Domain-specific modeling
with the Eclipse Modeling Framework

Ákos Horváth
Gábor Bergmann
Dániel Varró
István Ráth

Model Driven Software Development
Lecture 4a

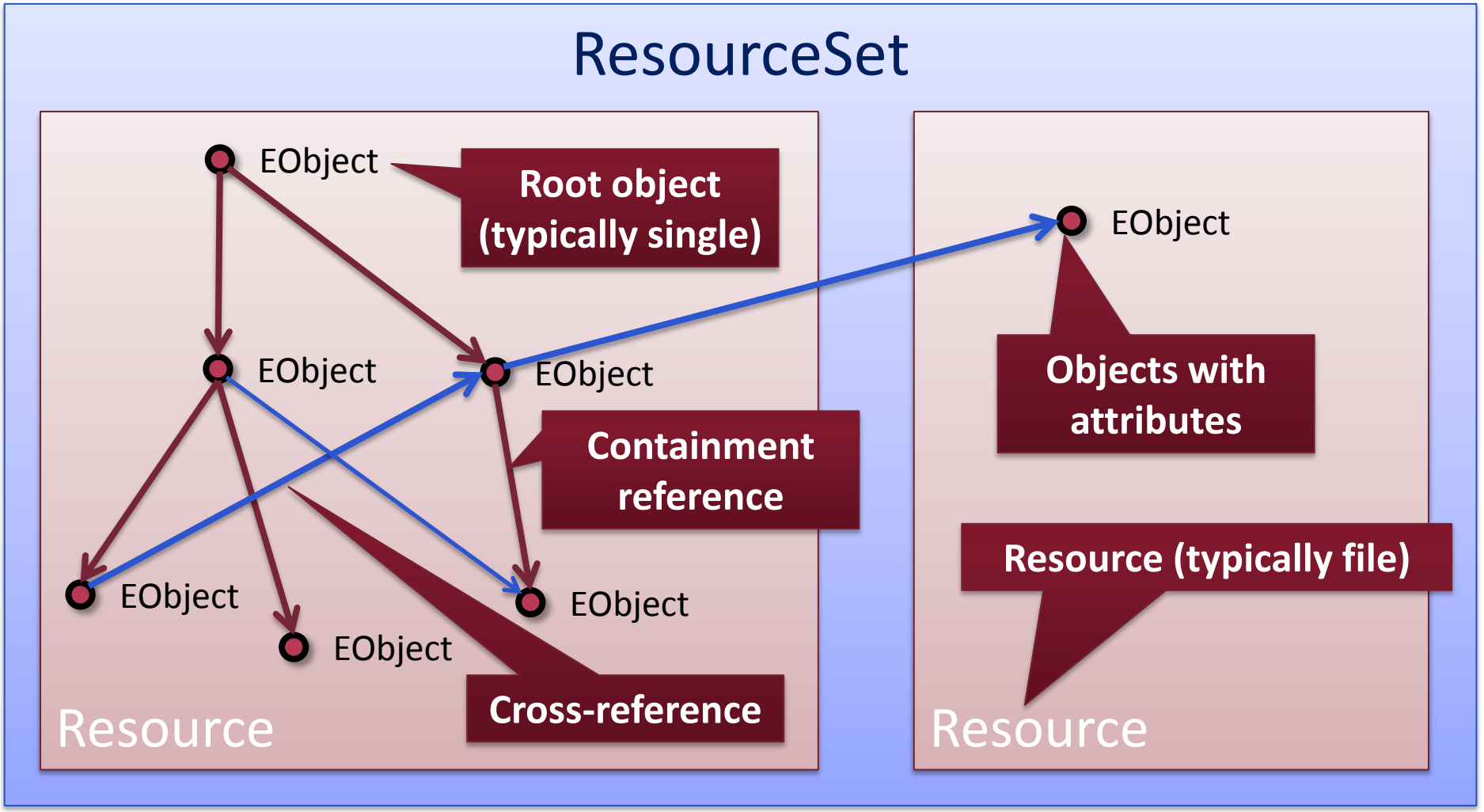
ECLIPSE MODELING FRAMEWORK

What does EMF provide?

- EMF = Eclipse Modeling Framework
  - Reflective Metamodeling Core (Ecore ➔ MOF 2.0)
  - Support for Domain Specific Languages
  - Editing Support (Notification, Undo, Commands)
  - Basic Editor Support
  - XMI Serialization, DB Persistence
  - Eclipse Integration

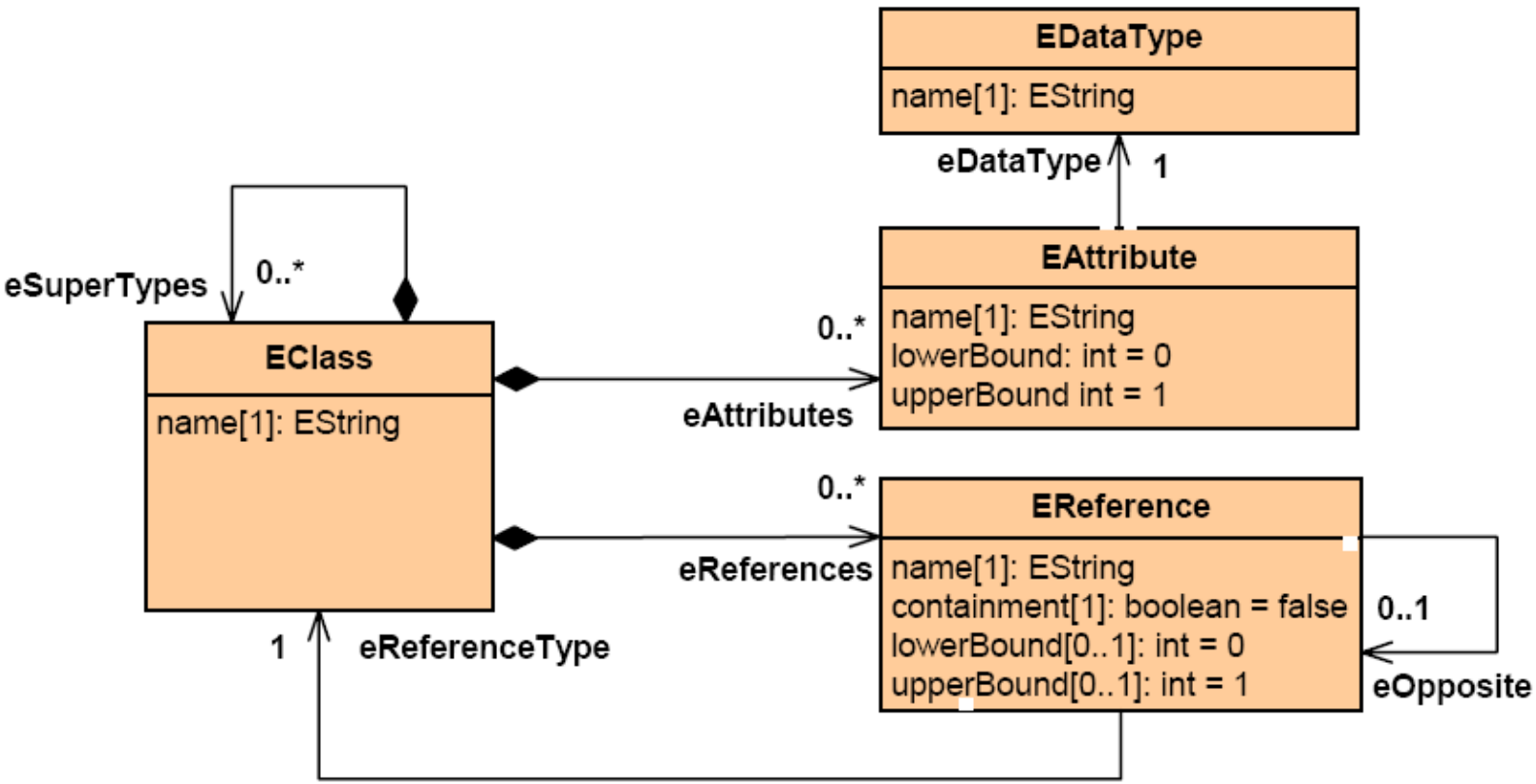
EMF model structure

- Containment hierarchy



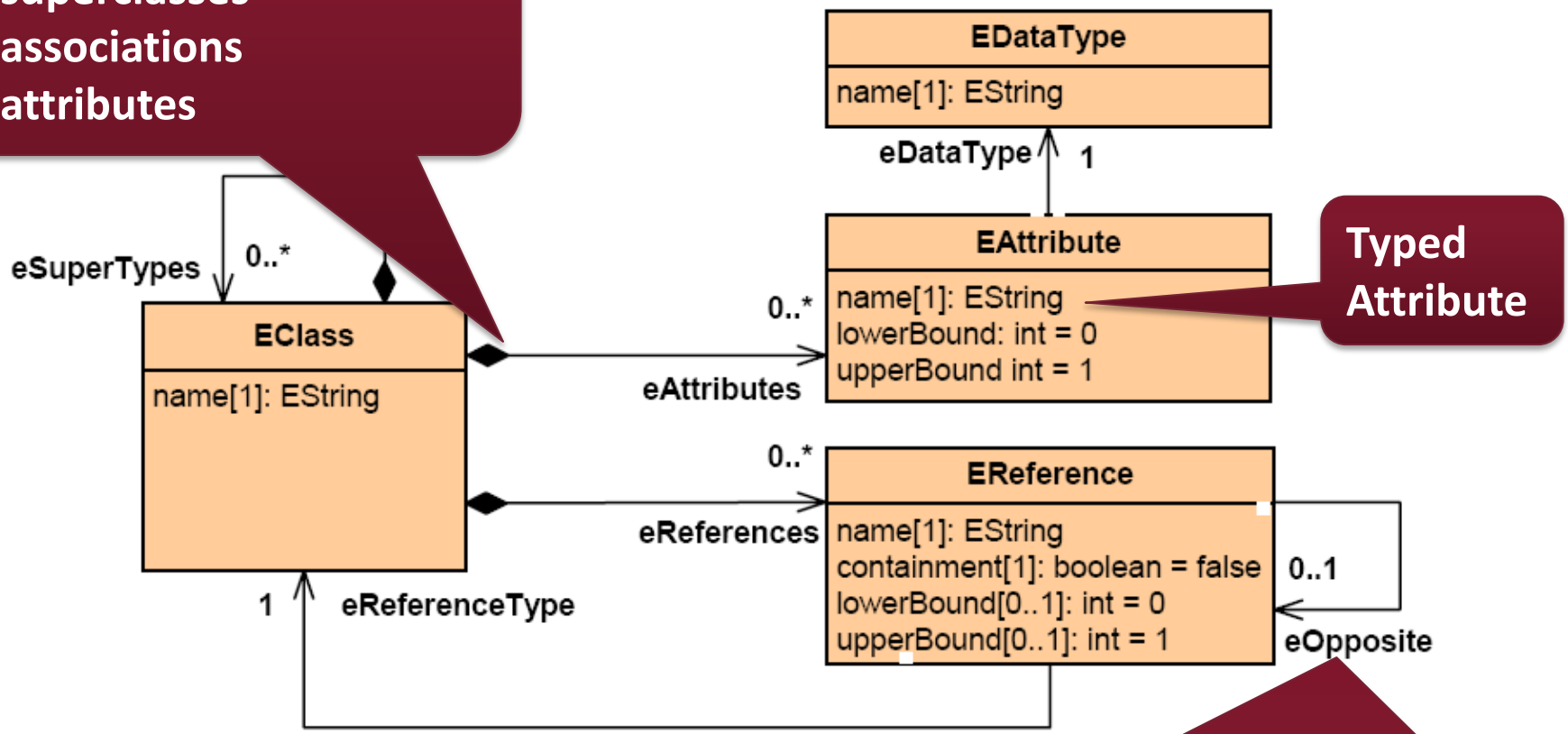
ECORE METAMODELLING

Core Ecore constructs



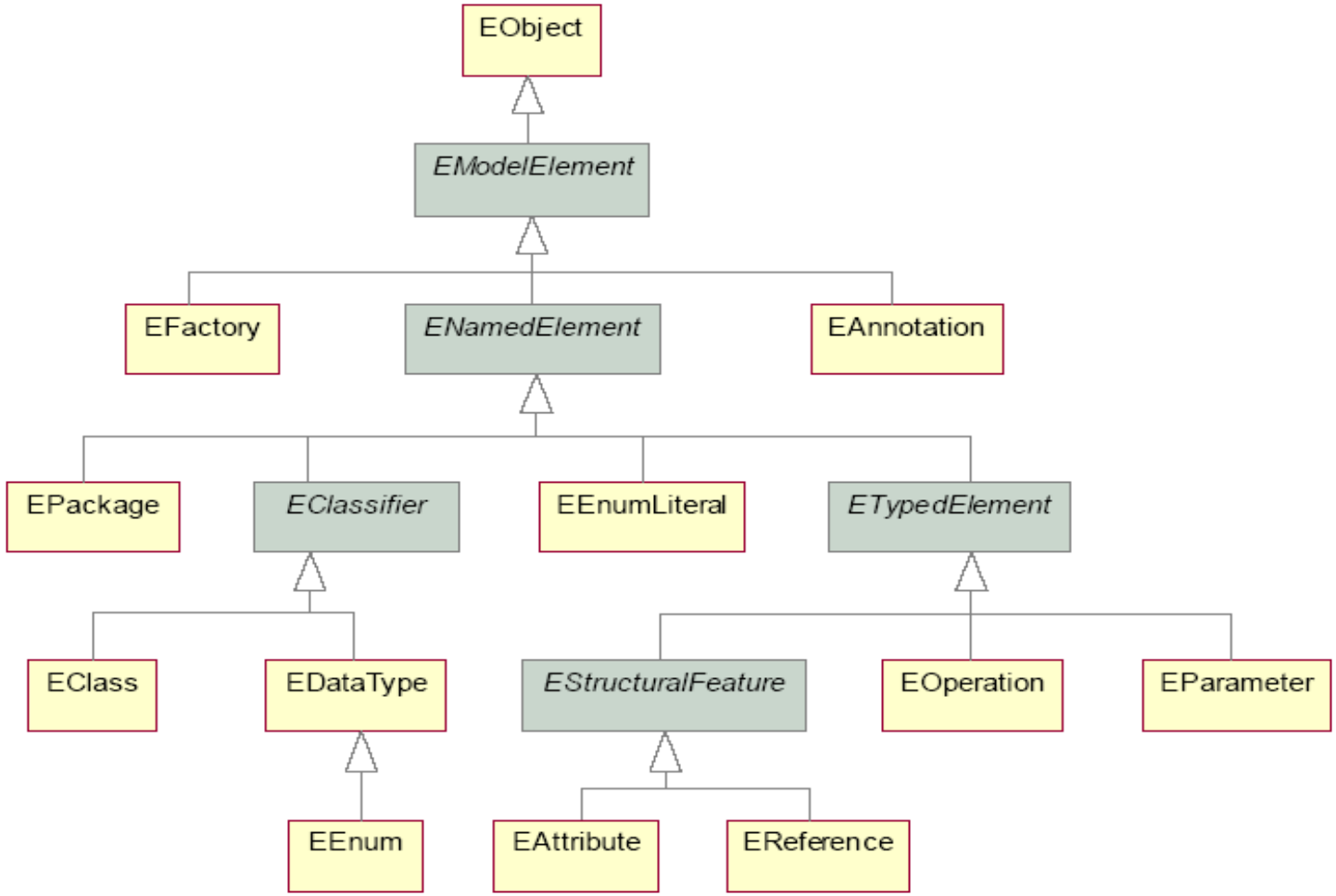
Core Ecore constructs

Class with arbitrary num. of
- **superclasses**
- **associations**
- **attributes**

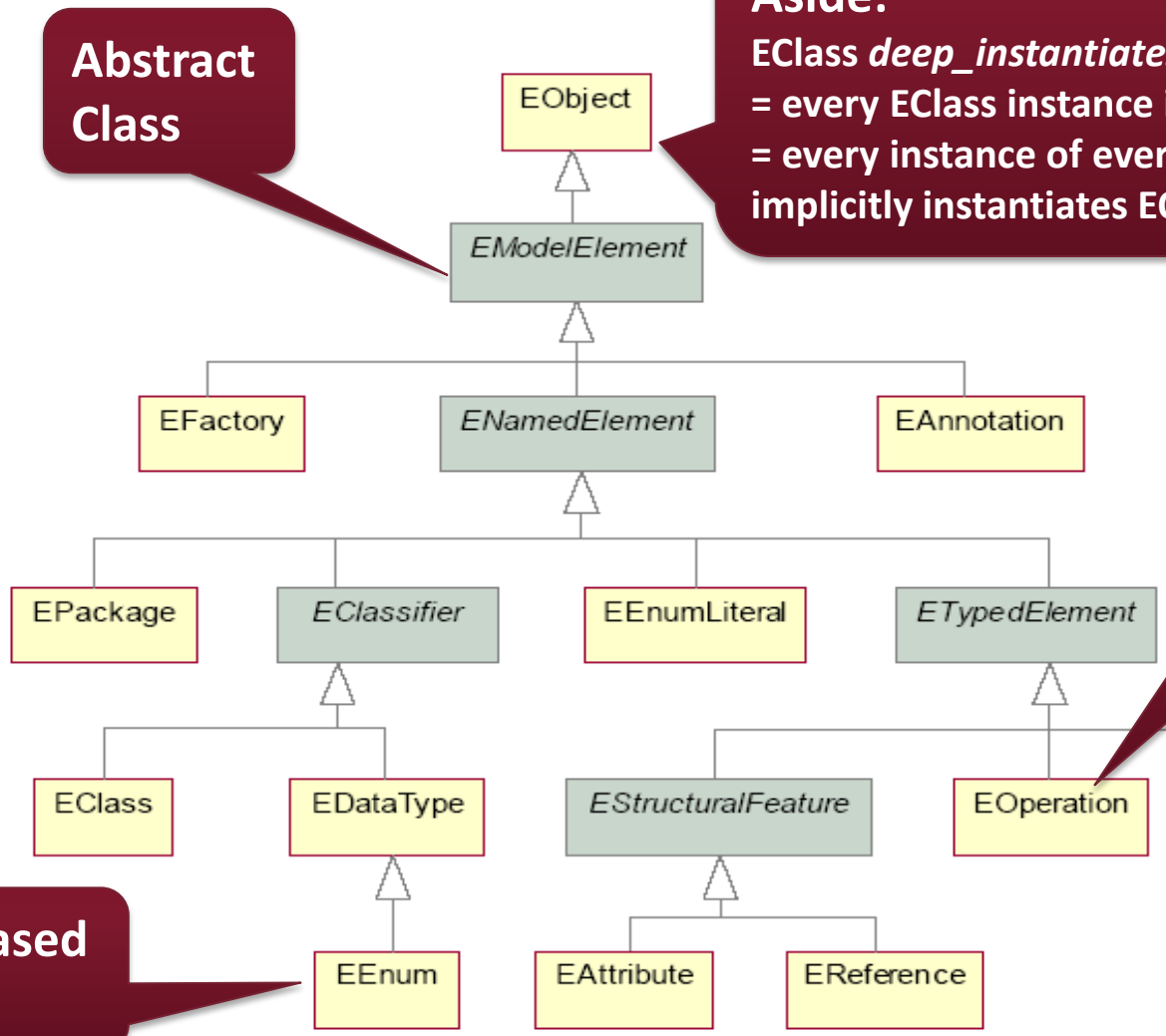


Unidirectional (binary) relation (Association)
- **typed**
- **optional inverse end**
- **multiplicities**

Complete Ecore hiearchy



Complete Ecore hiearchy



Abstract Class

Aside:
EClass *deep_instantiates(2)* EObject
= every EClass instance implicitly subtypes EObject
= every instance of every instance of EClass implicitly instantiates EObject

Methods connected to the EClasses

EMF-based Enums

Parameter for the Eoperation

DEFINING A DSM
…THE EMF WAY

The Classical EMF/Ecore Waterfall

Design domain metamodel
(Questionnaire.ecore)

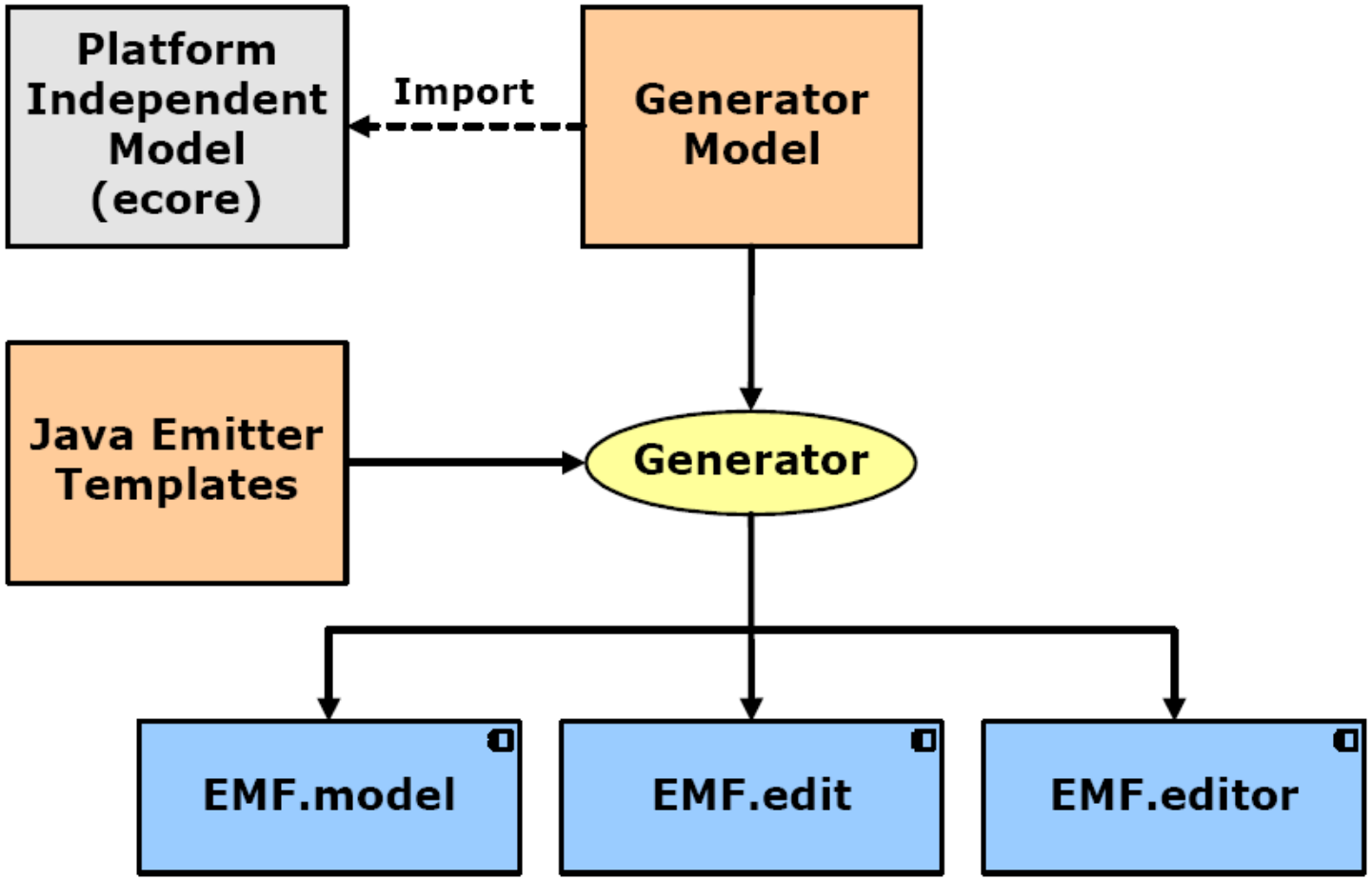
Specify derived features & constraints
(OCL, Epsilon, Viatra Query, Java)

Generate tooling
(Questionnaire.genmodel)

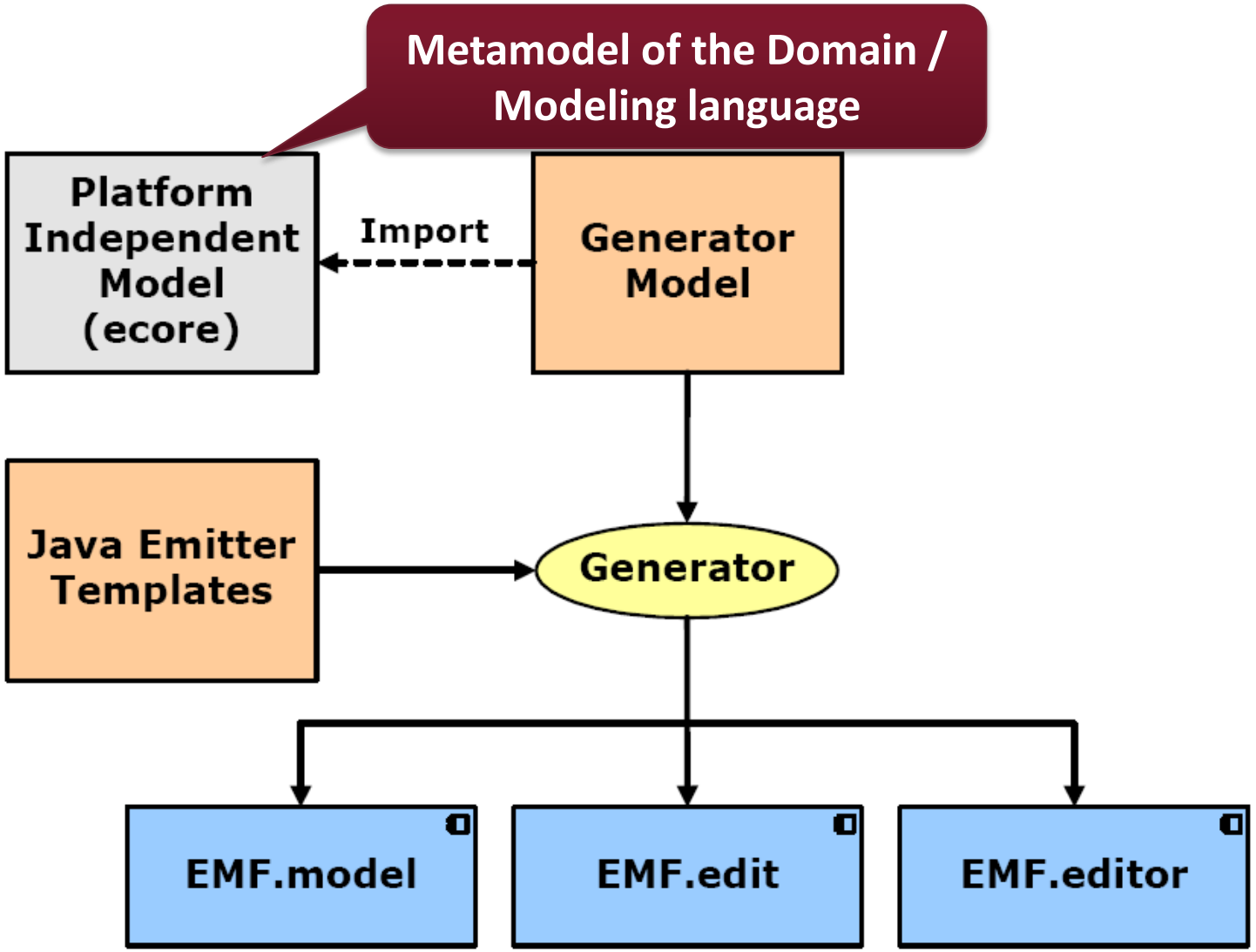
Edit instance models
(Form1.questionnaire)

Validate instance models

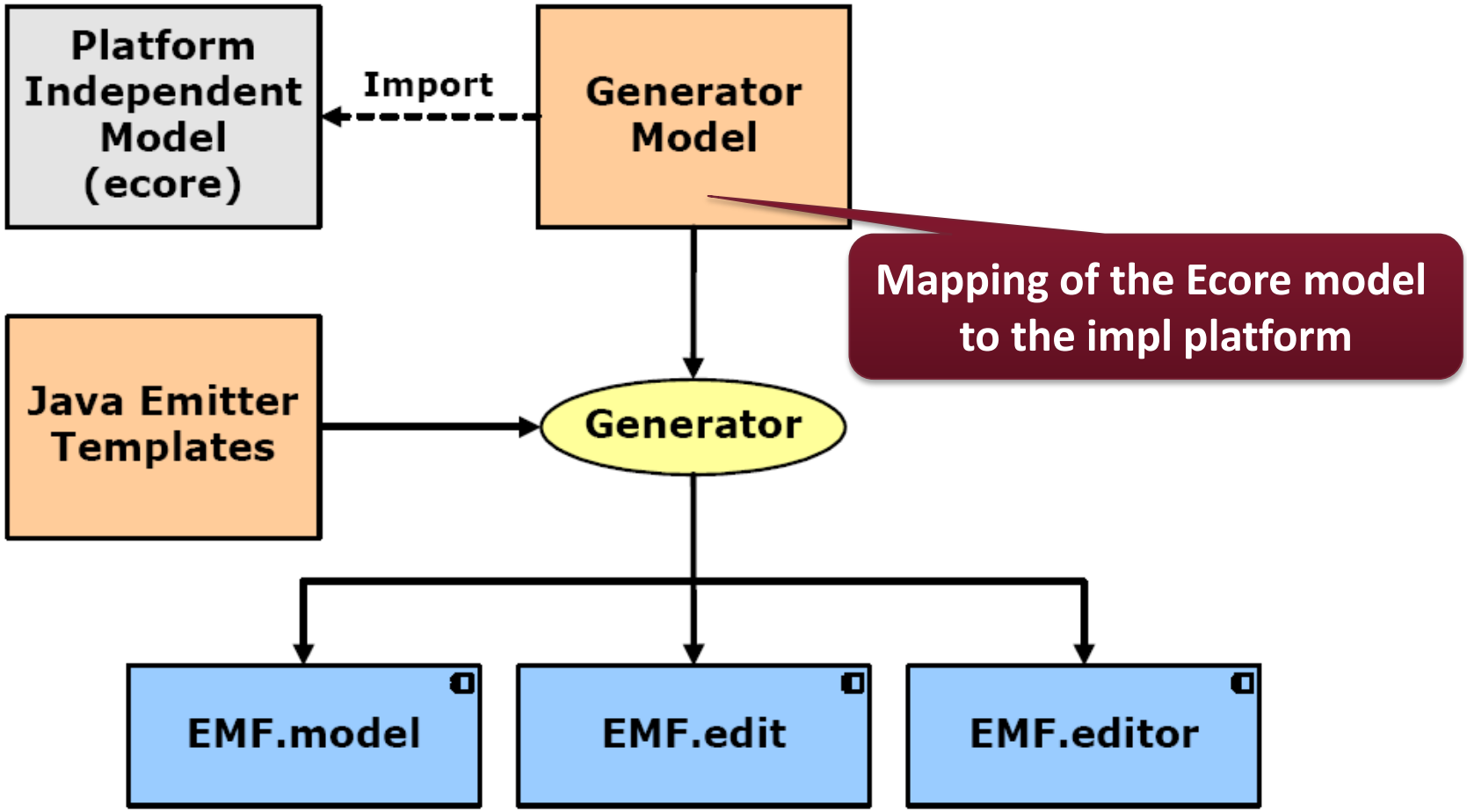
The EMF Toolkit



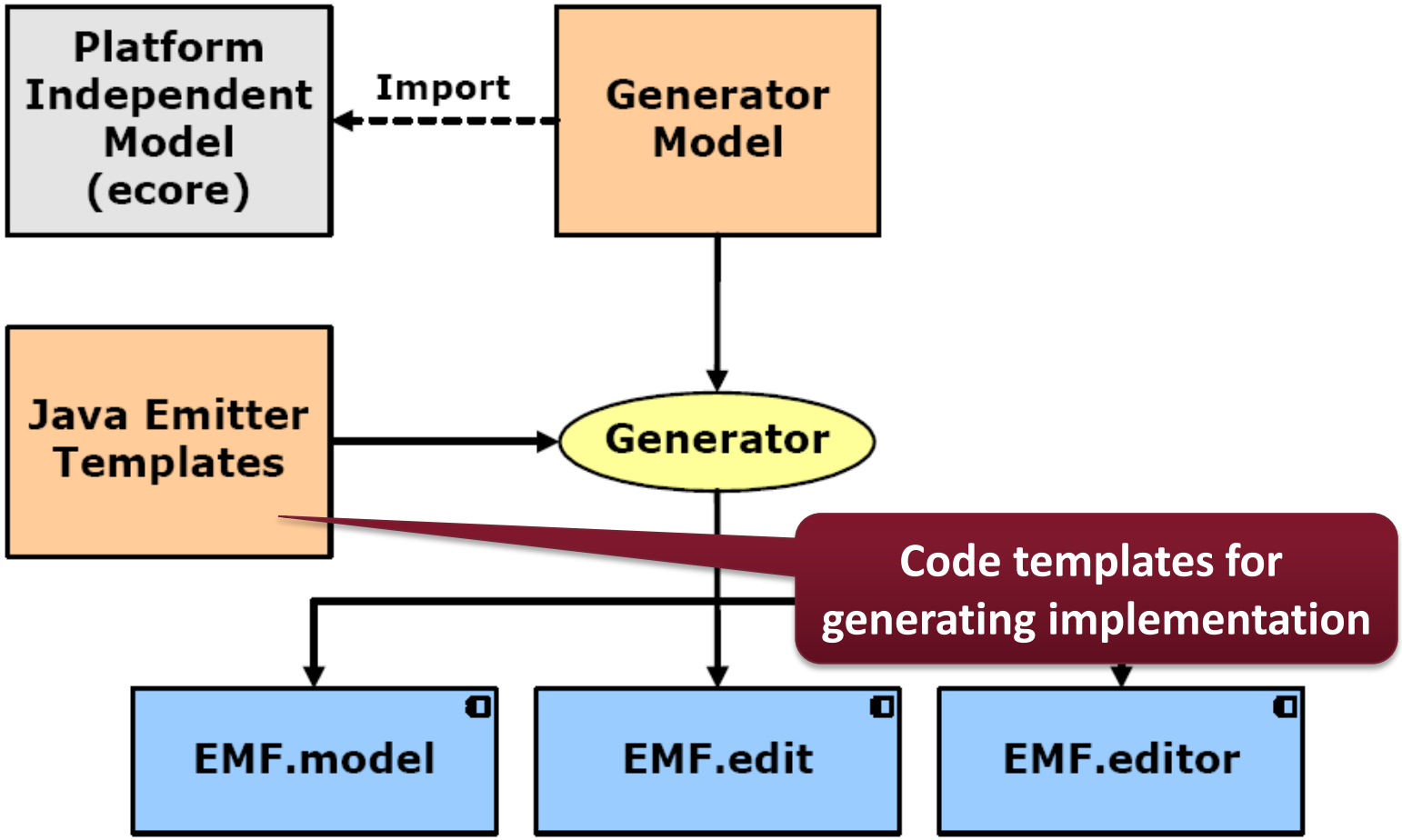
The EMF Toolkit



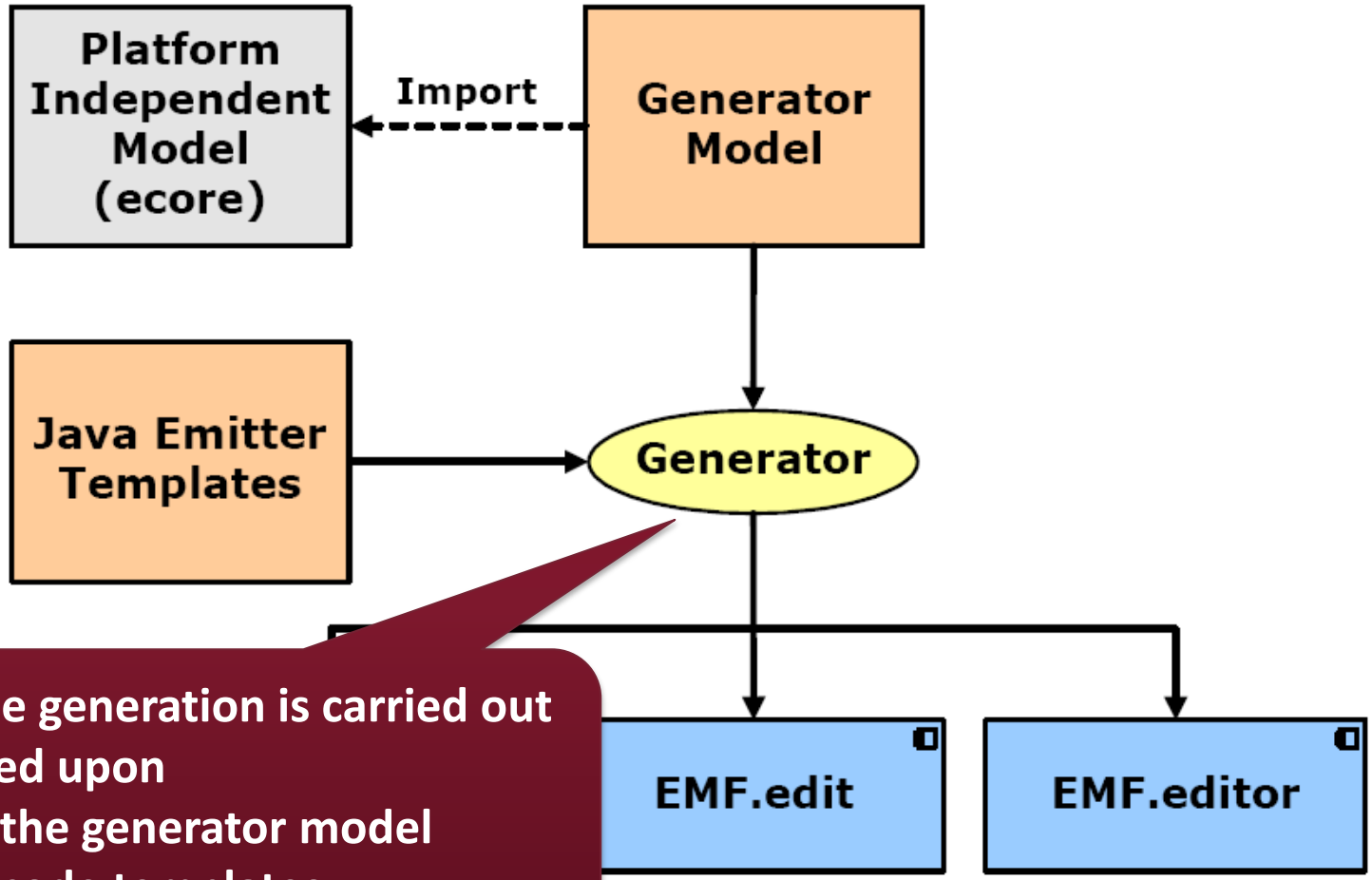
The EMF Toolkit



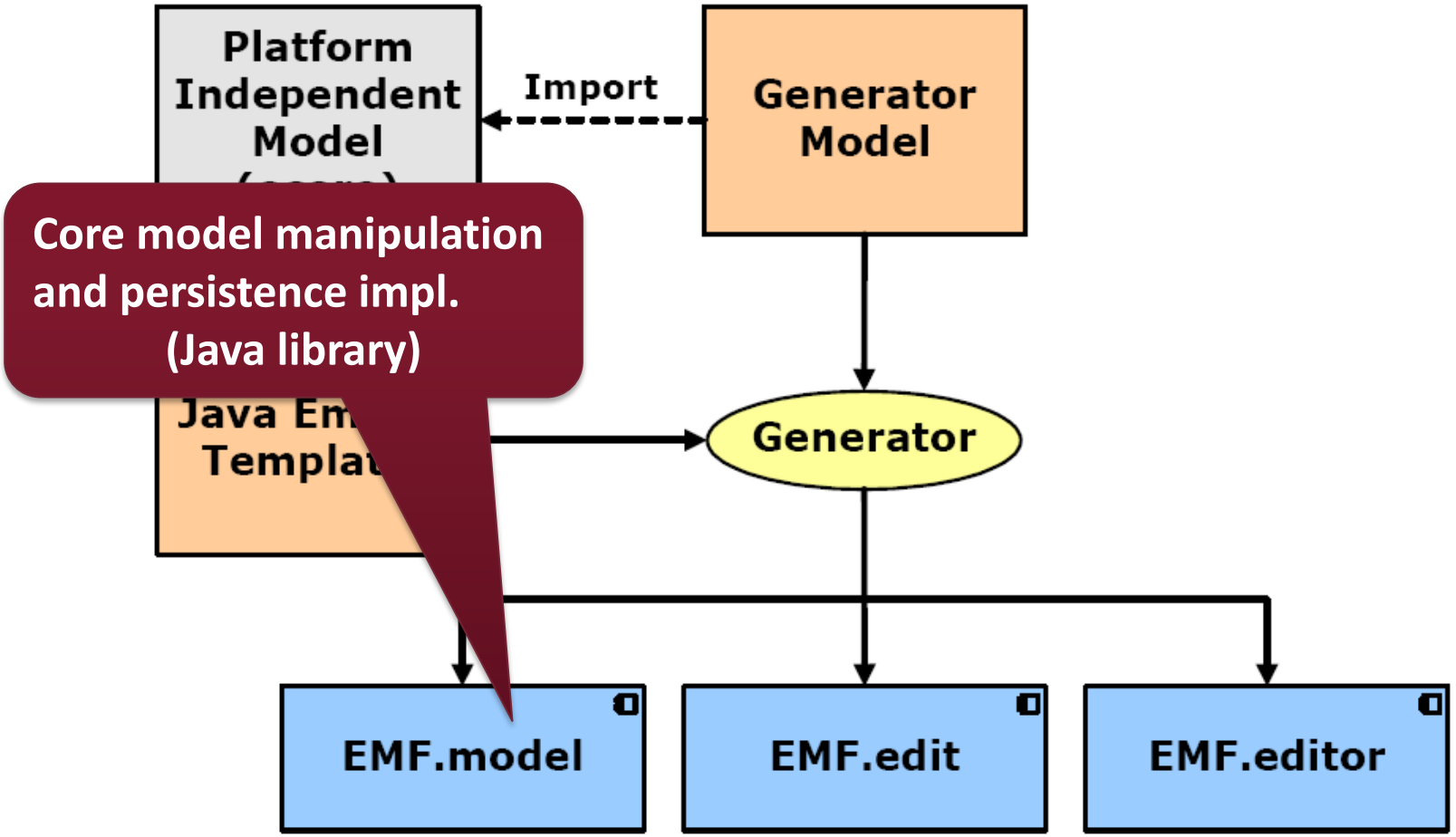
The EMF Toolkit



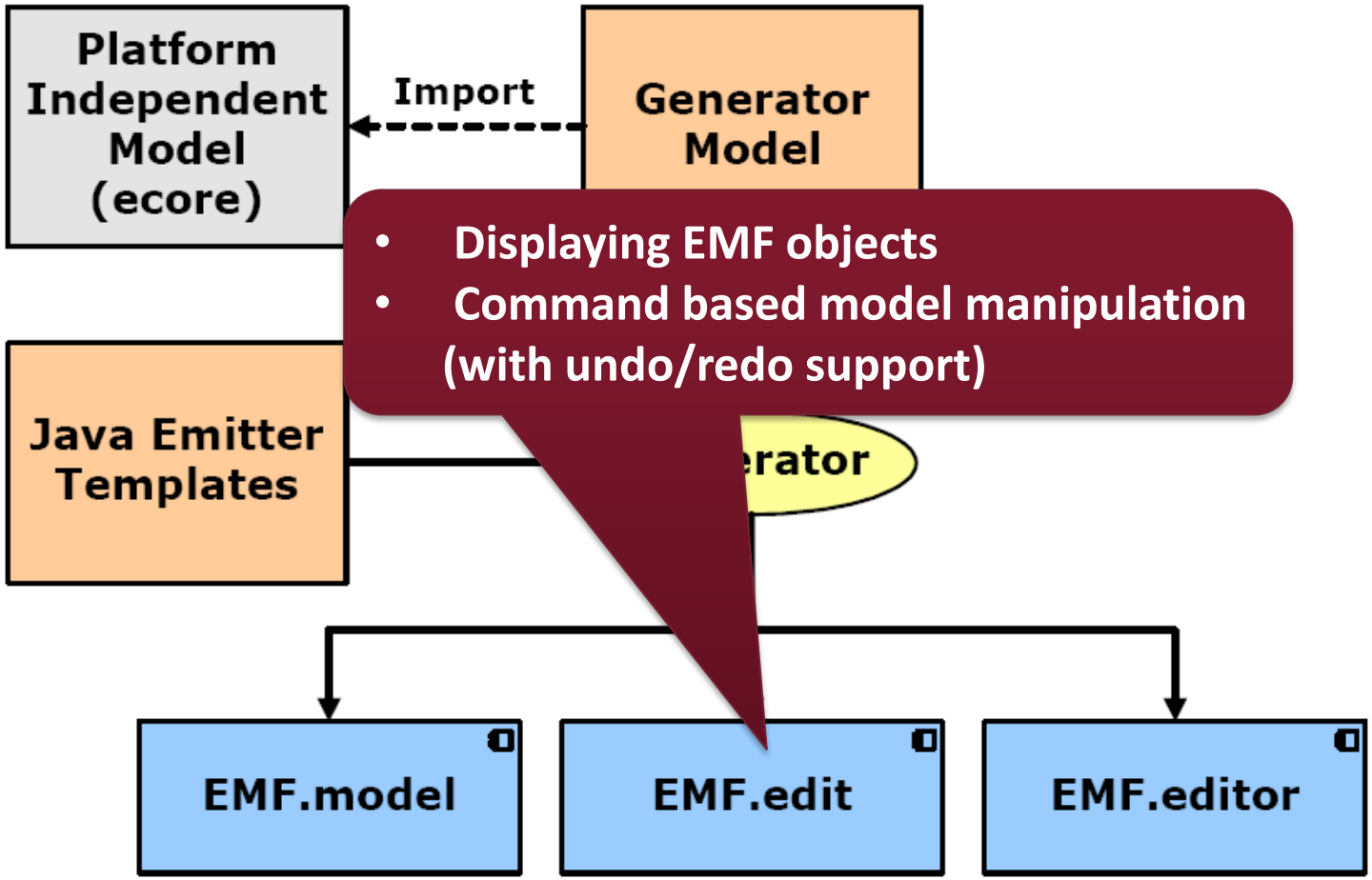
The EMF Toolkit



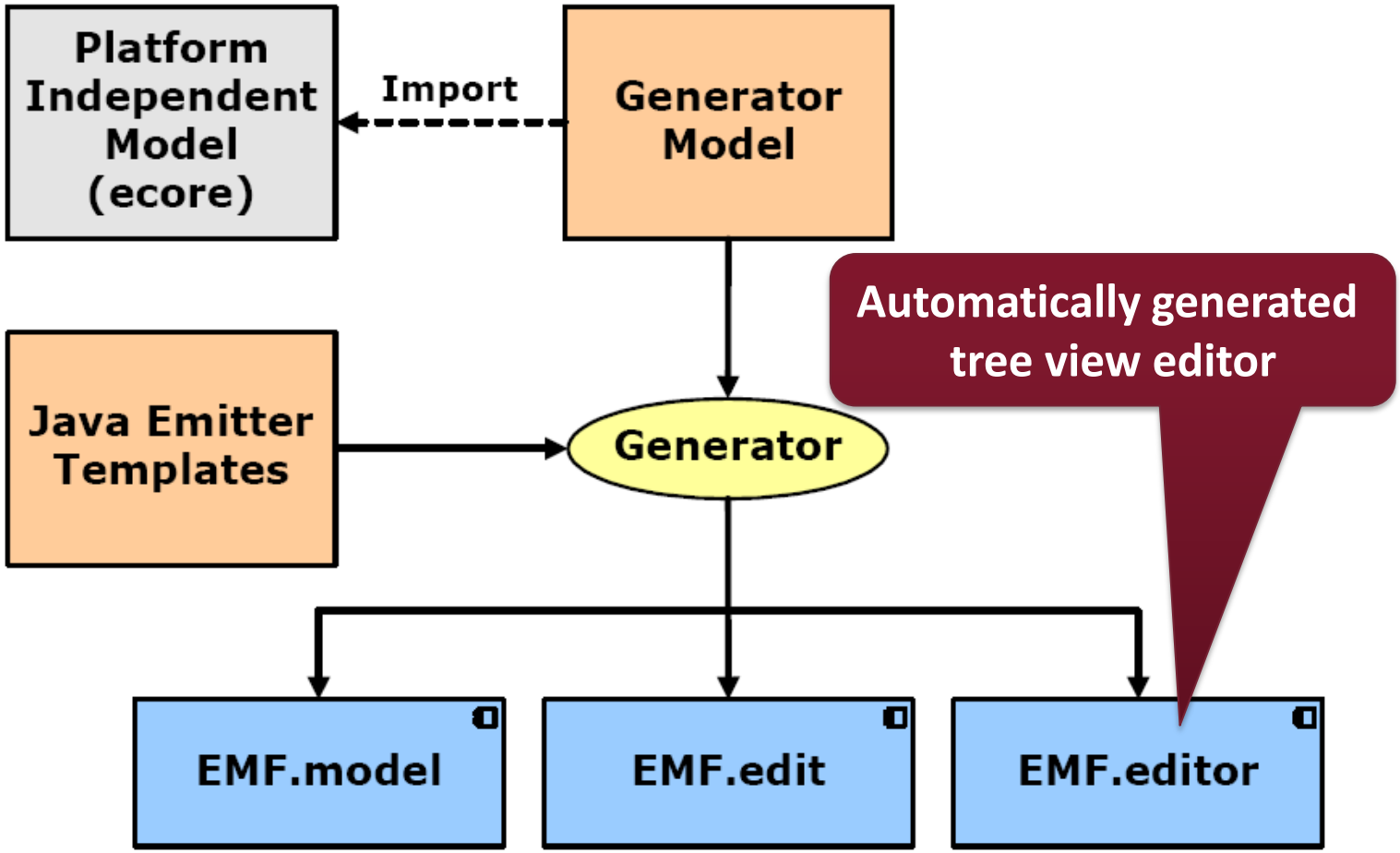
The EMF Toolkit



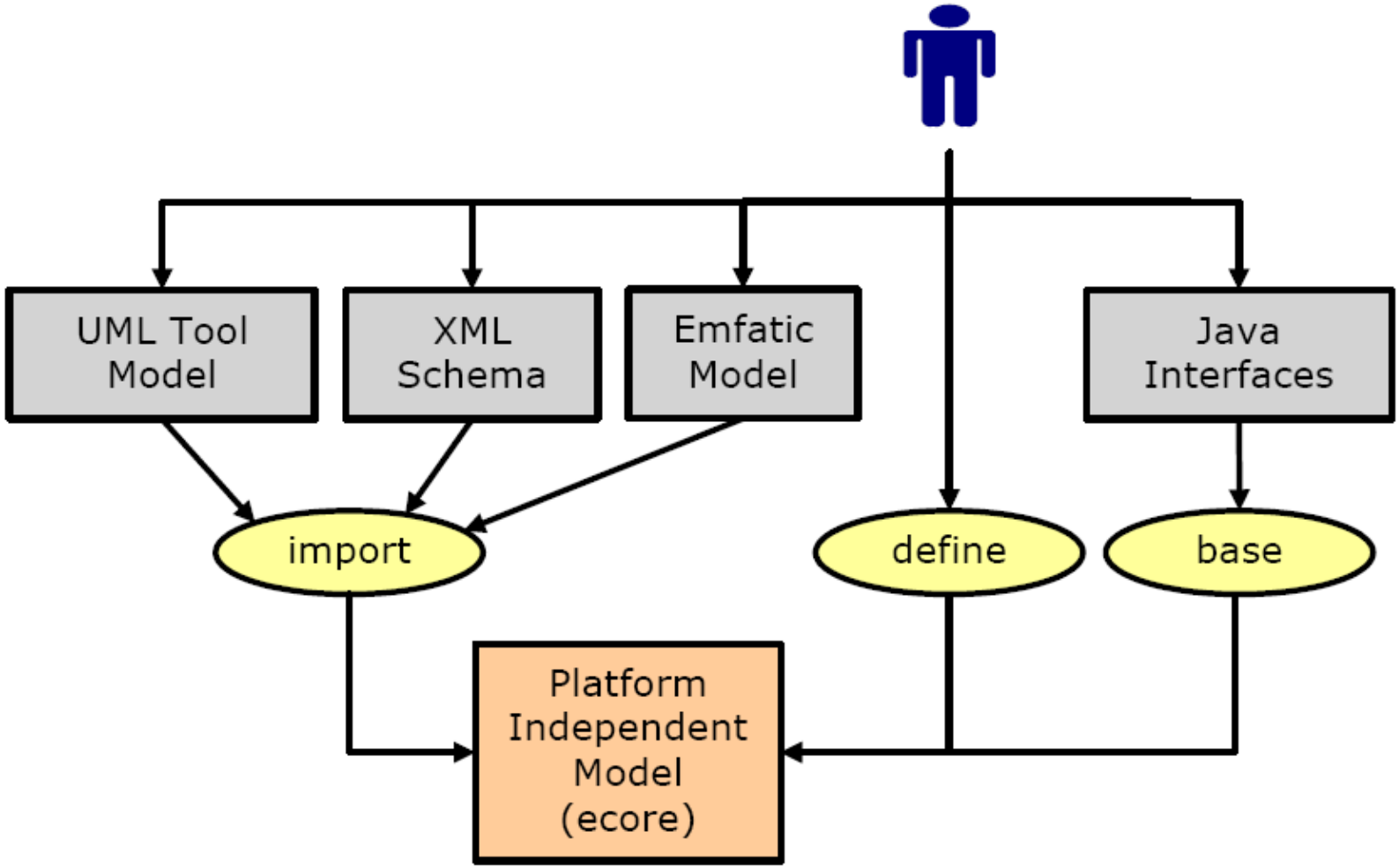
The EMF Toolkit



The EMF Toolkit

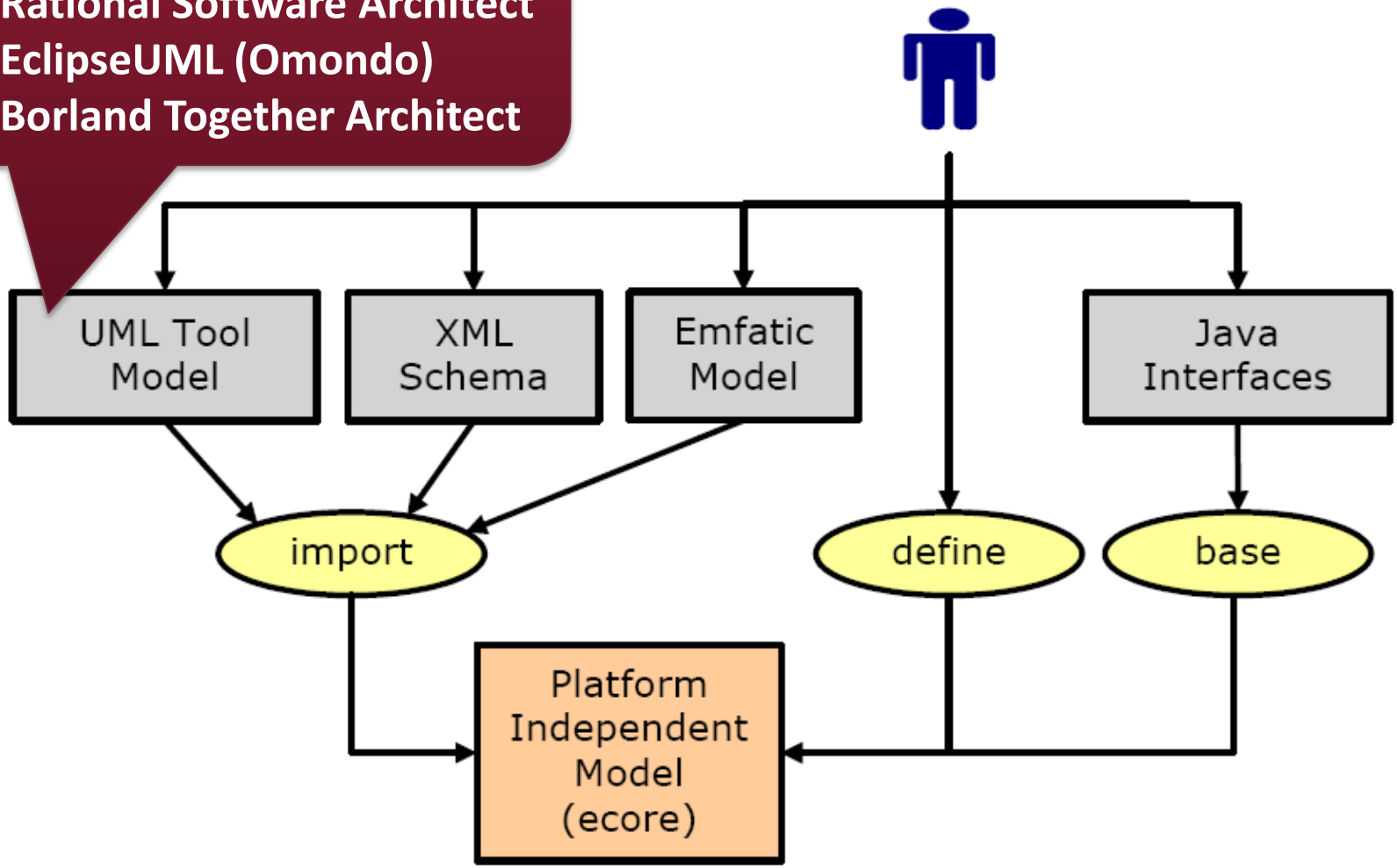


Creation of Ecore metamodels

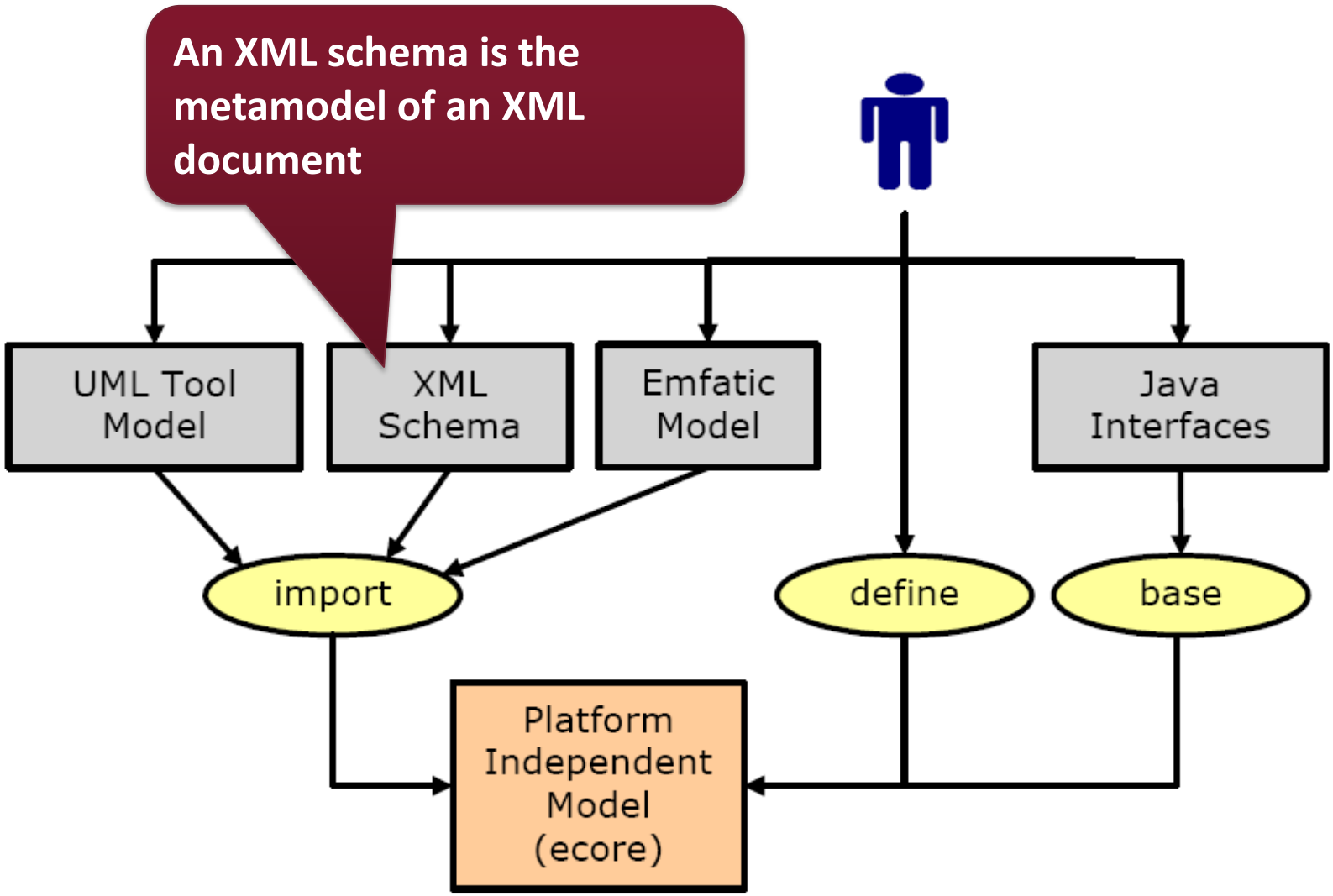


Creation of Ecore metamodels

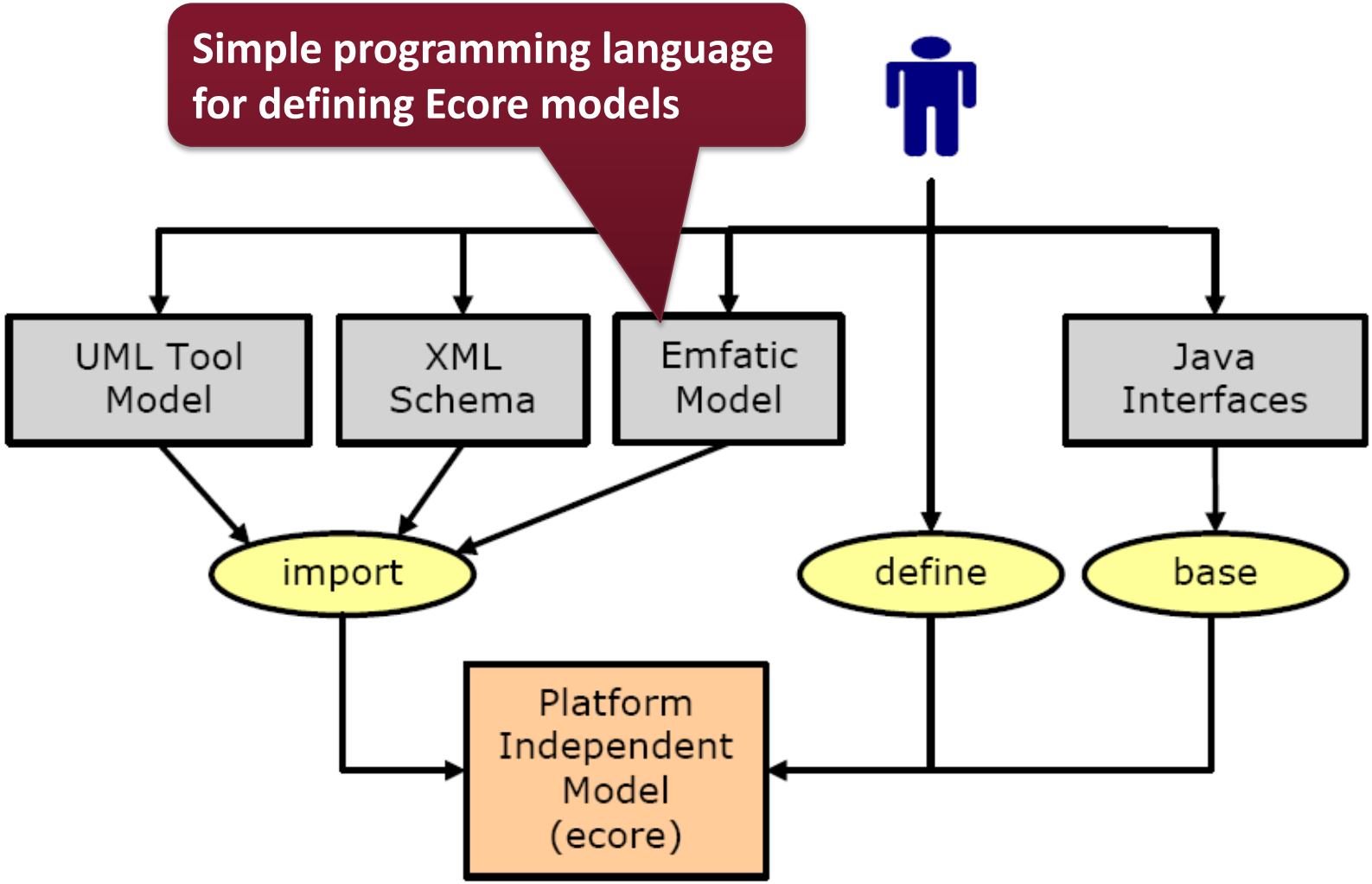
UML class diagram
- **Rational Software Architect**
- **EclipseUML (Omondo)**
- **Borland Together Architect**



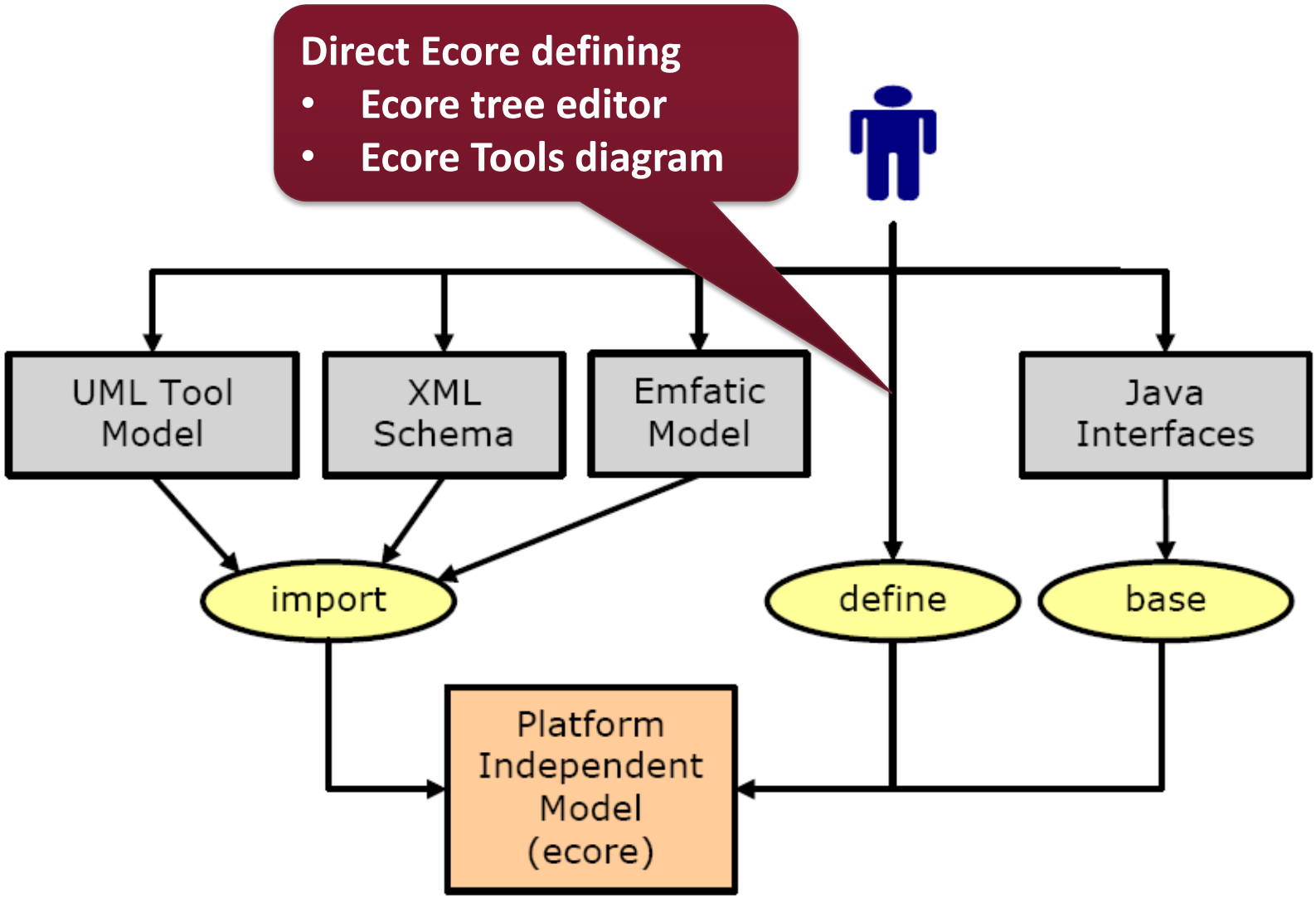
Creation of Ecore metamodels



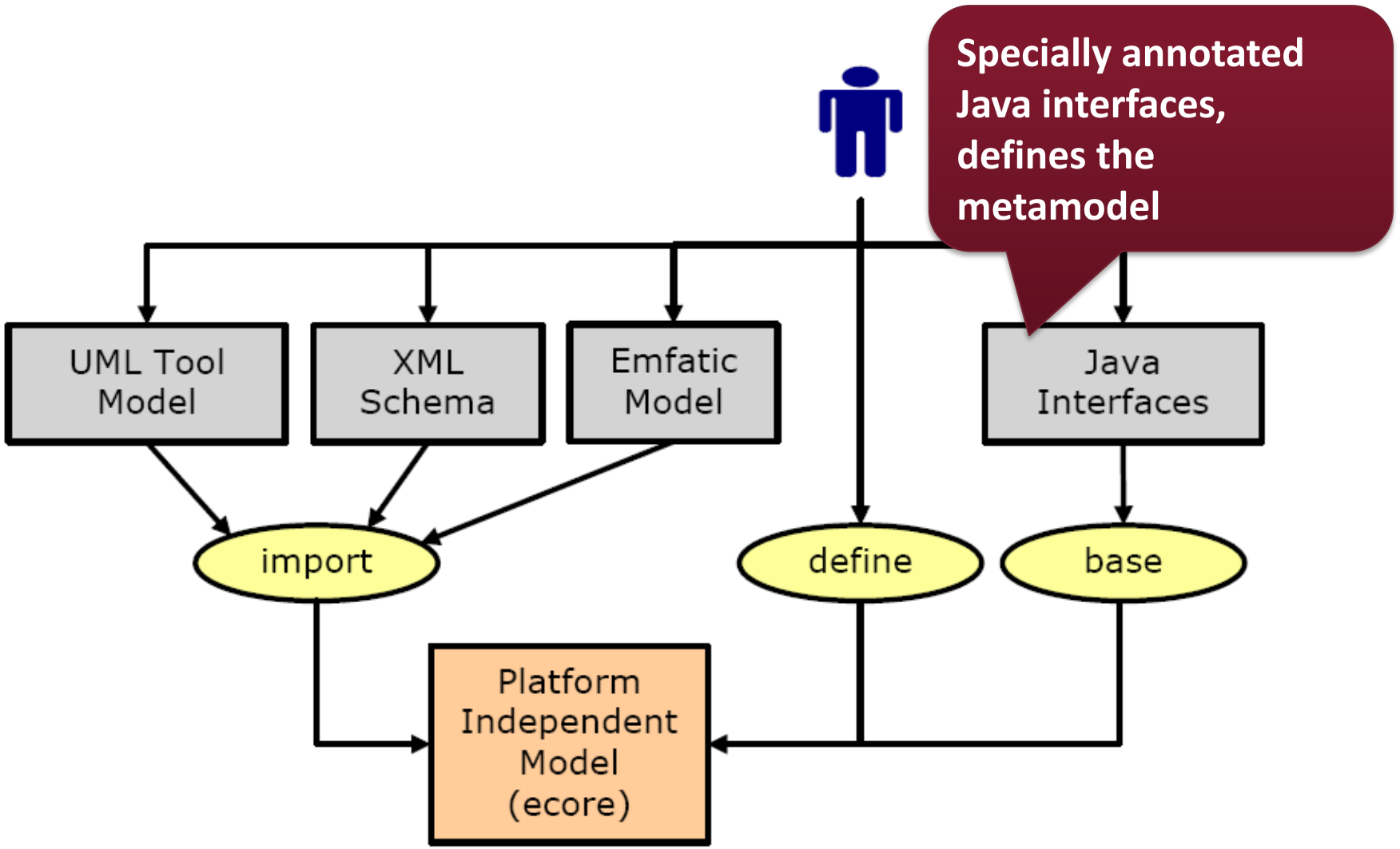
Creation of Ecore metamodels



Creation of Ecore metamodels

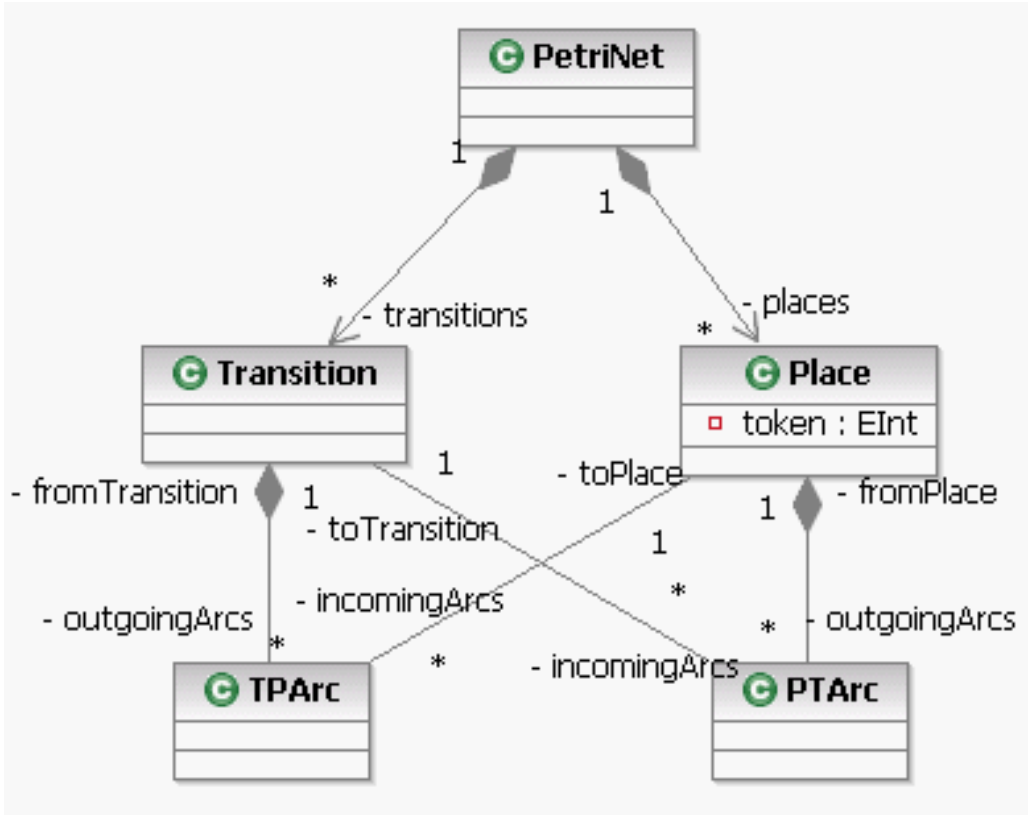


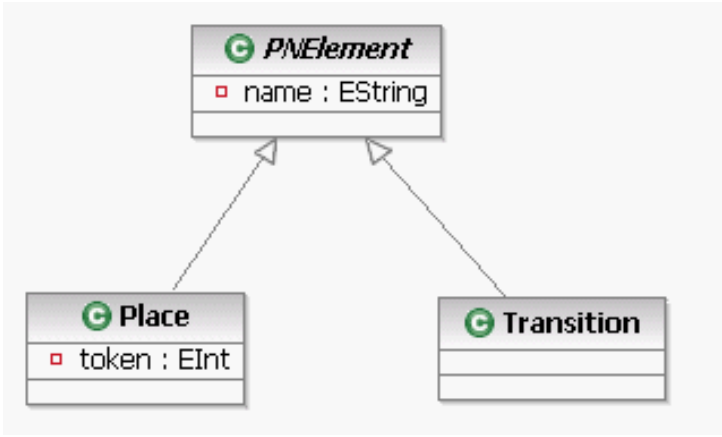
Creation of Ecore metamodels

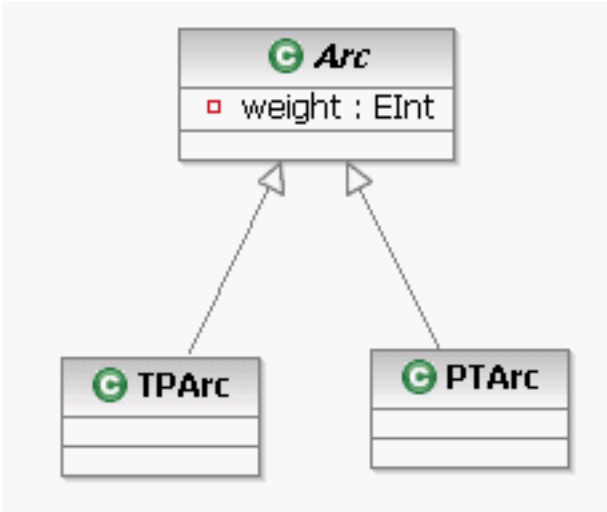


THE PETRI NET EXAMPLE

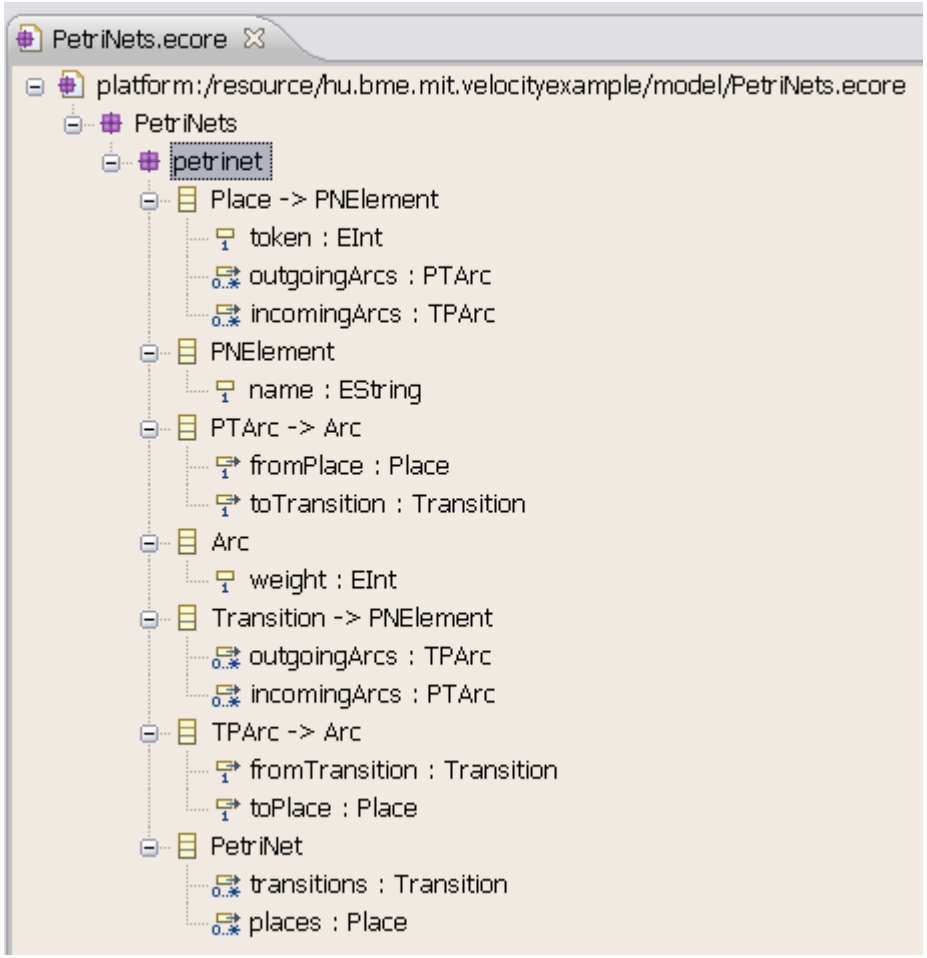
Domain Metamodel: Petri Nets



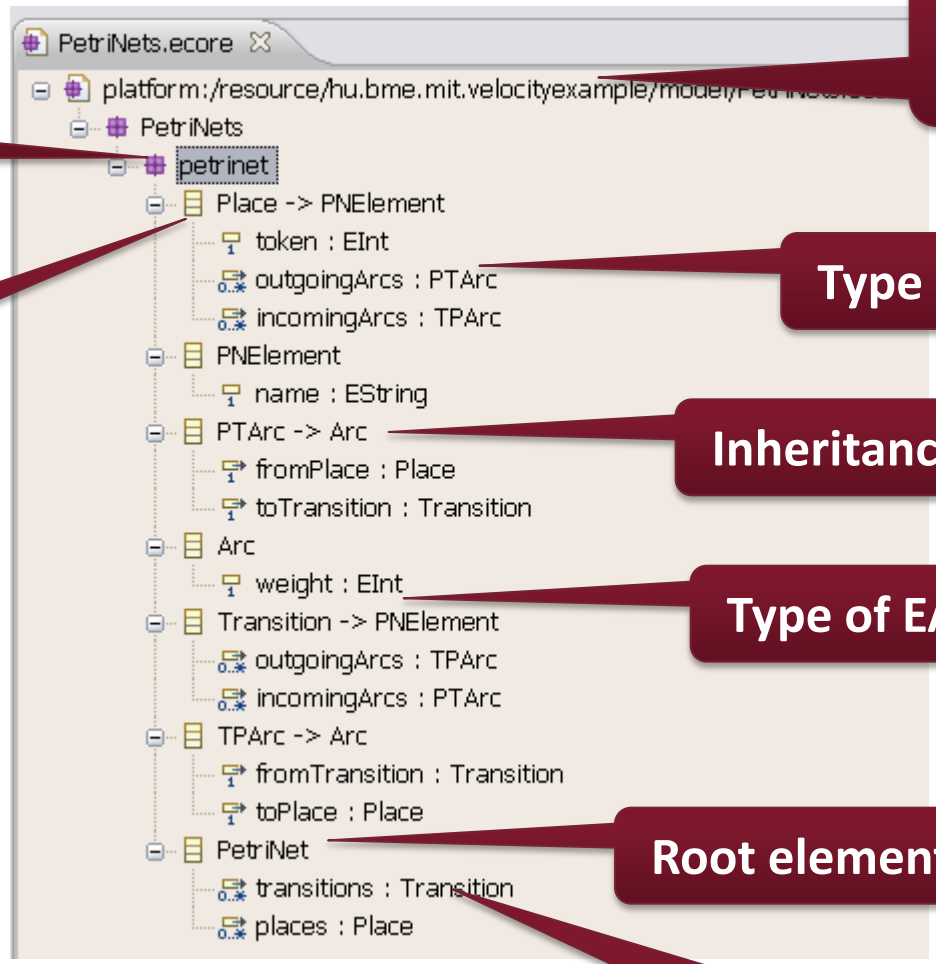




EMF model Ecore representation



EMF model Ecore representation



Path of containing resource

EPackage

EClass

Type of EReference

Inheritance

Type of EAttribute

Root element

Reference to all model elements

Class Definition in PetriNet.ecore

```
<eClassifiers xsi:type="ecore:EClass" name="Place"
    eSuperTypes="#//petrinet/PNElement">

 <eStructuralFeatures xsi:type="ecore:EAttribute" name="token" lowerBound="1"
    eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EInt"/>

 <eStructuralFeatures xsi:type="ecore:EReference" name="outgoingArcs"
    upperBound="-1"
    eType="#//petrinet/PTArc" containment="true"
    eOpposite="#//petrinet/PTArc/fromPlace"/>

 <eStructuralFeatures xsi:type="ecore:EReference" name="incomingArcs"
    upperBound="-1"
    eType="#//petrinet/TPArc" eOpposite="#//petrinet/TPArc/toPlace"/>
</eClassifiers>
```

Class Definition in PetriNet.ecore

```
<eClassifiers xsi:type="ecore:EClass" name="Place"
    eSuperTypes="#//petrinet/PNElement">

 <eStructuralFeatures xsi:type="ecore:EAttribute" name="token" lowerBound="1"
    eType="ecore:EDataType http://www.eclipse.org/emf/2002/Ecore#//EInt"/>

 <eStructuralFeatures xsi:type="ecore:EReference" name="outgoingArcs"
    upperBound="-1"
    eType="#//petrinet/PTArc" containment="true"
    eOpposite="#//petrinet/P

 <eStructuralFeatures xsi:type="ecore:EReference" name="incomingArcs"
    upperBound="-1"
    eType="#//petrinet/TPArc" eOpposite="#//petrinet/TPArc/toPlace"/>
</eClassifiers>
```

Class

Attribute

Reference

Multiplicity

Containment

Type

Opposite End

CODE GENERATION FROM ECORE

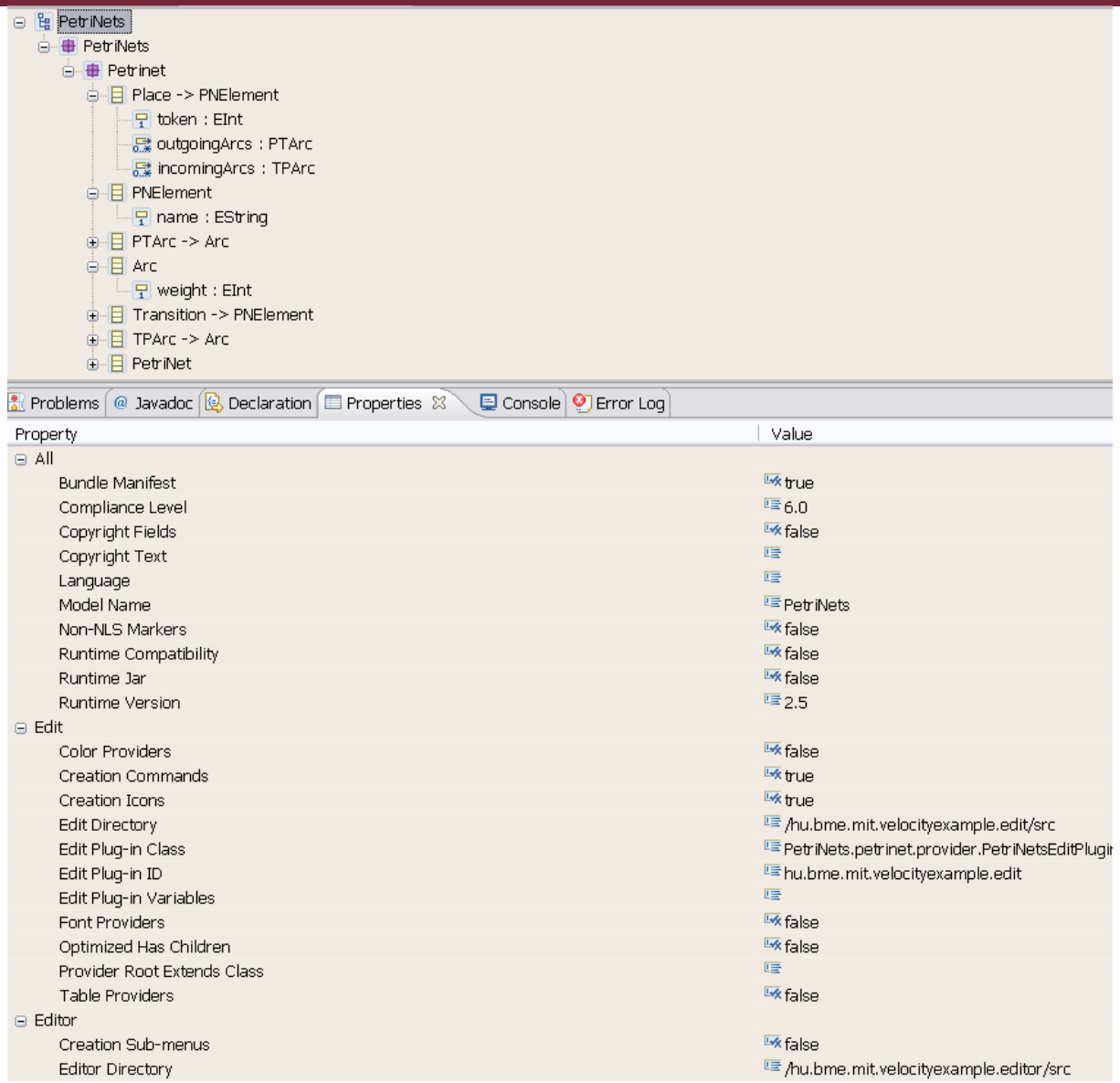
Generator model (.genmodel)

- Goal:
  - Specify the attributes of the code generation
- EMF model
  - Tree Editor
  - Refers to the Ecore model
- Code generation attributes
  - Java version (e.g., use Enums in case of Java 5 and higher)
  - Package/project names
  - …

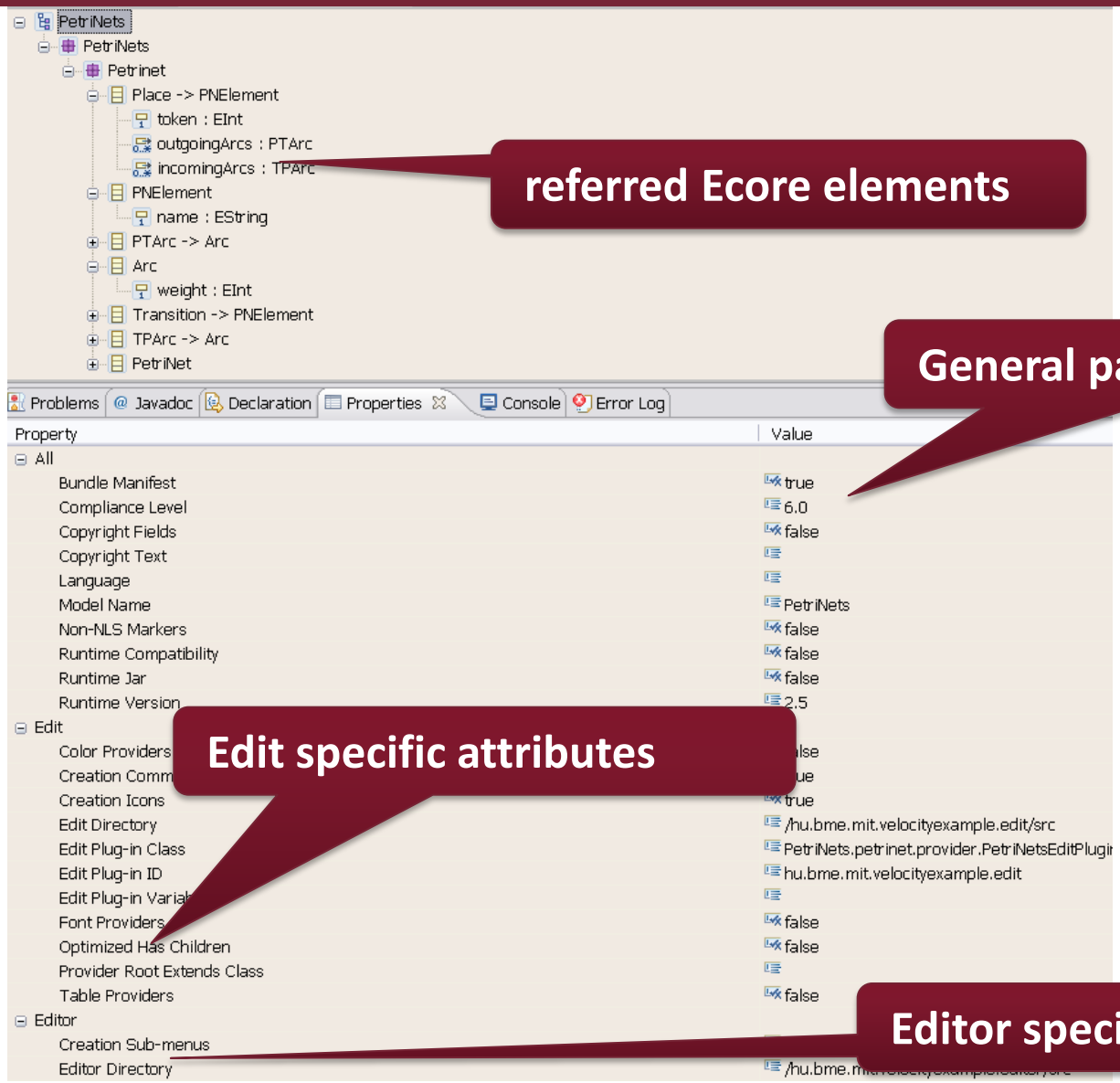
Code Generation from Ecore (.genmodel)

- Ecore model remains pure and independent
- Customizable (wrappers, code formatters, etc.)
- Generated plugins:
  - Model persistency (EMF.model)
  - Model management (EMF.edit)
  - Model editor (EMF.editor)
- Has some limitations
  - What happens when the underlying .ecore changes?

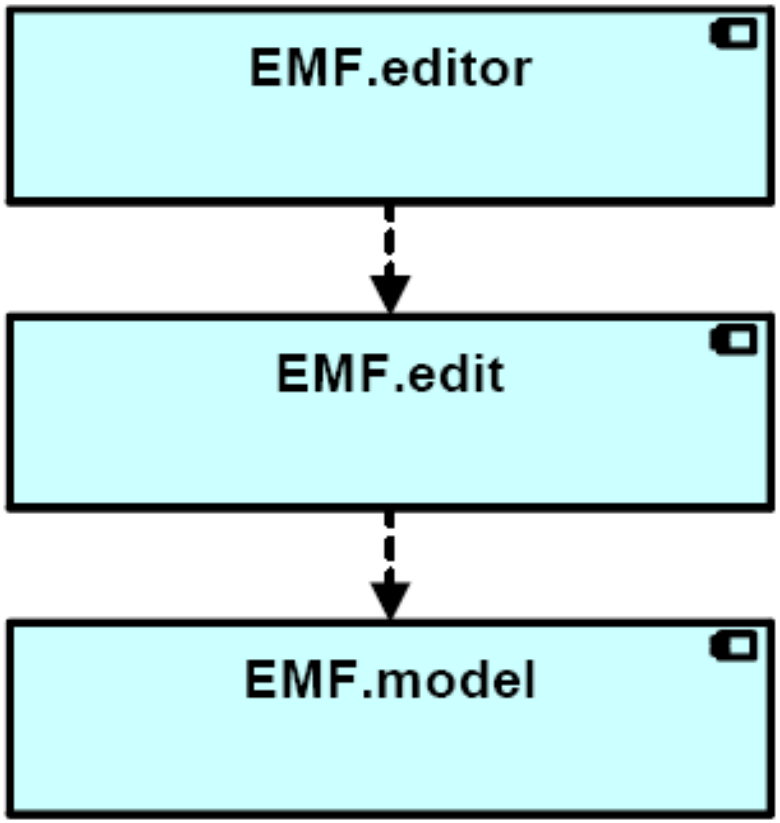
Generator model



Generator model

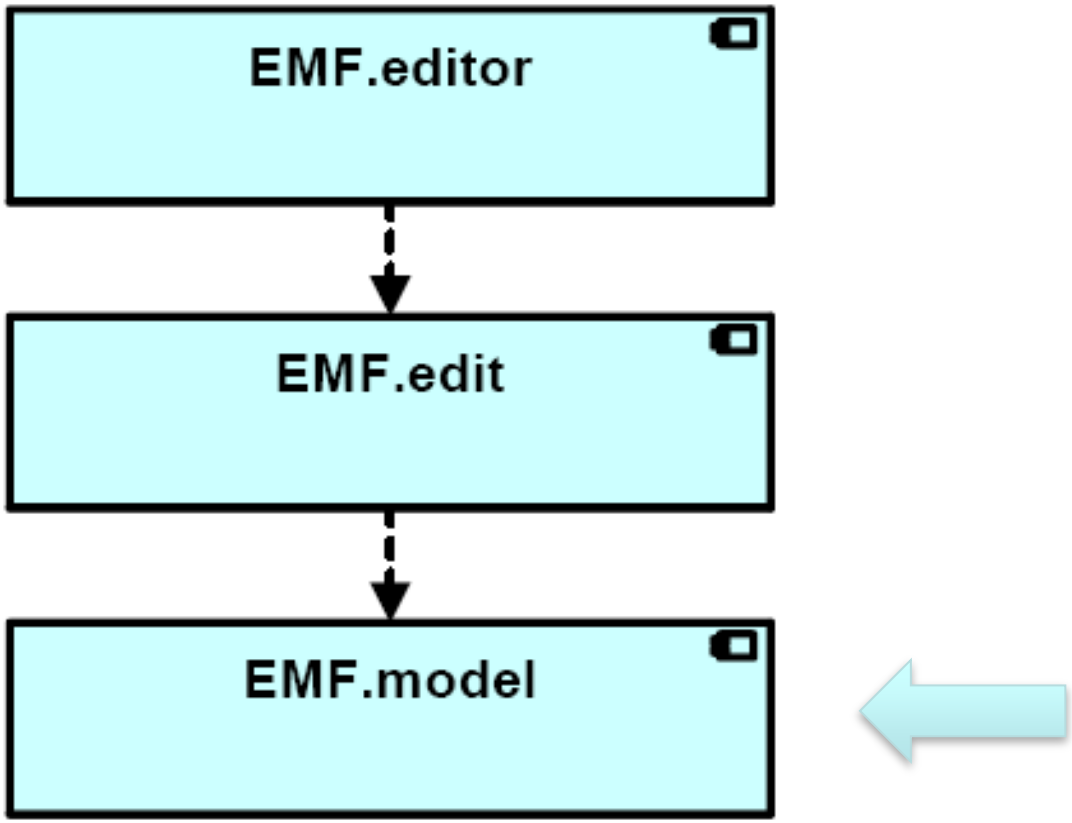


Generated EMF components



- 3. Tree Editor
- 2. Model Manipulation
- 1. Model Persistency

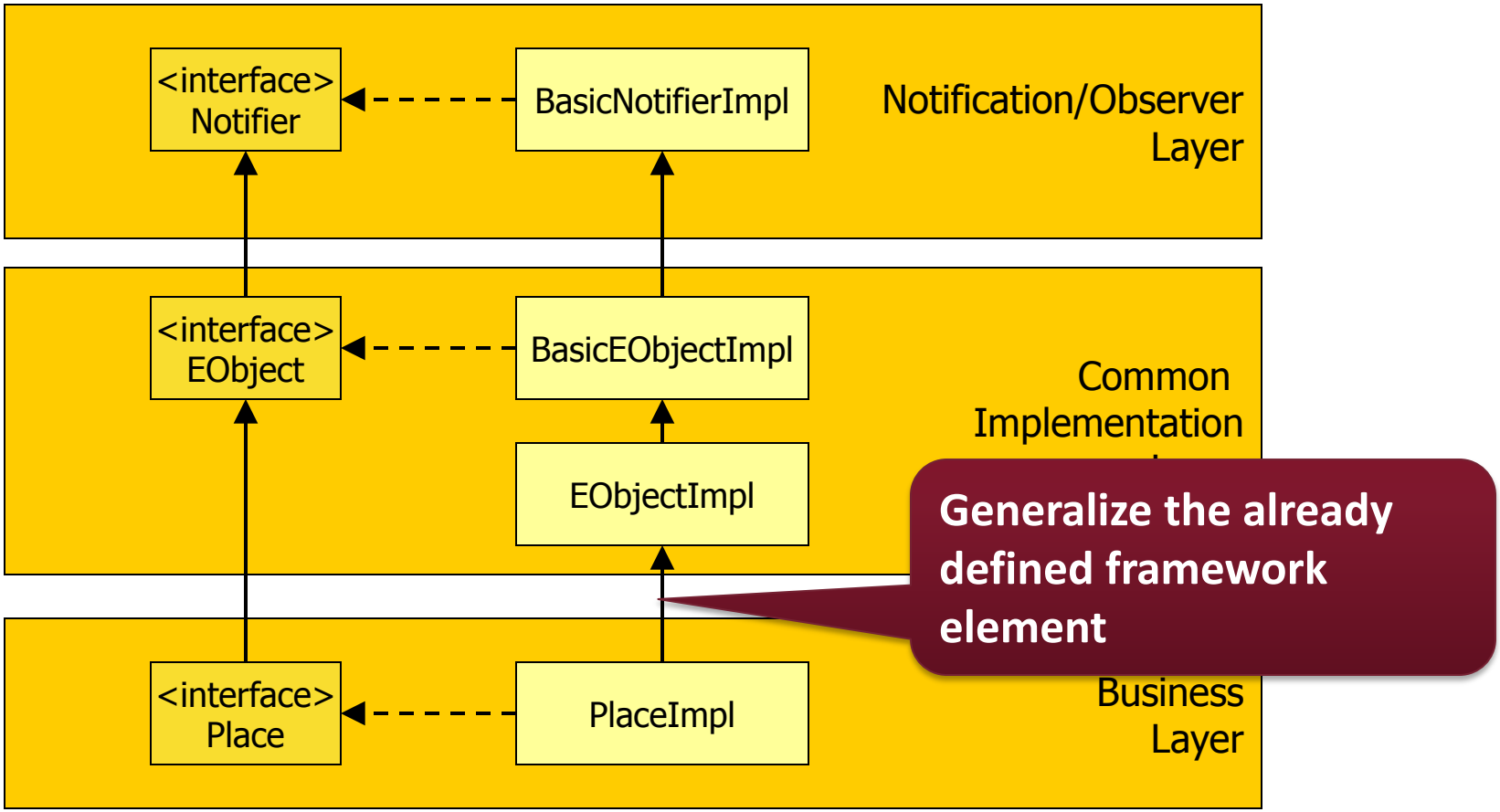
Generated EMF components



EMF.model

- Optimized persistency handling
- Fully featured Java code of the Ecore model
- Specific factories for all packages
- Notification mechanism (observer pattern)
- Possible extension points:
  - Advanced editor
  - Own file format with parser

EClass implementation



Auto-Generated Interface

```
 * @model
 * @generated
 */
public interface Place extends PNElement {
    /**
     * @model required="true"
     * @generated
     */
    int getToken();

    /**
     * @see #getToken()
     * @generated
     */
    void setToken(int value);

    /**
     * @model opposite="fromPlace" containment="true"
     * @generated
     */
    EList<PTArc> getOutgoingArcs();

    /**
     * @model opposite="toPlace"
     * @generated
     */
    EList<TPArc> getIncomingArcs();

} // Place
```

ESuperClass

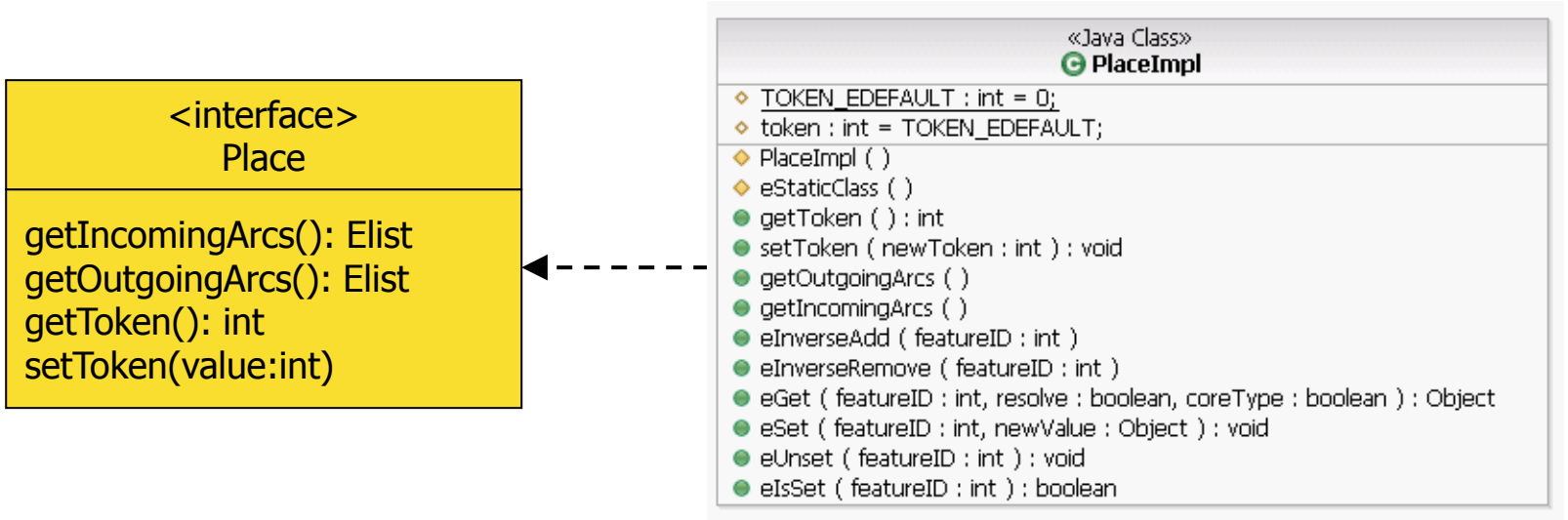
EMF specific „annotations”

Getters/Setters for attributes

No setter when multiplicity > 1 (use add/remove instead)

EList: EMF list interface (~10 implementations

EObject API



- Every class contains framework-specific methods:
  - Reflective get/set (eGet, eSet)
  - Consistent manipulation (eInverseRemove)
  - Notifications for feature changes (very useful e.g. in GUI!)
- Inherited from common supertype EObject
  - see deep instantiation earlier

EOperation Implementation

```
public class XImpl extends EObjectImpl implements X {

    /**
     * @generated NOT
     */
    void f() {
      // Provide the implementation
    }
}
```

- Represents the frame of a Java method
- Present in both the interface and implementing class
- Important:
  - Have to change the generated annotation to NOT
    - …so that next code generation phase does not overwrite it
  - Have to implement the method manually

Client Programming with EMF

```
Place p1 = PetrinetFactory.eINSTANCE.createPlace();
p1.setName("p1");
Place p2 = PetrinetFactory.eINSTANCE.createPlace();
p2.setName("p2");
Transition t1 = PetrinetFactory.eINSTANCE.createTransition();
t1.setName("t1");

// Inverse direction (p1.outgoingArcs) is set automatically
PTArc a0 = PetrinetFactory.eINSTANCE.createPTArc();
a0.setFromPlace(p1);
a0.setToTransition(t1);
TPArc a1 = PetrinetFactory.eINSTANCE.createTPArc();
a1.setToPlace(p2);
a1.setFromTransition(t1);
```

Create a place

Create a transition

Create a PT arc

Set source of PT arc

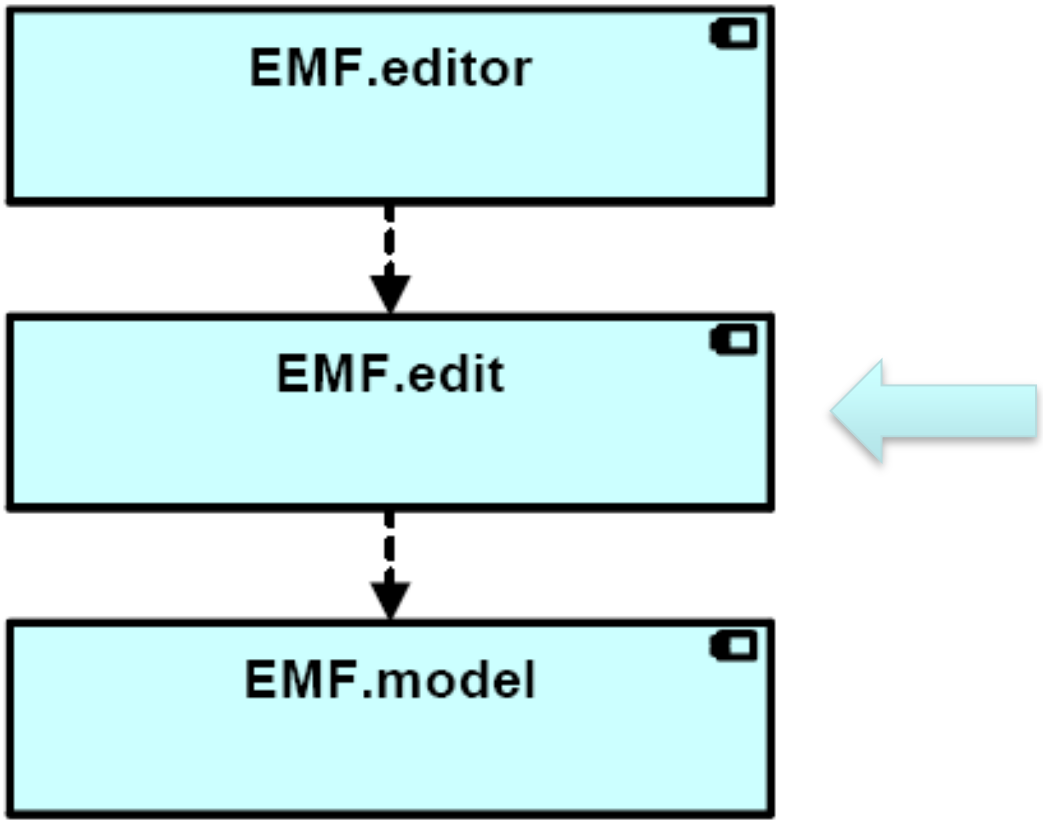
Set target of PT arc

Advanced client programming: Reflective Ecore API

The org.eclipse.emf.ecore.util Package

- Contains utility classes and interfaces:
  - *ECoreEContentAdapter*: maintains itself as a notification adapter for a whole containment (sub)tree
  - *UsageCrossReferencer*: finds each ModelElement pointing to the corresponding EObject
  - *ContentTreeIterator*: An iterator over the tree contents of a collection of EObjects
  - *Copier*: deep copy of EObject Elements and EReferences
  - Etc.
- (This is not generated but a generic component)

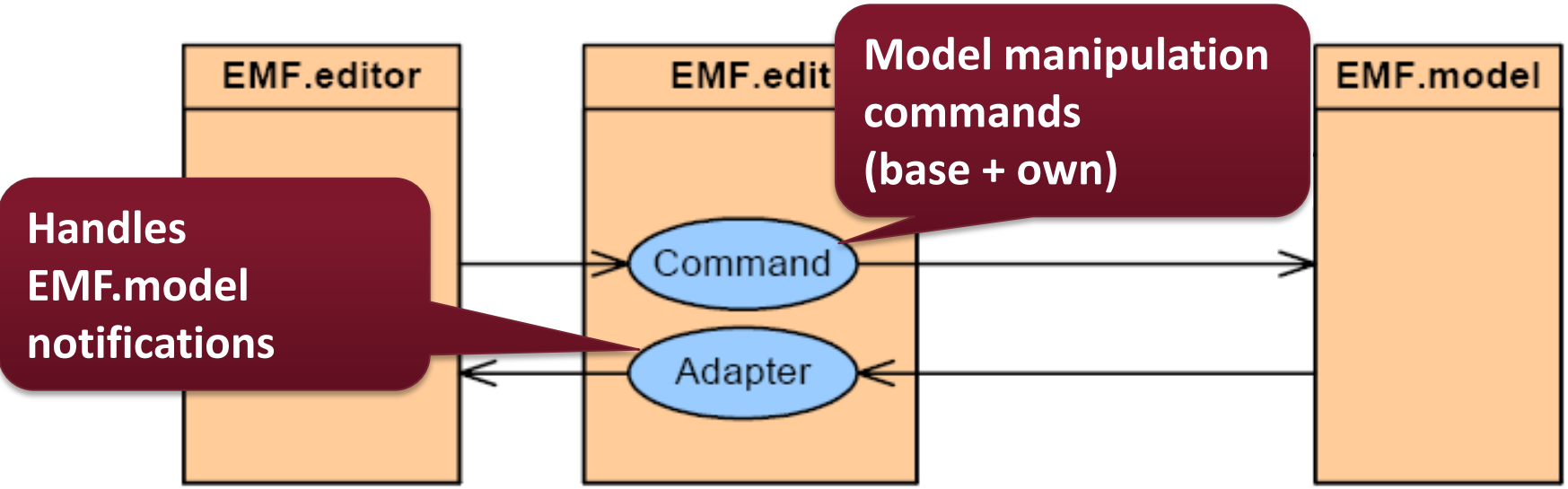
Generated EMF components



EMF.Edit

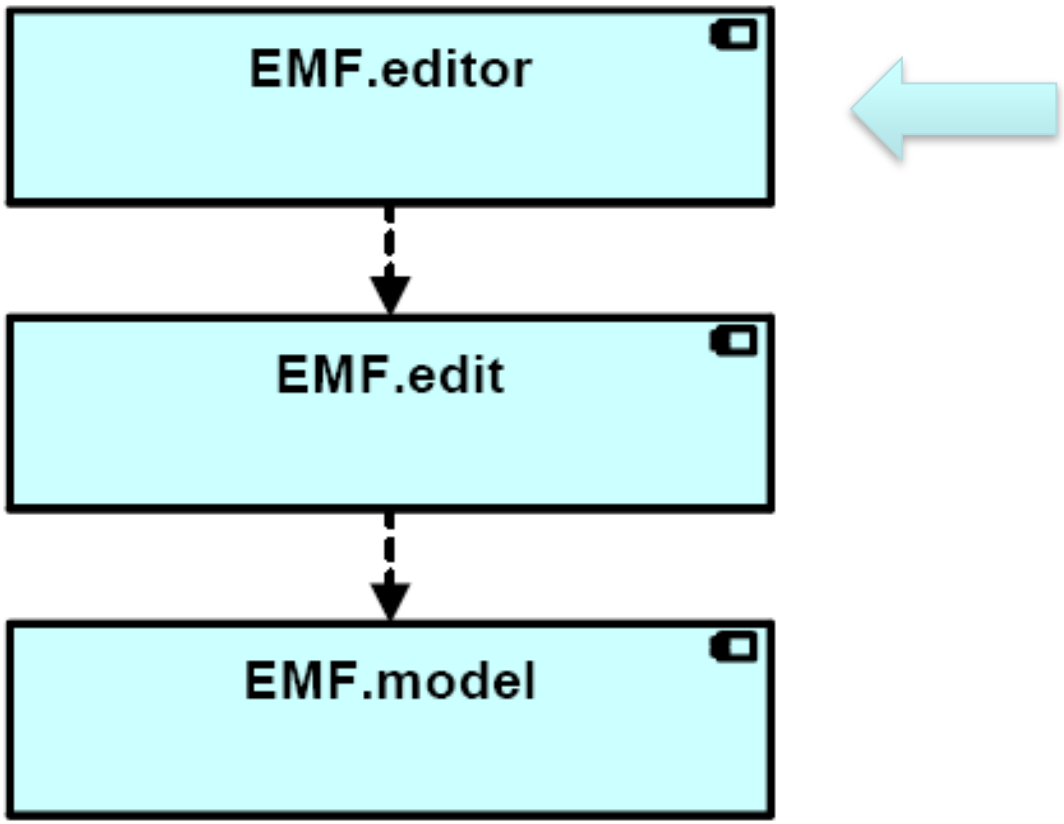
- Separates the GUI and the model
- Generator pattern:
  - Provider class for each model element
  - Base class: ItemProvider
  - Forward EMF model change notifications to the viewer
- Provides:
  - Element text
  - Icon
  - Description of features in EClass

EMF.Edit

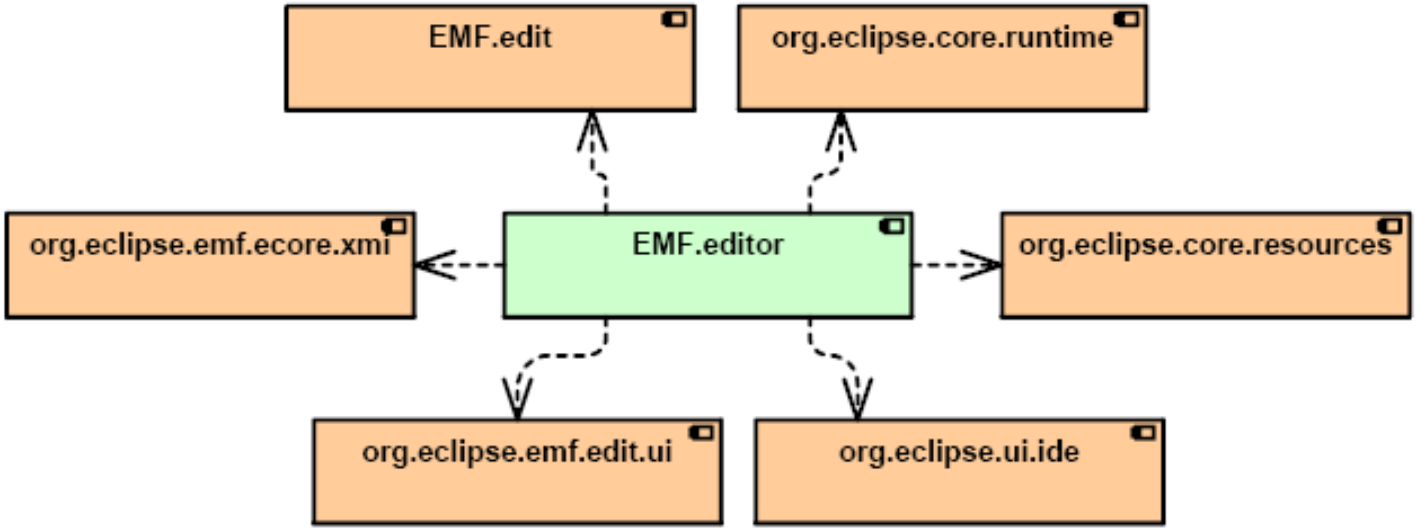


- Converts model notifications to GUI notifications
- Model manipulation through commands
  - Possible alternative to direct setters
  - Undoable, redoable
  - ItemProvider.create*Add*Command(…) etc.

Generated EMF components

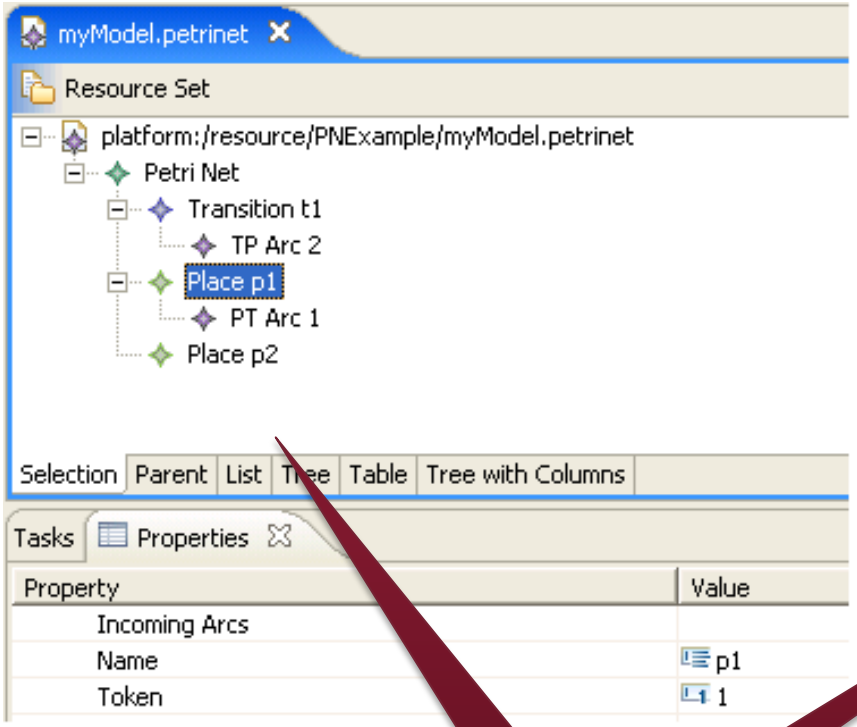


EMF.Editor



- EMF.Editor generates the SWT/JFace for the graphical editor
- Generates:
  - Tree editor
  - Wizards
  - Menus
  - plugins

The editor of Petri Net models




```
<?xml version="1.0" encoding="UTF-8"?>
<PetriNets.petrinet:PetriNet xmi:version="2.0"
    xmlns:xmi="http://www.omg.org/XMI"
    xmlns:PetriNets.petrinet=
    "http:///PetriNets/petrinet.ecore">
 <transitions name="t1" incomingArcs=
    "//@places.0/@outgoingArcs.0">
  <outgoingArcs weight="2"
    toPlace="//@places.1"/>
 </transitions>
 <places name="p1" token="1">
  <outgoingArcs weight="1"
    toTransition="//@transitions.0"/>
 </places>
 <places name="p2" incomingArcs=
    "//@transitions.0/@outgoingArcs.0"/>
</PetriNets.petrinet:PetriNet>
```

Place p1

Reference: URI (or XMI.id)

Tree View

XMI 2.0 View

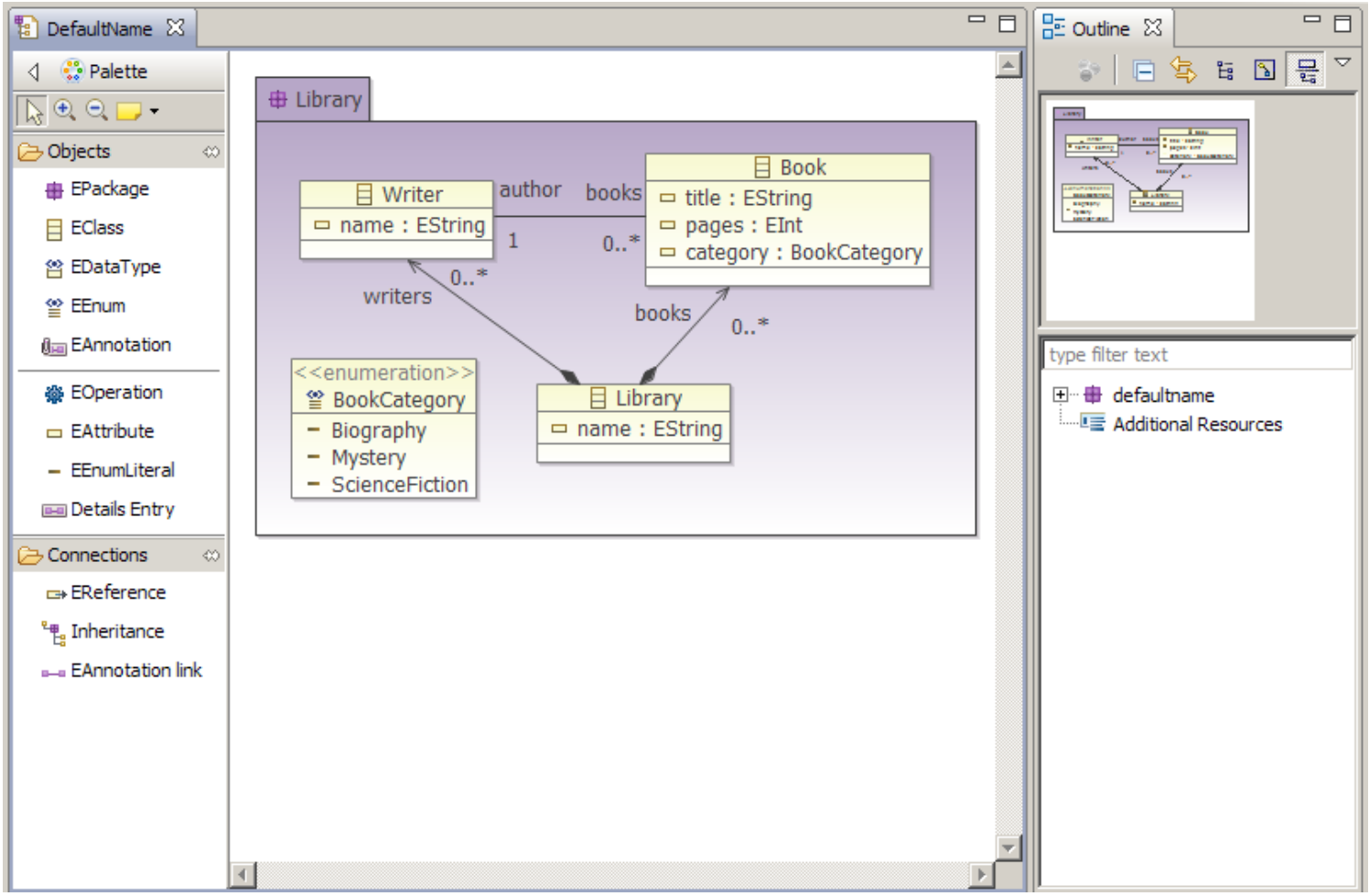
TOOLS, API AND UTILITIES

Basic EMF tools

- Validation
  - Validate constraints over EMF models
- Query
  - High-level query language for EMF
  - See also: Viatra Query ☺
- Compare
  - To structurally compare EMF models (e.g., versioning)
- Teneo
  - Persistency layer over relation databases
- SDO
  - Service Oriented Architecture based on EMF
- CDO
  - distributed, client-server EMF models

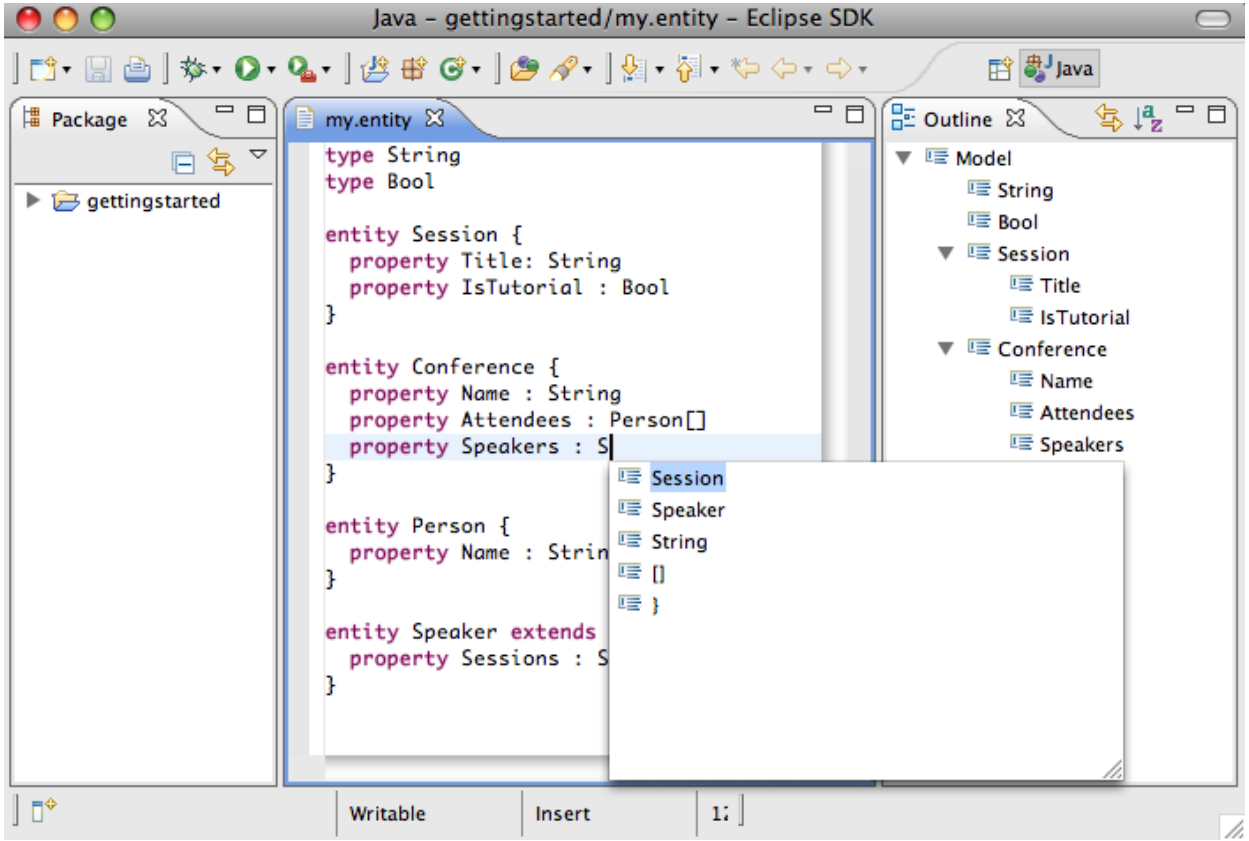
Ecore Tools: Ecore Diagram Editor

- Graphical DSL to define EMF metamodels
  - Based on GMF

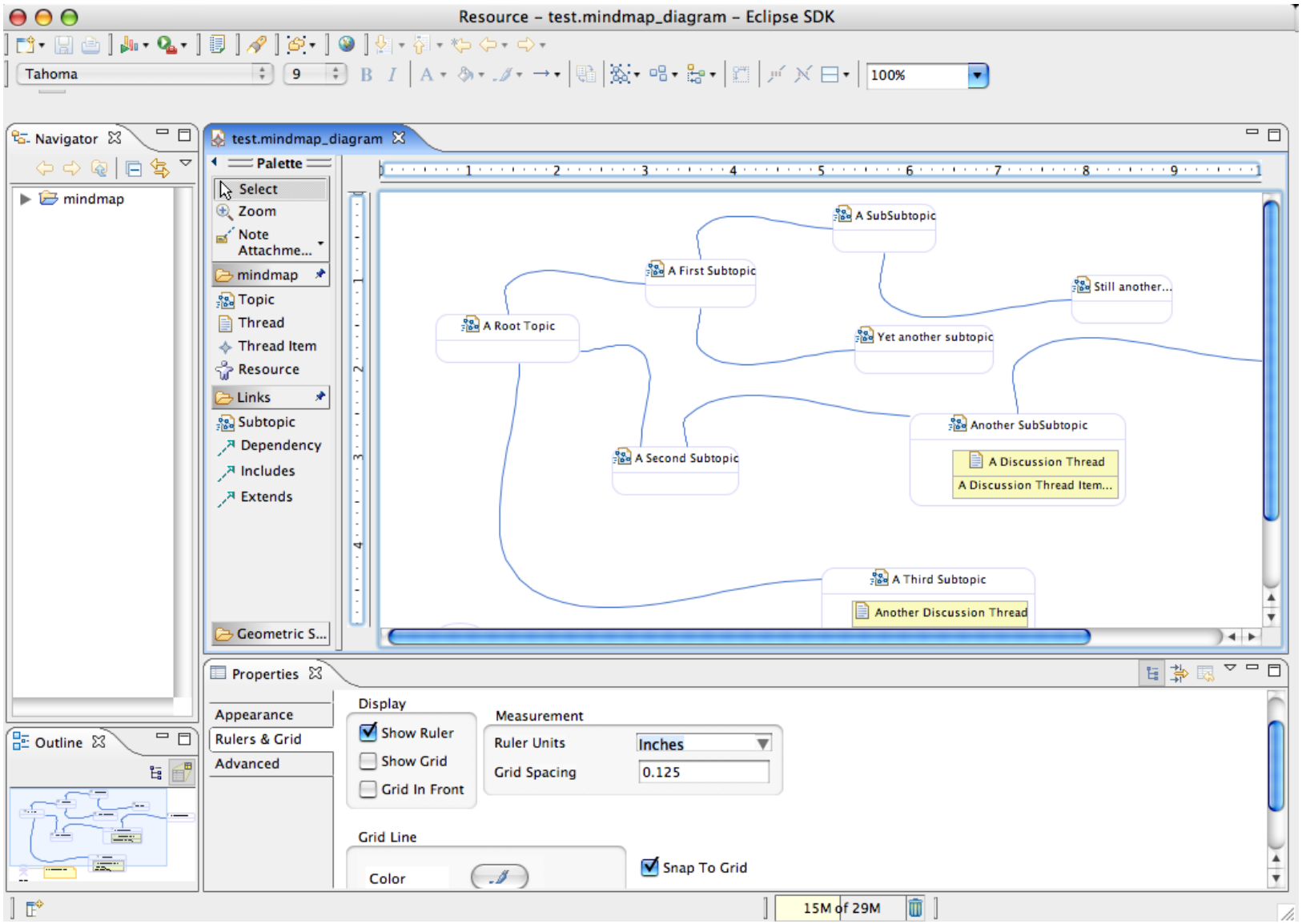


Xtext

- Textual DSL for defining metamodel + textual syntax
- Context-free grammar!
- Generates:
  - Metamodel
  - Parser
  - Editor features



GMF


Resource – test.mindmap_diagram – Eclipse SDK
Tahoma
9
100%
Navigator
mindmap
test.mindmap_diagram
Palette
Select
Zoom
Note Attachme...
mindmap
Topic
Thread
Thread Item
Resource
Links
Subtopic
Dependency
Includes
Extends
Geometric S...
A SubSubtopic
A First Subtopic
Still another...
A Root Topic
Yet another subtopic
Another SubSubtopic
A Second Subtopic
A Discussion Thread
A Discussion Thread Item...
A Third Subtopic
Another Discussion Thread
Outline
Properties
Appearance
Rulers & Grid
Advanced
Display
Show Ruler
Show Grid
Grid In Front
Measurement
Ruler Units
Inches
Grid Spacing
0.125
Grid Line
Color
Snap To Grid
15M of 29M