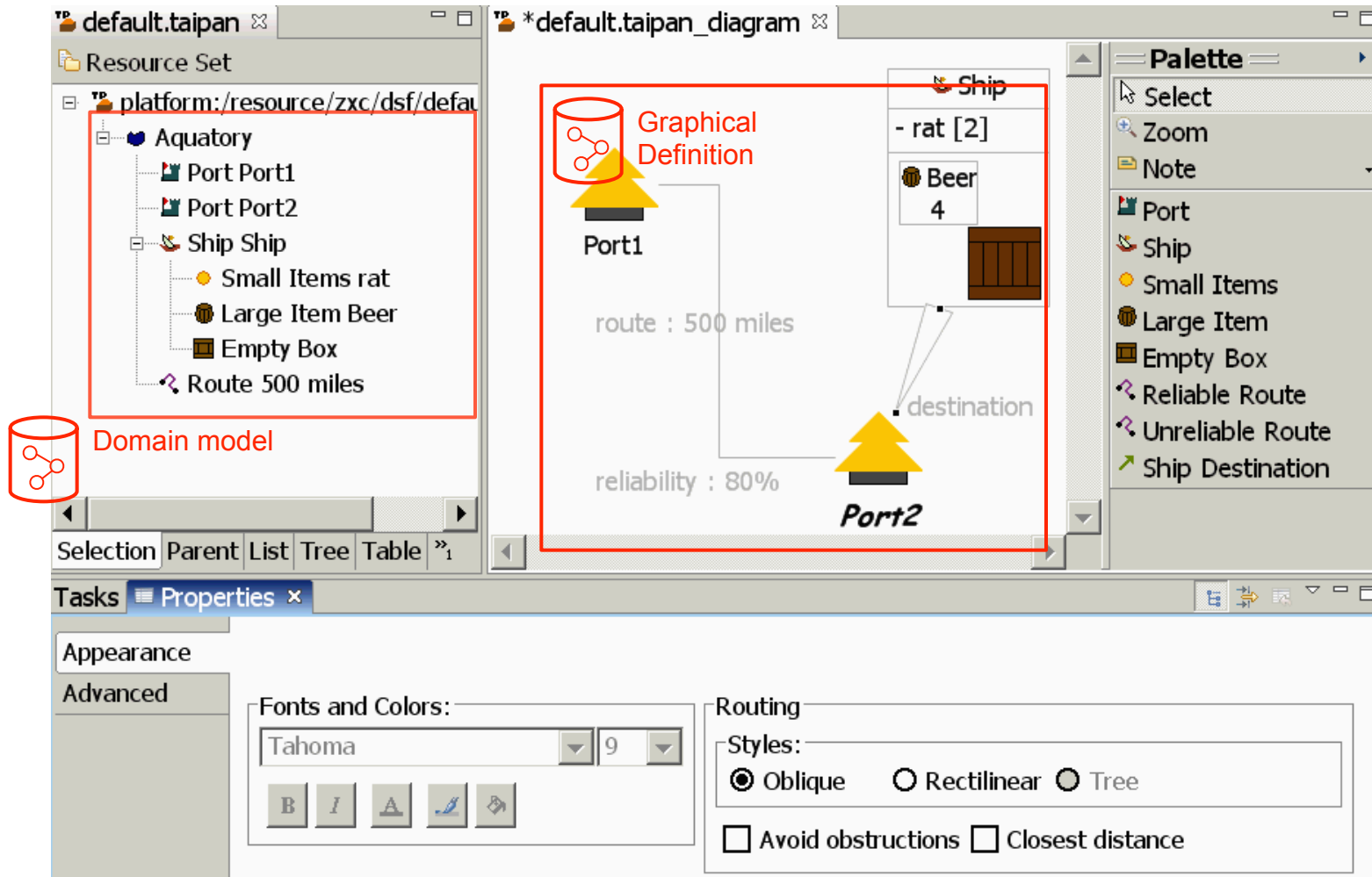
Mapping Model



Mapping Model



Mapping Model

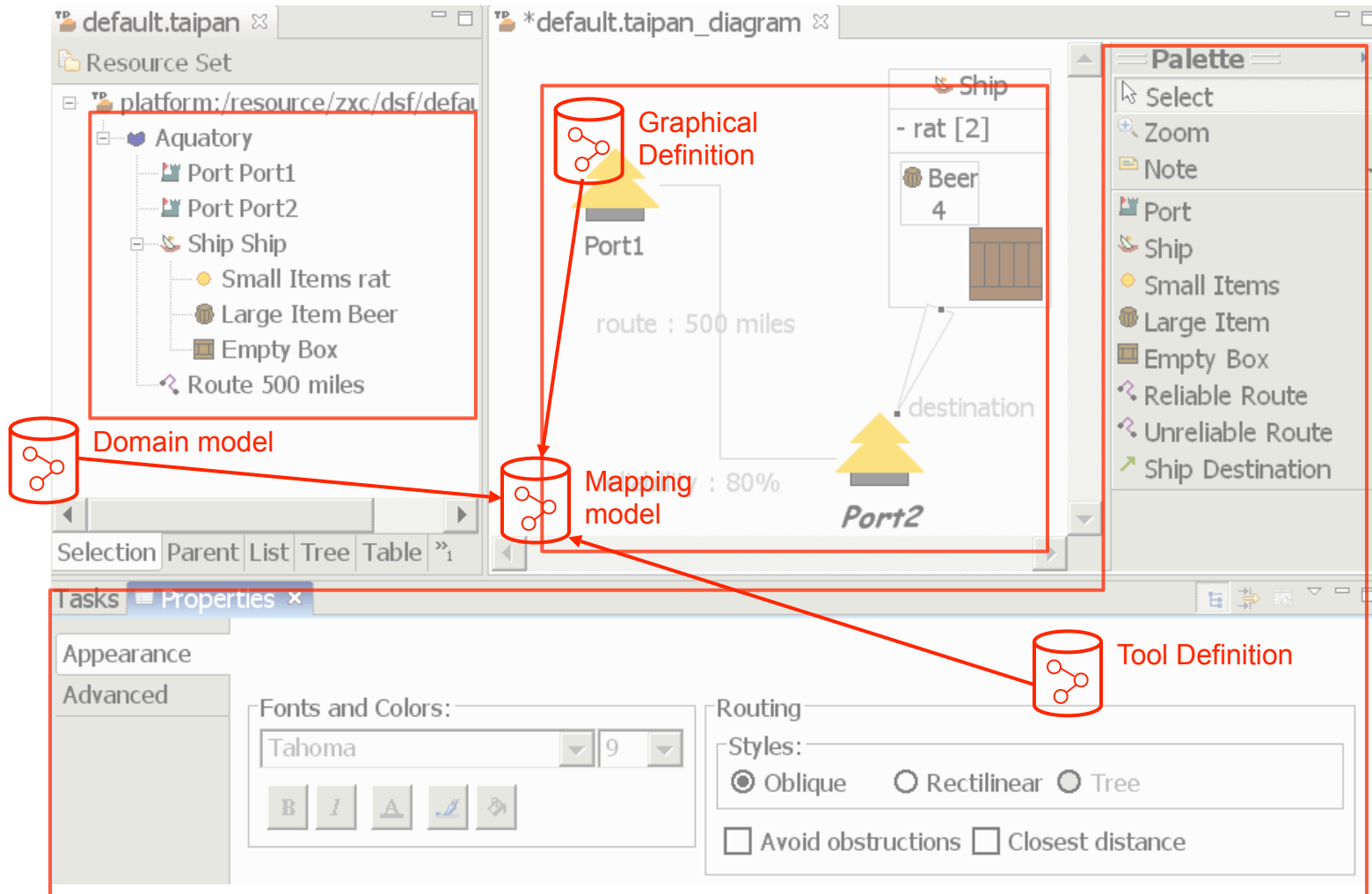


Mapping Model

The screenshot displays a software interface for a Mapping Model, divided into several panels:

- Domain model:** Located in the top-left panel, it shows a hierarchical tree structure under the path `platform:/resource/zxc/dsf/default.taipan`. The tree includes:
 - Aquatory
 - Port Port1
 - Port Port2
 - Ship Ship
 - Small Items rat
 - Large Item Beer
 - Empty Box
 - Route 500 miles
- Graphical Definition:** Located in the top-right panel, it shows a diagram of a ship's route. A yellow triangle labeled **Port1** is connected to another yellow triangle labeled **Port2** by a line labeled `route : 500 miles`. A label `reliability : 80%` is also present. A **Ship** object is shown with a `- rat [2]` property and a `Beer 4` property. A `destination` label points to the **Port2** triangle.
- Palette:** Located on the right side, it lists various tools and objects for selection:
 - Select
 - Zoom
 - Note
 - Port
 - Ship
 - Small Items
 - Large Item
 - Empty Box
 - Reliable Route
 - Unreliable Route
 - Ship Destination
- Tasks Properties:** Located at the bottom, it contains two tabs: **Appearance** and **Advanced**. The **Advanced** tab is active, showing:
 - Fonts and Colors:** A dropdown menu set to `Tahoma` and a size dropdown set to `9`. Below are icons for Bold (B), Italic (I), Underline (A), and other text formatting options.
 - Routing:** A section with a **Styles:** dropdown menu. Below it are three radio buttons: ☒ **Oblique**, ☐ **Rectilinear**, and ☐ **Tree**. At the bottom are two checkboxes: ☐ **Avoid obstructions** and ☐ **Closest distance**.

Mapping Model



Mapping Model

- Leírja a logikai kapcsolatot a következők között:
 - Domain elemek (.ecore)
 - Grafikus elemek (.gmfgraph)
 - Tooling elemek (.gmftool)
- Validációra alkalmas
 - Model Validation
- Kényszereket vehetünk fel
 - OCL nyelven
- A wizard generál egy alapváltozatot a bemeneti metamodellek alapján

Canvas Mapping

- Graphical: a diagram „háttére” (a gmfgraph Canvas gyökéreleme)
- Domain: a hierarchia gyökerét reprezentáló osztály
- Tooling:
 - Paletta
 - Menük
 - Eszközsorok

Node Mapping

- Top Node Reference
 - Domain: Containment Feature: a Canvas Mappingben megadott gyökérosztály tartalmazást reprezentáló referenciája
- Node Mapping
 - Graphical: a diagram node
 - Domain: az osztály, amit a csomópont reprezentál
 - Tooling: az osztályt létrehozó creation tool
- Gyerekei lehetnek:
 - Label Mapping
 - Child Reference
 - Compartment Mapping

Label Mapping

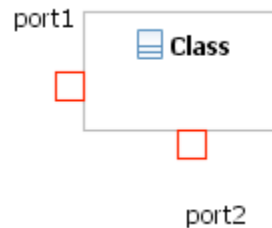
- Graphical: Diagram label
- Domain:
 - (Design) Label Mapping esetén: semmi, statikus szöveg
 - Feature Label Mapping esetén:
 - a megjelenítendő (és szerkesztendő) feature-ök
 - szöveges minták a megjelenítésükhöz és parse-olásukhoz

Property	Value
[-] Domain meta information	
Features to display	[-] ModelElement.name:EString, Property.typeName:EString
Features to edit	[-] ModelElement.name:EString
[+] Misc	
[-] Visual representation	
Edit Method	[...] MESSAGE_FORMAT
Editor Pattern	[...] {0}
Edit Pattern	[...] {0}
View Method	[...] MESSAGE_FORMAT
View Pattern	[...] {0} : {1}

Library
[-] name : EString

Child Reference

- Csomópont gyerekei
- Ugyanúgy Containment Feature és Node Mapping a megadandó adatok, mint Top Node Reference esetén
- Két típus:
 - Affixed: a szülő node oldalán jelenik meg (pl. portok)



- Compartment: a szülő node egy compartmentjében jelenik meg (következő dia)

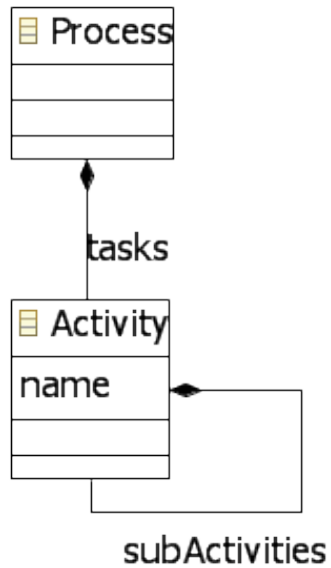
Compartment Mapping

- Graphical: Compartment grafikus definícióját kell megadni
- Child Reference-re kell hivatkozni
- Így egy compartment csak azonos típusú node-okat tartalmazhat!

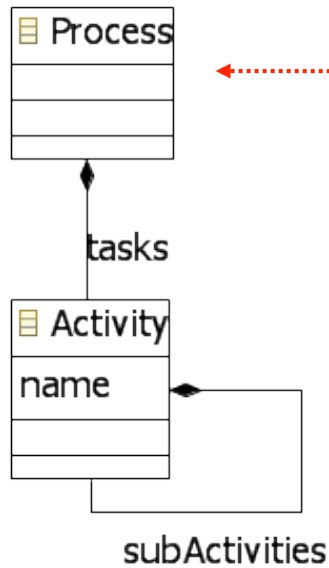


Node hierarchia – példa

Node hierarchy – példa

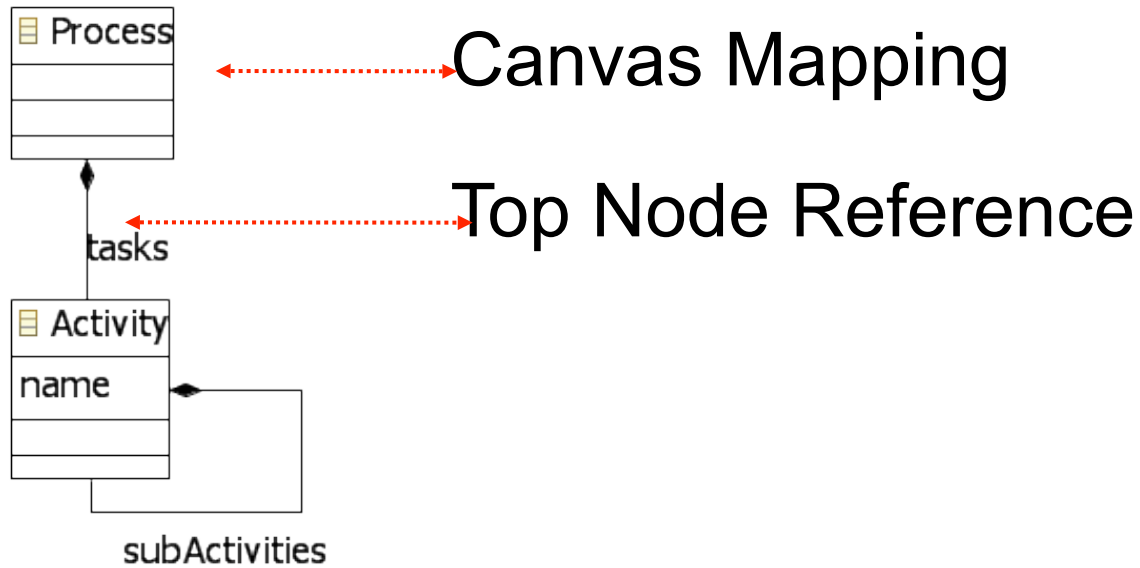


Node hierarchy – példa

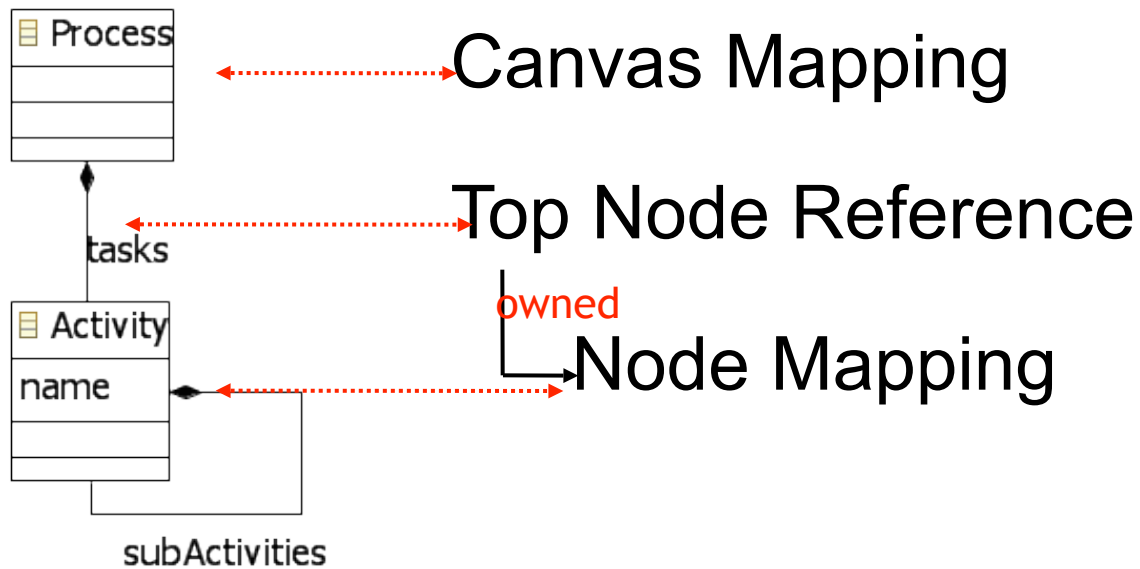


Canvas Mapping

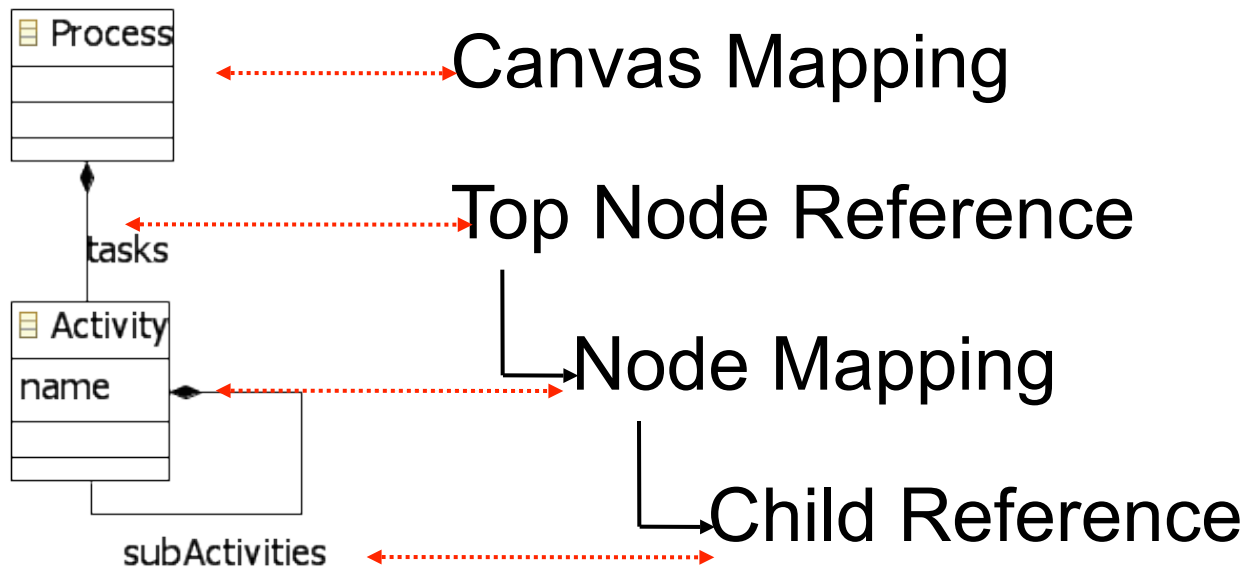
Node hierarchy – példa



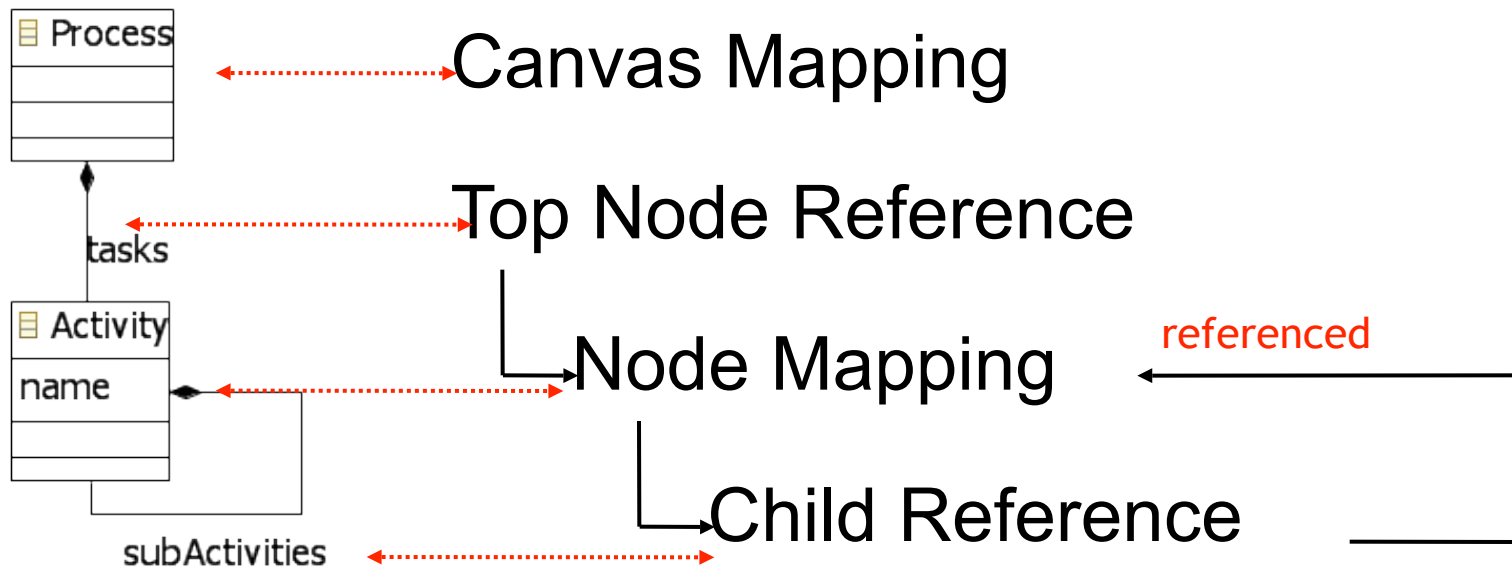
Node hierarchy – példa



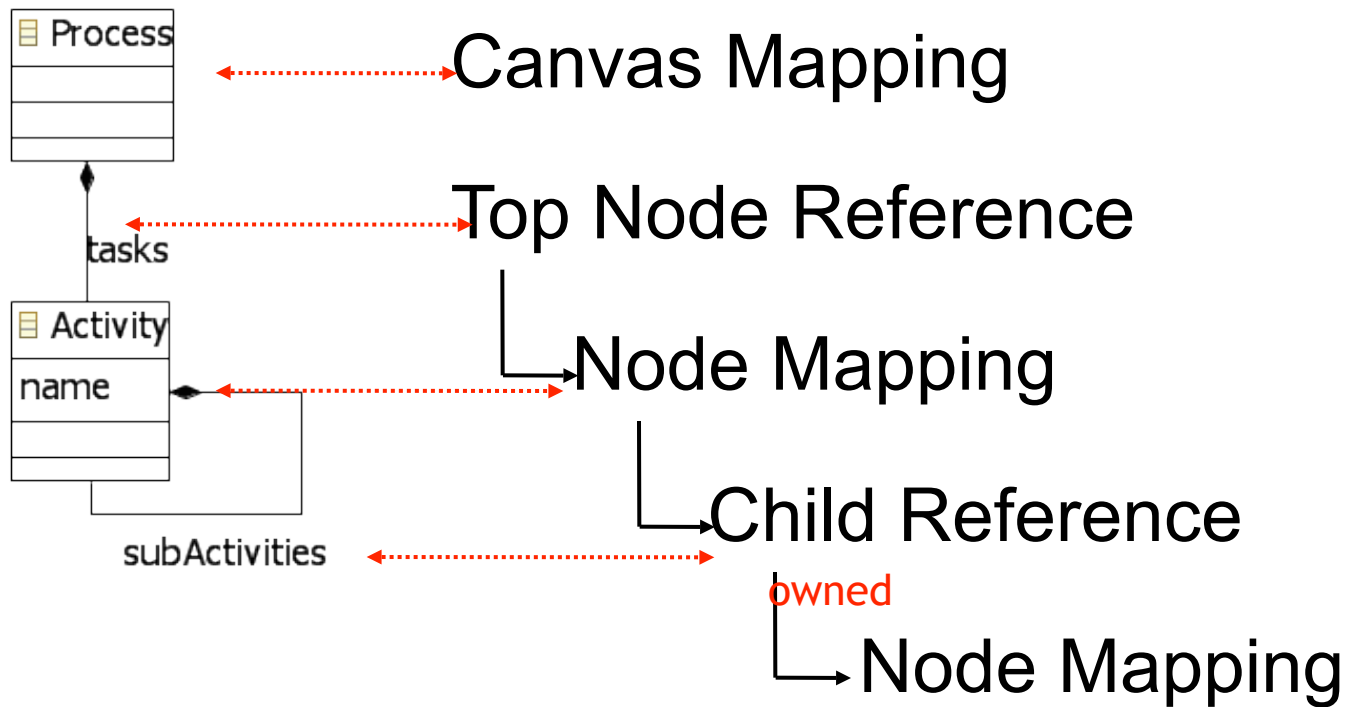
Node hierarchy – példa



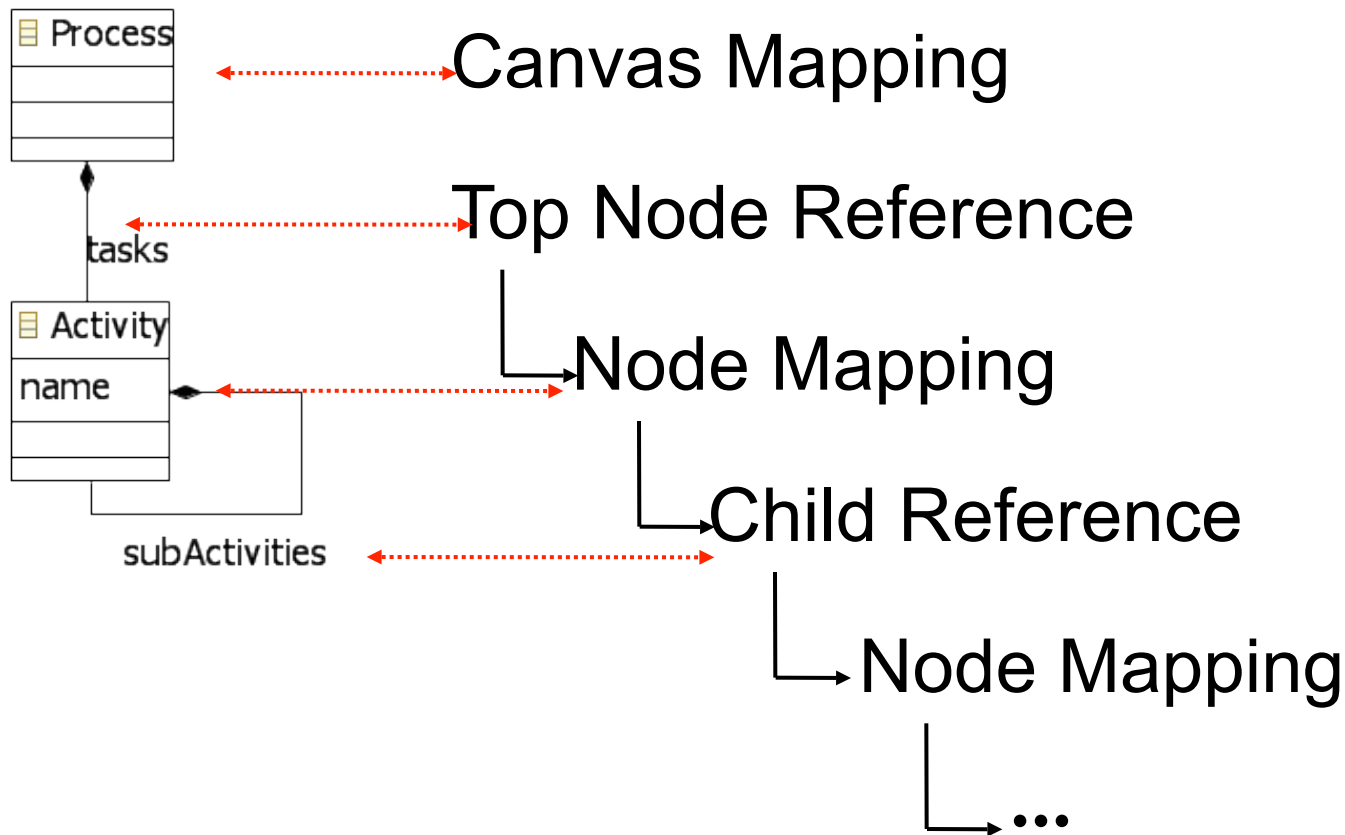
Node hierarchy – példa



Node hierarchy – példa



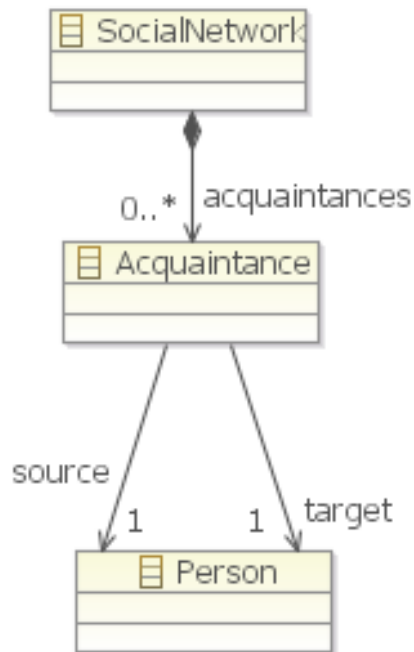
Node hierarchy – példa



Link Mapping

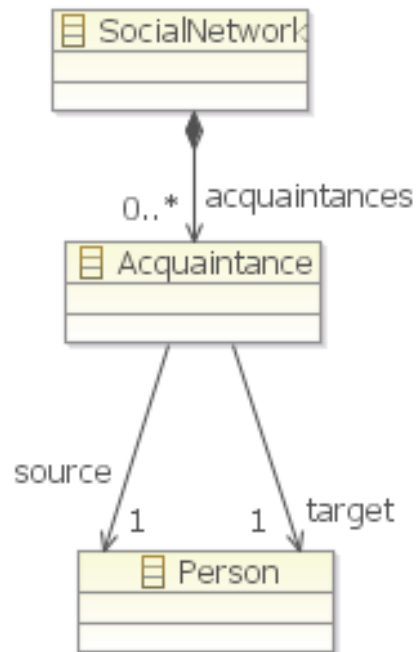
- Graphical: connection
- Domain:
 - Ha a kapcsolatot egy feature reprezentálja: a Target Feature-ben kell megadni
 - Ha a kapcsolatot egy osztály reprezentálja:
 - Element: a kapcsolatot reprezentáló osztály
 - Containment Feature: a kapcsolatokat tartalmazó feature
 - Source/Target Feature: a kapcsolat két végét reprezentáló feature
- Tooling: az összeköttetést létrehozó creation tool

Összeköttetés osztállyal – példa



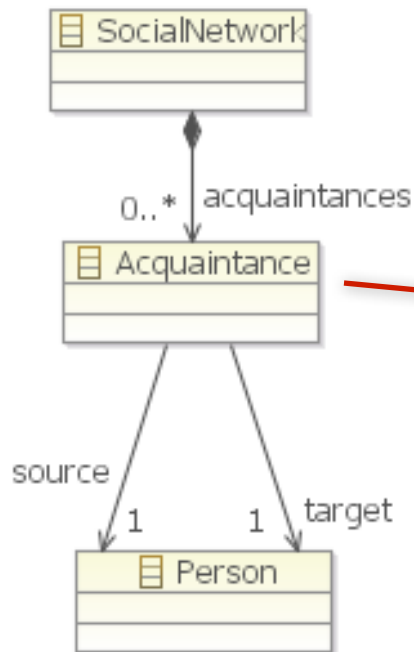
Properties	
Property	Value
Domain meta information	
Containment Feature	0..* SocialNetwork.acquaintances:Acquaintance
Element	Acquaintance
Source Feature	1 Acquaintance.source:Person
Target Feature	1 Acquaintance.target:Person
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Link	Connection Acquaintance

Összeköttetés osztállyal – példa



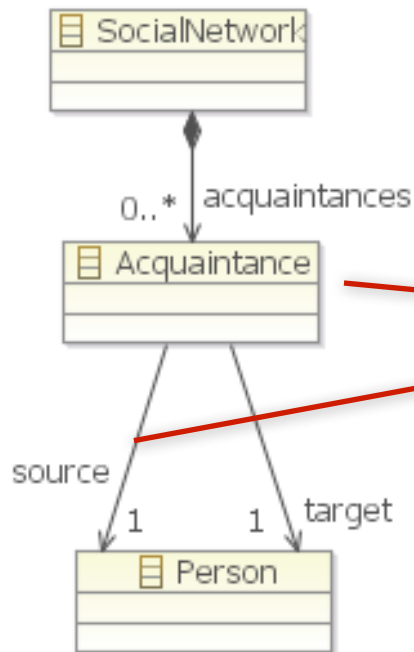
Properties	
Property	Value
Domain meta information	
Containment Feature	0..* SocialNetwork.acquaintances:Acquaintance
Element	Acquaintance
Source Feature	1 Acquaintance.source:Person
Target Feature	1 Acquaintance.target:Person
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Link	Connection Acquaintance

Összeköttetés osztállyal – példa



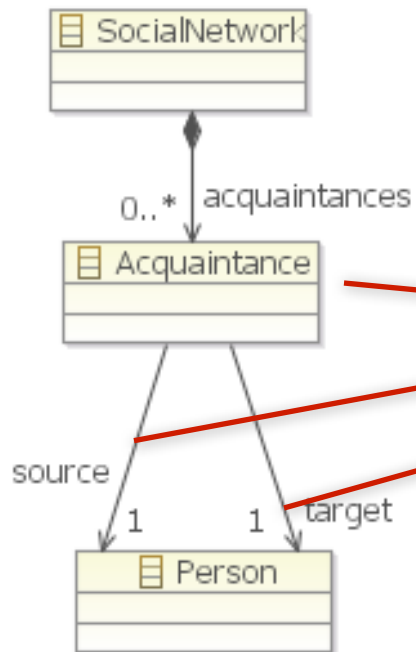
Properties	
Property	Value
Domain meta information	
Containment Feature	0..* SocialNetwork.acquaintances:Acquaintance
Element	Acquaintance
Source Feature	1 Acquaintance.source:Person
Target Feature	1 Acquaintance.target:Person
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Link	Connection Acquaintance

Összeköttetés osztállyal – példa



Properties	
Property	Value
Domain meta information	
Containment Feature	0..* SocialNetwork.acquaintances:Acquaintance
Element	Acquaintance
Source Feature	1 Acquaintance.source:Person
Target Feature	1 Acquaintance.target:Person
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Link	Connection Acquaintance

Összeköttetés osztállyal – példa



Properties	
Property	Value
Domain meta information	
Containment Feature	0..* SocialNetwork.acquaintances:Acquaintance
Element	Acquaintance
Source Feature	1 Acquaintance.source:Person
Target Feature	1 Acquaintance.target:Person
Misc	
Related Diagrams	
Visual representation	
Appearance Style	
Context Menu	
Diagram Link	Connection Acquaintance

Diagram Editor Generator Model

- Generátor modell (GMFGen)
- Kódgenerálás paramétereit tartalmazza
 - Az EMF genmodeljéhez hasonló
 - A mapping modell transzformációjával áll elő
- Kódgenerálás Java Emitter Templates/Xpand technológiával
 - A template-eket ki lehet cserélni
- A kódgenerálás a GMF runtime-on futó kódot állít elő
- A generálást finomhangolhatjuk
 - Plug-in ID, provider name, package namespace stb.
- Runtime opciók
 - Nyomtatás támogatása, validation stb.

Generátor adatok

- Lehetőség van RCP alkalmazás generálására
- Fájl tulajdonságai (Gen Editor)
 - Modell- és diagramfájl kiterjesztése
 - Modell- és diagramfájl külön vagy egyben legyen-e
- Plugin azonosító adatok (Gen Plugin)
 - ID, név, provider
- Diagramszerkesztő képességei – később (Gen Diagram)
 - Validáció
 - Shortcuts
 - Providers

Generált kód

- Komplettn kód
- Grafikus editor GEF alapon
- Sok extrával
 - Ld. Runtime features

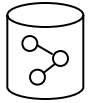
Hasonló az EMF-hez...

Hasonló az EMF-hez...

EMF

Hasonló az EMF-hez...

EMF



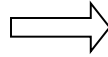
Domain model
(ECore)

Hasonló az EMF-hez...

EMF



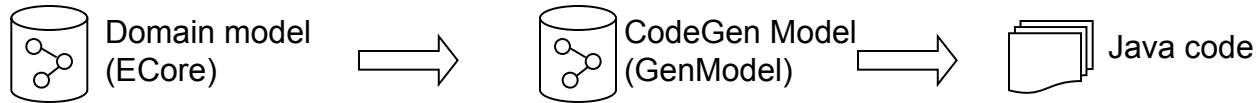
Domain model
(ECore)



CodeGen Model
(GenModel)

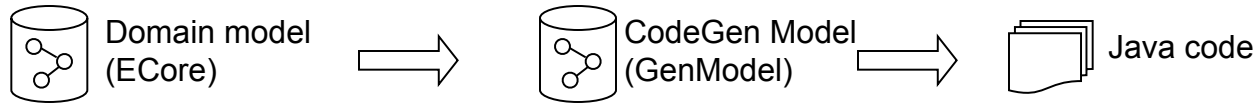
Hasonló az EMF-hez...

EMF



Hasonló az EMF-hez...

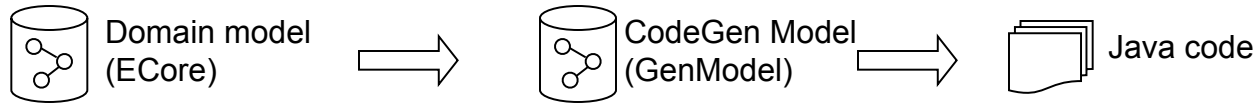
EMF



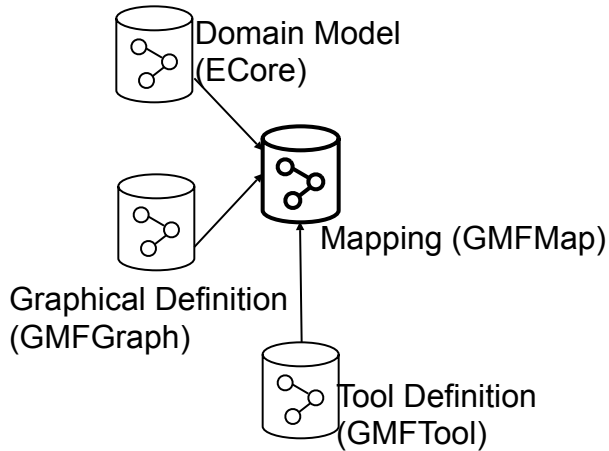
GMF

Hasonló az EMF-hez...

EMF

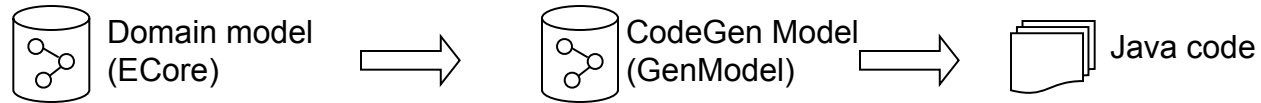


GMF

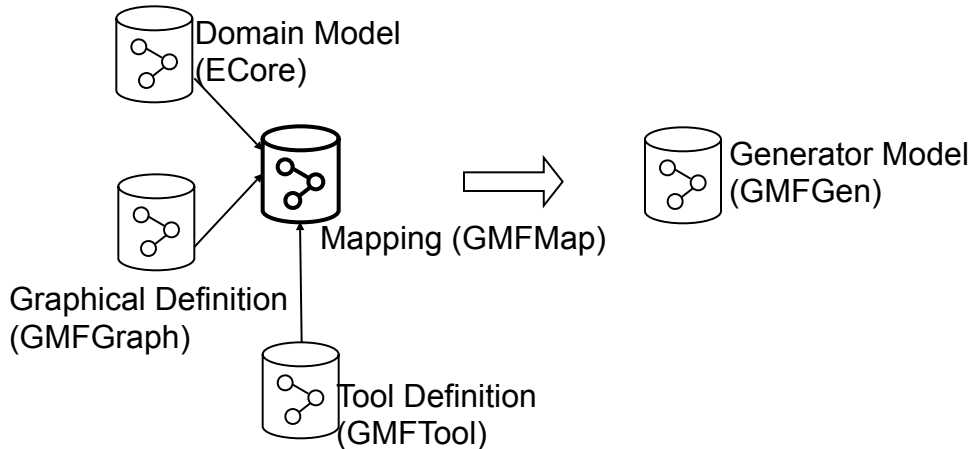


Hasonló az EMF-hez...

EMF

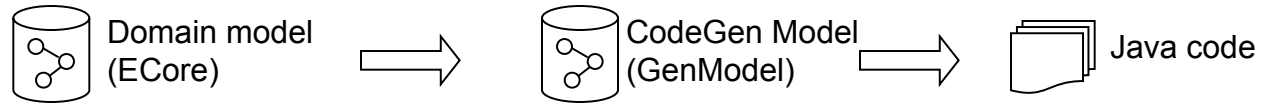


GMF

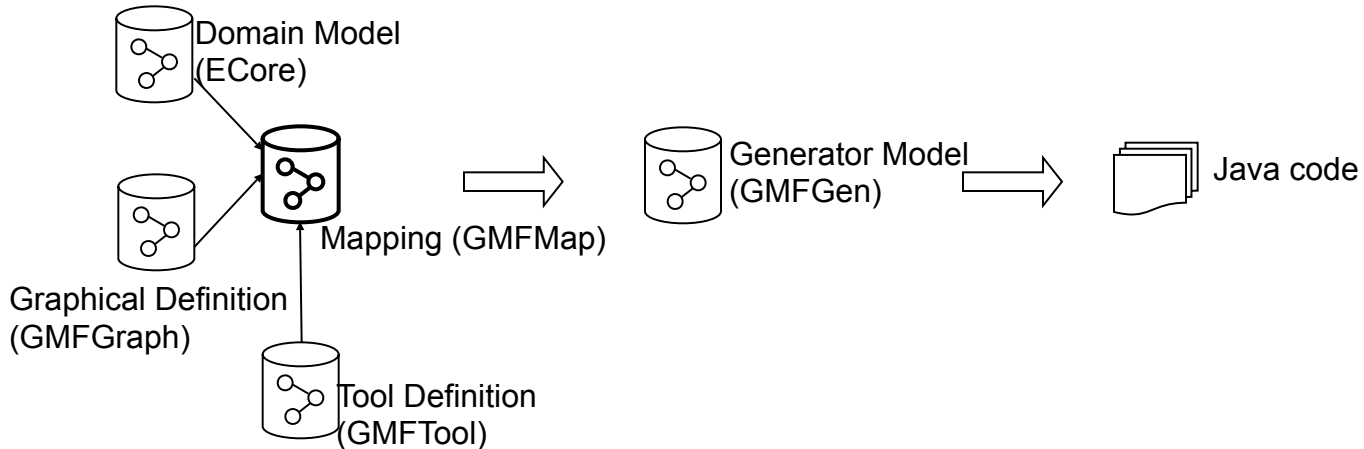


Hasonló az EMF-hez...

EMF

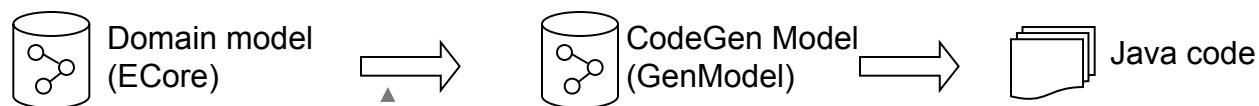


GMF

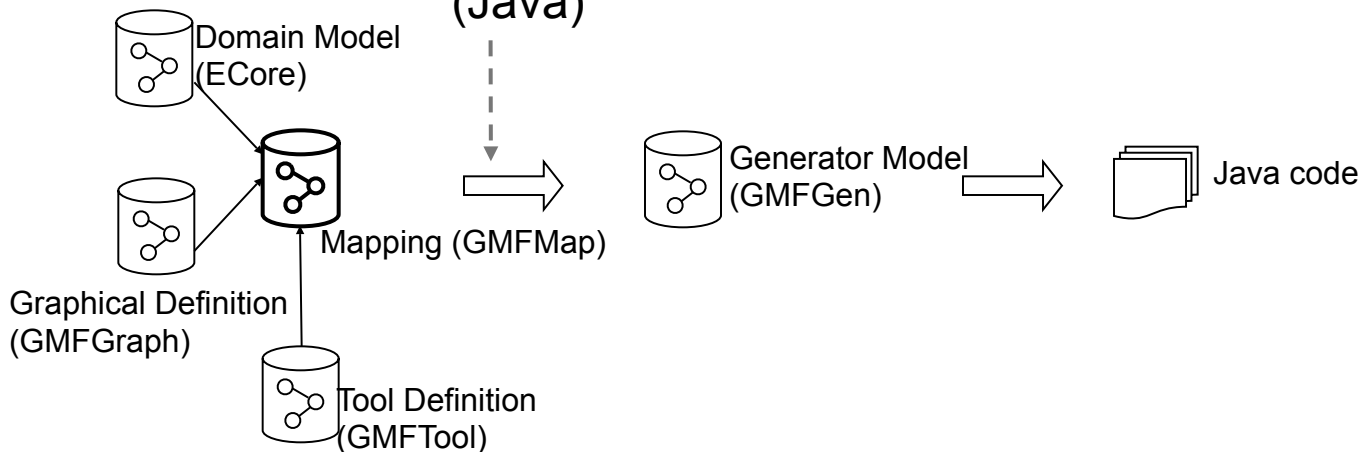


Hasonló az EMF-hez...

EMF

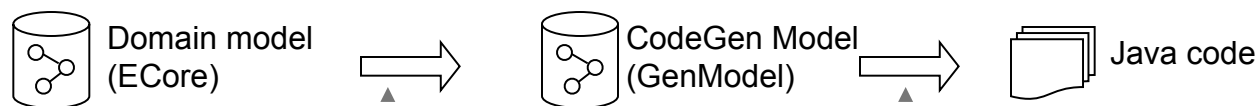


GMF

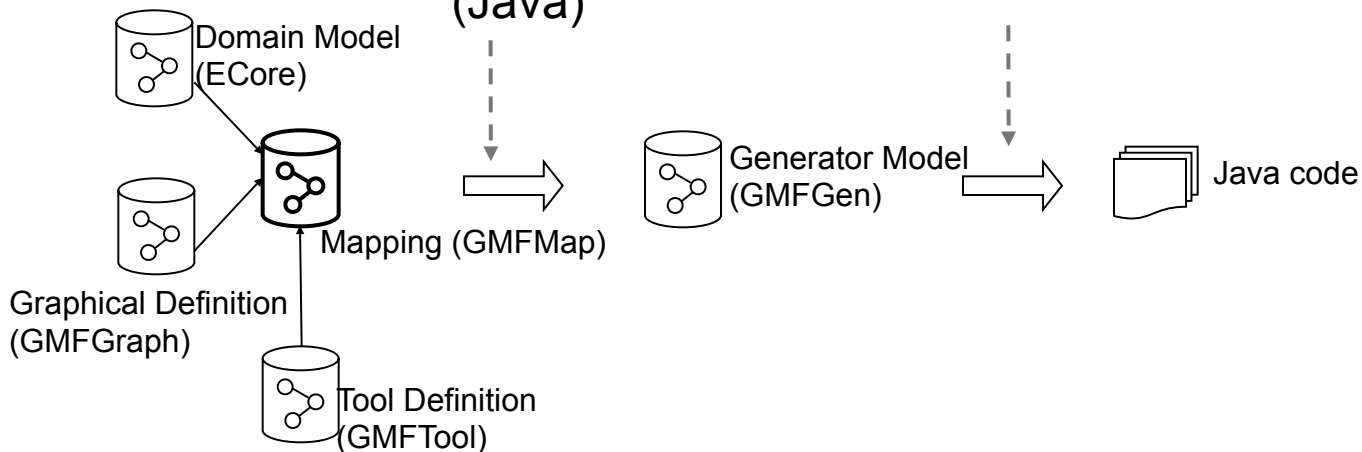


Hasonló az EMF-hez...

EMF



GMF



A generált kód testreszabása

- Lehetőségek:
 - Custom osztályok a tooling modellben
 - Kód felülírása
 - Template módosítása
 - Extension point bővítése
- Érdemes külön pluginbe rakni a módosításokat

A generált kód testreszabása – Custom osztályok

- Graphical
 - Figure, Connection, Decoration, Border, Layout
- Generator
 - Behaviour (dupla klikk lekezelése)
- Előnyök
 - Modellalapú megközelítés (még attribútumok is)
- Hátrányok
 - Refactoring, statikus ellenőrzés nehézkes modell-kód átjárhatatlanság miatt

A generált kód testreszabása – Custom osztályok

■ Példa



Custom Behaviour
hozzáadása generator
modelhez

A generált kód testreszabása – Custom osztályok

■ Példa



Custom Behaviour
hozzáadása generator
modelhez

Saját implementáció
megadása

Property	Value
Edit Policy Qualified Class Name	org.eclipse.gmf.ecore.editor.custom.MyEditPolicy
Key	EditPolicyRoles.OPEN_ROLE

A generált kód testreszabása – Kód felülírása

- Generált metódusok felülírása, újak írása
- @generated törlése vagy @generated NOT
 - Újonnan beszúrt kódnál nem kell
- Előnyök
 - Egyszerű
- Hátrányok
 - Törékeny

A generált kód testreszabása – Kód felülírása

■ Példa



```
/**
 * @generated
 */
public static Image getImageGen(IAdaptable hint) {
    ENamedElement element = getElement(hint);
    if (element == null) {
        return null;
    }
    return getImage(element);
}

/**
 * @generated NOT
 */
public static Image getImage(IAdaptable hint) {
    String iconName = null;
    if (hint == EAttribute_3001) {
        iconName = "attribute.gif";
    } else if (hint == EOperation_3002) {
        iconName = "operation.gif";
    } else if (hint == EAnnotation_3003) {
        iconName = "annotation.gif";
    }

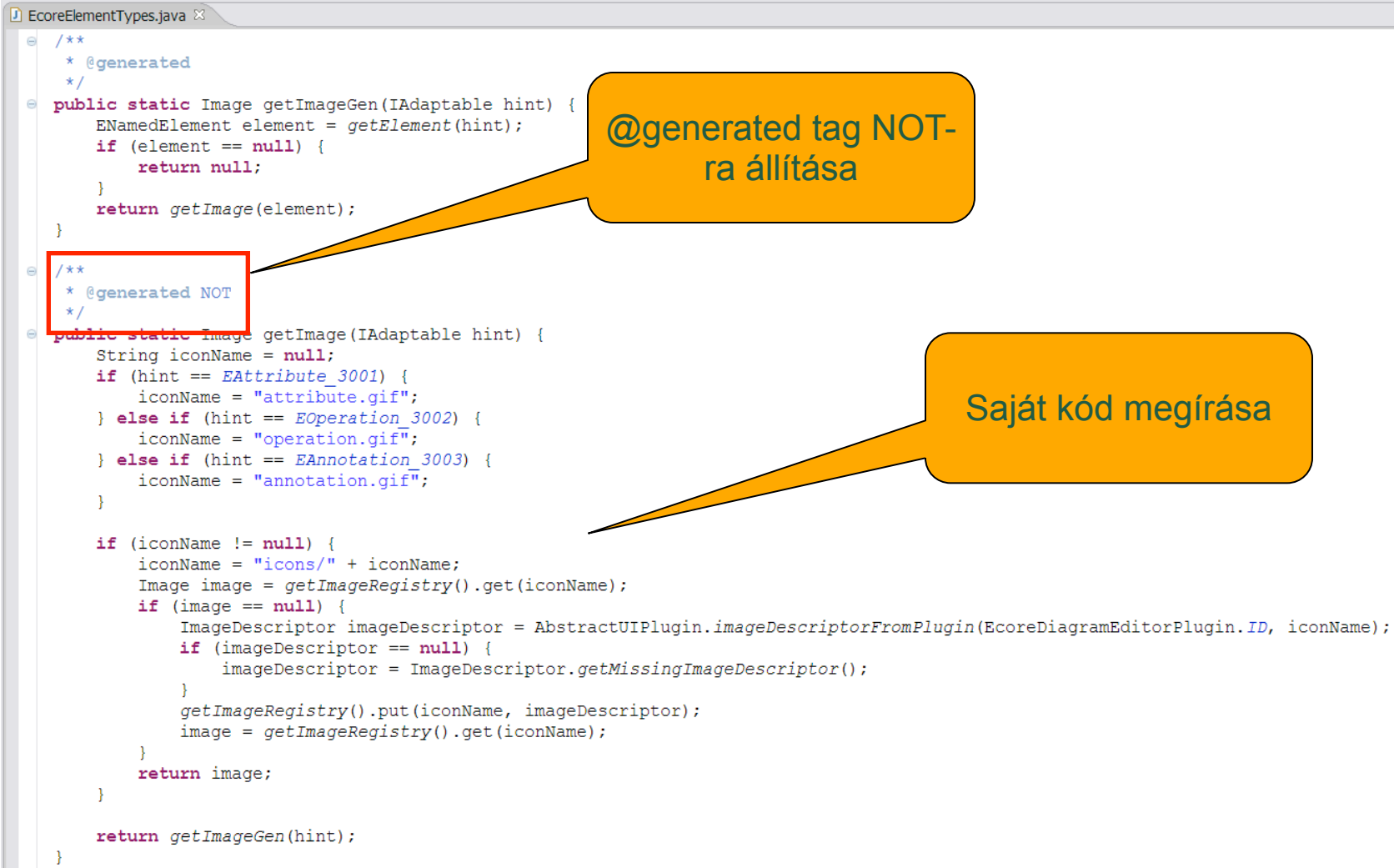
    if (iconName != null) {
        iconName = "icons/" + iconName;
        Image image = getImageRegistry().get(iconName);
        if (image == null) {
            ImageDescriptor imageDescriptor = AbstractUIPlugin.imageDescriptorFromPlugin(EcoreDiagramEditorPlugin.ID, iconName);
            if (imageDescriptor == null) {
                imageDescriptor = ImageDescriptor.getMissingImageDescriptor();
            }
            getImageRegistry().put(iconName, imageDescriptor);
            image = getImageRegistry().get(iconName);
        }
        return image;
    }

    return getImageGen(hint);
}
```

@generated tag NOT-ra állítása

A generált kód testreszabása – Kód felülírása

■ Példa



```
/**
 * @generated
 */
public static Image getImageGen(IAdaptable hint) {
    ENamedElement element = getElement(hint);
    if (element == null) {
        return null;
    }
    return getImage(element);
}

/**
 * @generated NOT
 */
public static Image getImage(IAdaptable hint) {
    String iconName = null;
    if (hint == EAttribute_3001) {
        iconName = "attribute.gif";
    } else if (hint == EOperation_3002) {
        iconName = "operation.gif";
    } else if (hint == EAnnotation_3003) {
        iconName = "annotation.gif";
    }

    if (iconName != null) {
        iconName = "icons/" + iconName;
        Image image = getImageRegistry().get(iconName);
        if (image == null) {
            ImageDescriptor imageDescriptor = AbstractUIPlugin.imageDescriptorFromPlugin(EcoreDiagramEditorPlugin.ID, iconName);
            if (imageDescriptor == null) {
                imageDescriptor = ImageDescriptor.getMissingImageDescriptor();
            }
            getImageRegistry().put(iconName, imageDescriptor);
            image = getImageRegistry().get(iconName);
        }
        return image;
    }

    return getImageGen(hint);
}
```

@generated tag NOT-ra állítása

Saját kód megírása

A generált kód testreszabása – Template

- JET sablonok módosítása
 - Generator modellben template directory megadása
- Előnyök
 - Újrahasznosítható
- Hátrányok
 - JET technológia ismerete szükséges

A generált kód testreszabása – Template

■ Példa

```
«AROUND getAdaptableImage FOR gmfgn::GenDiagram-»
  «EXPAND xpt::Common::generatedMemberComment»
  public static org.eclipse.swt.graphics.Image getImage(org.eclipse.core.runtime.IAdaptable hint) {
    org.eclipse.gmf.runtime.emf.type.core.IElementType elementType = (org.eclipse.gmf.runtime.emf.type.core.IElementType)
      hint.getAdapter(org.eclipse.gmf.runtime.emf.type.core.IElementType.class);
    «EXPAND addCustomIcon FOREACH palette.groups.entries-»
    «EXPAND xpt::diagram::providers::ElementTypes::getNamedElement-»
    return getImage(element);
  }
«ENDAROUND»

«DEFINE addCustomIcon FOR gmfgn::ToolEntry-»
  «IF null != largeIconPath-»
    «EXPAND getImage(this) FOREACH genNodes-»
  «ENDIF-»
«ENDDFINE»

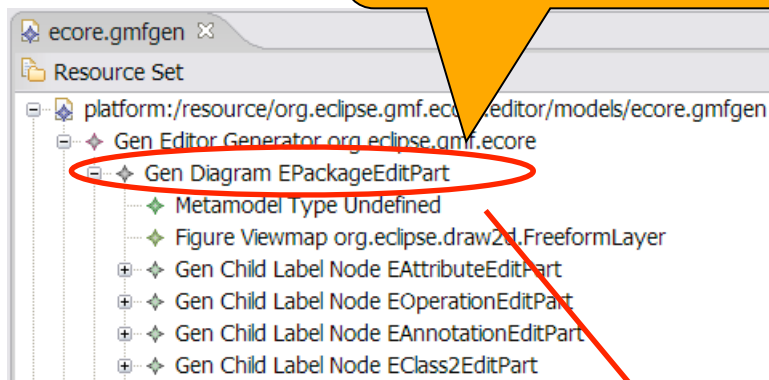
«DEFINE getImage(gmfgn::ToolEntry entry) FOR gmfgn::GenNode-»
if (elementType == «getUniqueIdentifier()») {
  String key = "«entry.largeIconPath»";
  org.eclipse.swt.graphics.Image image = getImageRegistry().get(key);
  if (image == null) {
    org.eclipse.jface.resource.ImageDescriptor imageDescriptor = org.eclipse.ui.plugin.AbstractUIPlugin.
      imageDescriptorFromPlugin(«getDiagram().editorGen.plugin.getActivatorQualifiedClassName()».ID, key);
    if (imageDescriptor == null) {
      imageDescriptor = org.eclipse.jface.resource.ImageDescriptor.getMissingImageDescriptor();
    }
    getImageRegistry().put(key, imageDescriptor);
    image = getImageRegistry().get(key);
  }
  return image;
}
«ENDDFINE»
```

**Xpand template
megírása**

A generált kód testreszabása – Template

■ Példa

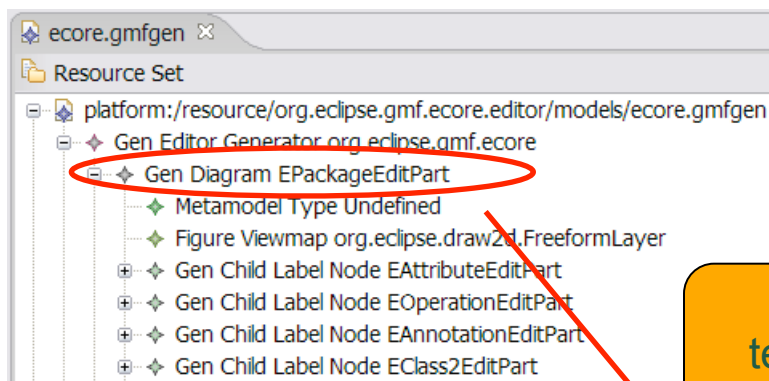
Generator modelben
megfelelő elem
tulajdonságai között...



Property	Value
Copyright Text	Copyright (c) 2006, 2007 Borland Software
Diagram File Extension	ecore_diagram
Domain File Extension	ecore
Domain Gen Model	Ecore
Dynamic Templates	true
Model ID	Ecore
Package Name Prefix	org.eclipse.gmf.ecore
Same File For Diagram And Model	false
Template Directory	/org.eclipse.gmf.ecore.editor/templates/

A generált kód testreszabása – Template

■ Példa

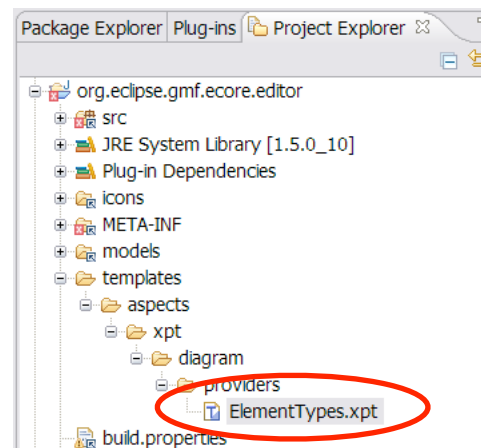
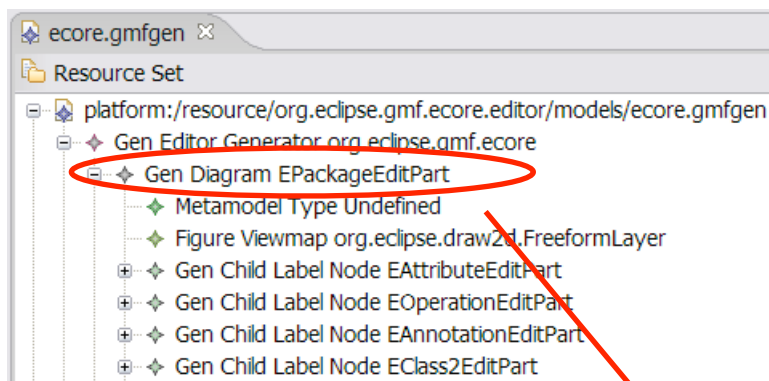


Template
testreszabásának
jelzése

Property	Value
Copyright Text	Copyright (c) 2006, 2007 Borland Software
Diagram File Extension	ecore_diagram
Domain File Extension	ecore
Domain Gen Model	Ecore
Dynamic Templates	true (highlighted with a red circle)
Model ID	Ecore
Package Name Prefix	org.eclipse.gmf.ecore
Same File For Diagram And Model	false
Template Directory	/org.eclipse.gmf.ecore.editor/templates/

A generált kód testreszabása – Template

■ Példa



Properties	
Property	Value
Copyright Text	Copyright (c) 2006, 2007 Borland Software
Diagram File Extension	ecore_diagram
Domain File Extension	ecore
Domain Gen Model	Ecore
Dynamic Templates	true
Model ID	Ecore
Package Name Prefix	org.eclipse.gmf.ecore
Same File For Diagram And Model	false
Template Directory	/org.eclipse.gmf.ecore.editor/templates/

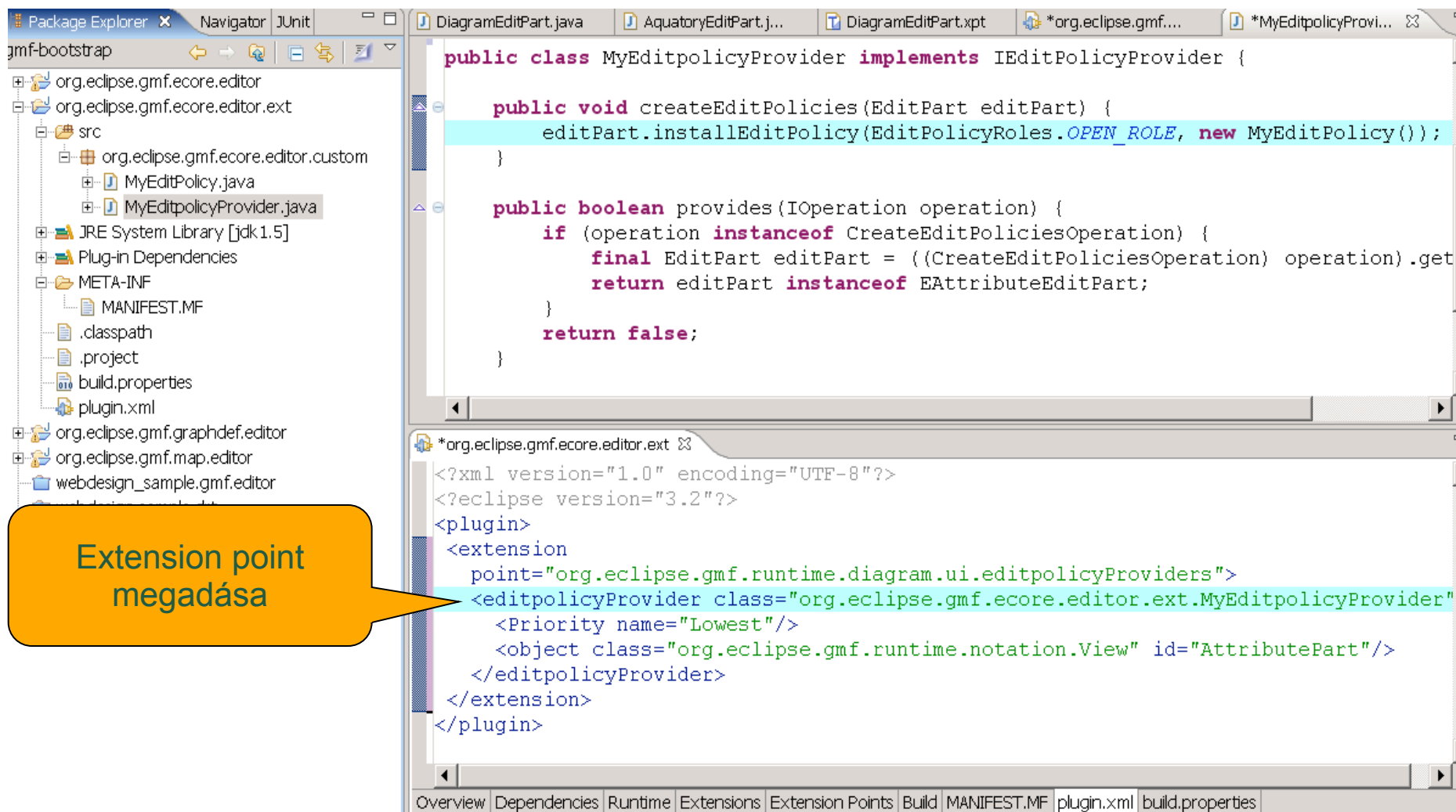
Hivatkozás saját
template-re

A generált kód testreszabása – Extension point

- *Provider extension point bővítése
 - View, EditPart, EditPolicy, Icon stb.
- Előnyök
 - Biztonságos
- Hátrányok
 - Sok boilerplate code

A generált kód testreszabása – Extension point

■ Példa



The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows the project structure for 'gmf-bootstrap', including the 'src' directory with 'org.eclipse.gmf.ecore.editor.custom' containing 'MyEditPolicy.java' and 'MyEditpolicyProvider.java'. The main editor shows the code for 'MyEditpolicyProvider' which implements 'IEditPolicyProvider'. The 'createEditPolicies' method is highlighted in blue. Below it, the 'provides' method is shown. The bottom editor shows the 'plugin.xml' file with an extension point definition for 'org.eclipse.gmf.runtime.diagram.ui.editpolicyProviders'. The 'editpolicyProvider' element is highlighted in blue. An orange callout bubble points to this element with the text 'Extension point megadása'.

```
public class MyEditpolicyProvider implements IEditPolicyProvider {  
  
    public void createEditPolicies(EditPart editPart) {  
        editPart.installEditPolicy(EditPolicyRoles.OPEN_ROLE, new MyEditPolicy());  
    }  
  
    public boolean provides(IOperation operation) {  
        if (operation instanceof CreateEditPoliciesOperation) {  
            final EditPart editPart = ((CreateEditPoliciesOperation) operation).get  
            return editPart instanceof EAttributeEditPart;  
        }  
        return false;  
    }  
}
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<?eclipse version="3.2"?>  
<plugin>  
    <extension  
        point="org.eclipse.gmf.runtime.diagram.ui.editpolicyProviders">  
        <editpolicyProvider class="org.eclipse.gmf.ecore.editor.ext.MyEditpolicyProvider"  
            <Priority name="Lowest"/>  
            <object class="org.eclipse.gmf.runtime.notation.View" id="AttributePart"/>  
        </editpolicyProvider>  
    </extension>  
</plugin>
```

Extension point megadása

A generált kód testreszabása – Extension point

■ Példa

Hivatkozott osztály
megvalósítása

Extension point
megadása

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure for 'gmf-bootstrap', including packages like 'org.eclipse.gmf.ecore.editor' and 'org.eclipse.gmf.ecore.editor.ext'. The main editor area shows two files:

- DiagramEditPart.java**: A Java class implementing `IEditPolicyProvider`. The `createEditPolicies` method is highlighted in blue, showing the instantiation of `MyEditPolicy`.
- *org.eclipse.gmf.ecore.editor.ext**: An XML plugin file. The `<editpolicyProvider class="org.eclipse.gmf.ecore.editor.ext.MyEditpolicyProvider">` line is highlighted in blue. An orange callout points from this line to the `MyEditpolicyProvider` class in the Java file above.

The bottom of the IDE shows a series of tabs: Overview, Dependencies, Runtime, Extensions, Extension Points, Build, MANIFEST.MF, plugin.xml, and build.properties.

Haladó GMF technikák

Diagram partitioning

- Dig in/Drill down
- Két eset:
 - Azonos diagramtípus (rekurzív tartalmazás)
 - Pl. package-ek
 - Különböző diagramtípus
 - Pl. séma-tábla

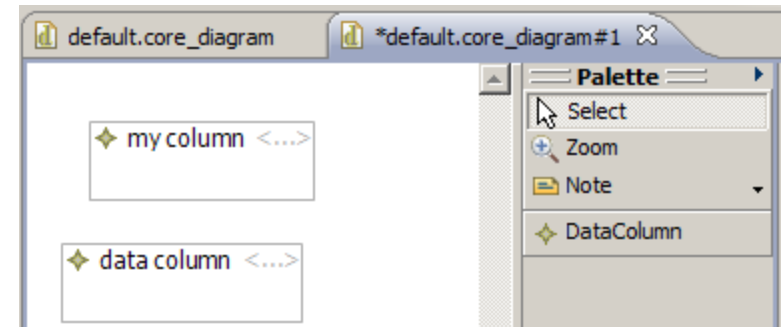
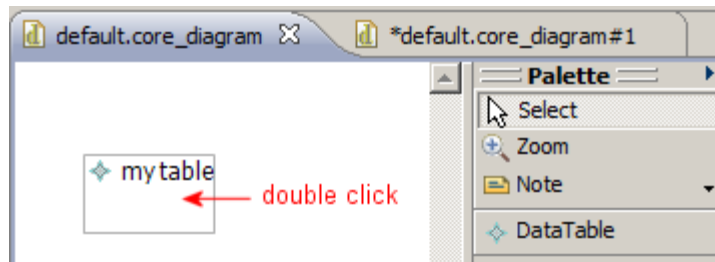


Diagram partitioning megvalósítása

- Generator model
 - Node
 - Open Diagram Behaviour
- Azonos diagramtípus:
 - Diagram Kind, Editor ID üresen hagyandó
- Különböző diagramtípus:
 - Diagram Kind: céleditor Editor Generator/Model ID tulajdonsága
 - Editor ID: céleditor Editor Generator/Editor View/ID tulajdonsága

Kezdőérték megadása

- EMF modellben
 - Default value nevű property
 - Egyszerű érték
- GMF modellben: Mapping model
 - Node/Link Mapping
 - Feature Seq Initializer
 - Feature Value Spec
 - » Value Expression
 - OCL kifejezés

Kényszerek – Összeköttetések

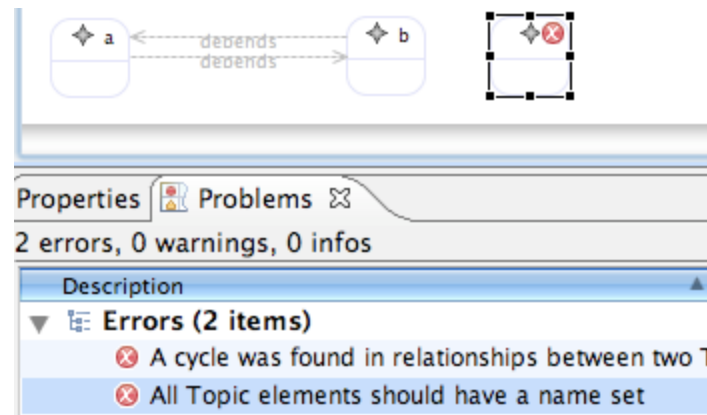
- OCL nyelven
- Mapping Model
 - Link Mapping
 - Link Constraint
- Pl. összekötő nem köthet össze egy elemet önmagával:
 - self <> oppositeEnd
- Már létrehozásnál sem sérülhet kényszer

Kényszerek

- Ugyanúgy OCL nyelven
- Mapping model
 - Audit Container
 - Audit Rule
 - Target
 - Constraint
- Csak validációnál van hibajelzés

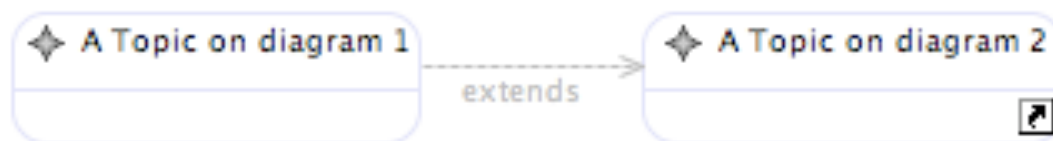
Validáció

- Engedélyezés: Generator model
 - Gen Diagram
 - Validation Enabled
 - Validation Decorators
 - Live Validation UI Feedback
- Validáció elvégzése: Edit/Validate



Hivatkozás (shortcut)

- Egy node élő másolata
- Use case: egy elemre másik diagramban szeretnénk hivatkozni
- Engedélyezés: Generator Model
 - Gen Diagram
 - Forrásdiagramban: Shortcuts Provided For = céldiagram fájlkiterjesztése
 - Céldiagramban: Contains Shortcuts To = forrásdiagram fájlkiterjesztése



Szemantikus információk tükrözése vizuálisan

- Ha csak bizonyos megjelenítési jellemzők függenek az állapottól:
- Általunk definiált Figure-metódus (pl. `updateFace()`) meghívása
 - Konstruktorban
 - EMF értesítés esetén

```
@Override  
protected void handleNotificationEvent(Notification notification) {  
    if (notification.getNotifier() instanceof Port) {  
        getPrimaryShape().updateFace();  
    }  
    super.handleNotificationEvent(notification);  
}
```


Szemantikus információk tükrözése vizuálisan

- Ha állapottól függően teljesen más figure kell:
- 2 különböző node, különböző constraintekkel és kezdőértékekkel

Mapping

Top Node Reference

Node Mapping

Constraint

Feature Seq Initializer

Feature Value Spec

Value Expression

Diagram Node: Node TypeAFigure

Body: self.type = CommonClassType::TypeA, **Language:** ocl

Element Class: CommonClass

Feature: CommonClass.type:CommonClassType

Body: CommonClass.type::TypeA, **Language:** ocl

Top Node Reference

Node Mapping

Constraint

Feature Seq Initializer

Feature Value Spec

Value Expression

Diagram Node: Node TypeBFigure

Body: self.type = CommonClassType::TypeB, **Language:** ocl

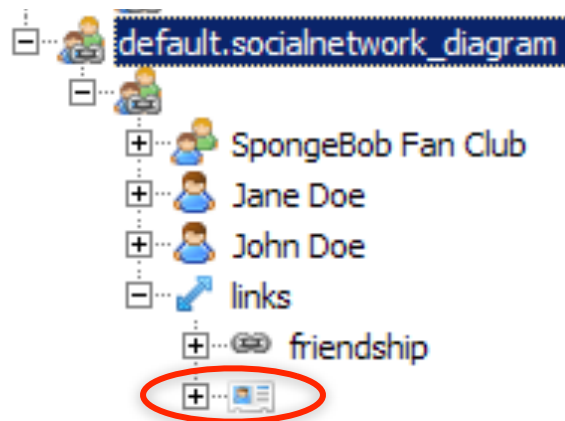
Element Class: CommonClass

Feature: CommonClass.type:CommonClassType

Body: CommonClass.type::TypeB, **Language:** ocl

Ikonok megváltoztatása – összekötők

- Ha nincs hozzá külön osztály:
- Asszociációhoz kép rendelése ImageRegistryben



Ikonok megváltoztatása – összekötők

- Ha nincs hozzá külön osztály:
- Asszociációhoz kép rendelése ImageRegistryben

```
/**
 * @generated NOT
 */
private static ImageRegistry getImageRegistry() {
    if (imageRegistry == null) {
        imageRegistry = new ImageRegistry();
        imageRegistry.put(
            getImageRegistryKey(SocialNetworkPackage.eINSTANCE
                .getPerson_Membership()),
            SocialNetworkDiagramEditorPlugin
                .findImageDescriptor("icons/Membership.gif"));
    }
    return imageRegistry;
}
```

Összegzés

	GEF	GMF
Modell	Tetszőleges	EMF
Nem gráf jellegű megjelenítés	Viszonylag könnyű	Sok, bonyolult testreszabás
Kód mennyisége	Sok boilerplate kód	Főleg modellezés, kevés kódolás
Munkafolyamat	Csak kódolás	Többlépcsős

❖ Megéri használni, ha:

- ⦿ Nem kell EMF-től eltérő modellhez igazodni
- ⦿ A modell nem változik gyakran
- ⦿ A megjelenítés nem tér el nagyon a csúcs-él paradigmától
- ⦿ Gazdag funkcionalitással rendelkező szerkesztő gyors implementációja szükséges
- ⦿ Integráció kell az Eclipse többi modellezési eszközével