From Models to Code

Marc-Philippe Huget
LISTIC-Polytech'Annecy Chambéry
University of Savoie
(Marc-Philippe.Huget@univ-savoie.fr)

Who am I?

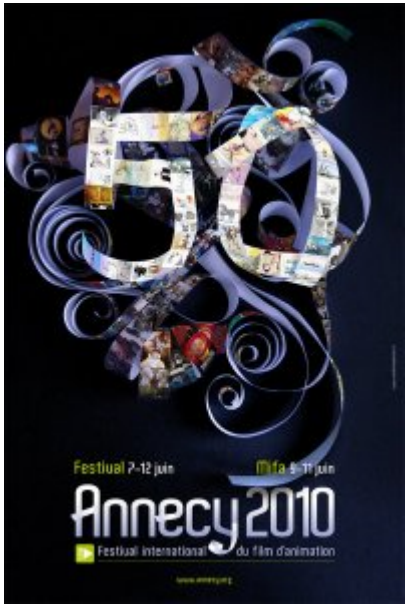
- Associate Professor in Computer Science, mainly in Software Engineering
- Research interests:
  - Multiagent systems
  - Agent oriented software engineering (AOSE)
    - Modeling (Agent UML among others)
    - Methodology
  - Model-driven Engineering
  - Big data

Where do I work?

LISTIC laboratory

- Image processing
- Information fusion and processing
- Fuzzy functions
- Ontology
- Software engineering
- Performance analysis

Annecy, little Venice



Talk outline

- Introduction towards model-driven development
- What is a model?
- Model-Driven Architecture (MDA)
- Model-Driven Engineering (MDE)
- Domain-Specific Languages (DSL)

Model-driven Development (MDD)

Model

Model to Model (M2M)

Model-driven Architecture (MDA)

Model-driven Engineering (MDE)

Domain-specific Language (DSL)

Model to Text (M2T)

Model-Driven Development

(or: Why I'd like writing programs that write programs rather than writing programs…) (J.-M. Jezequel)

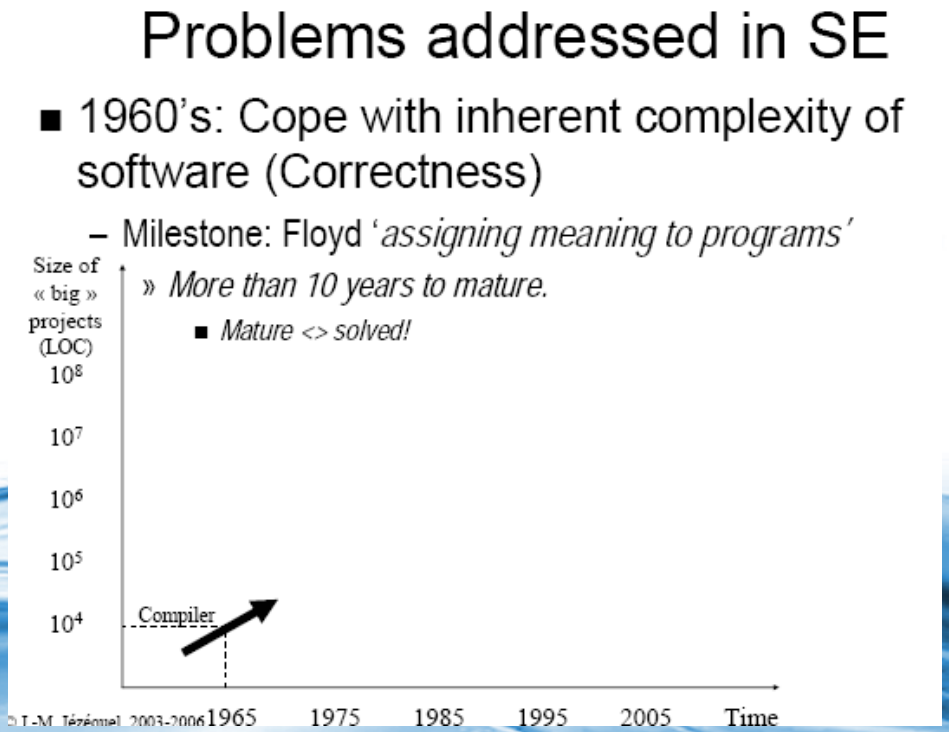
Introduction towards Model-Driven Development

1950s

- Computers and programs appear
- Programs are no longer military ones
- Programs are written with machine code: 100110011…
- Machine code is part of first generation languages
- But second generation languages appear such as Assembler to reduce complexity to develop programs
- And some third generation languages such as FORTRAN (1954)

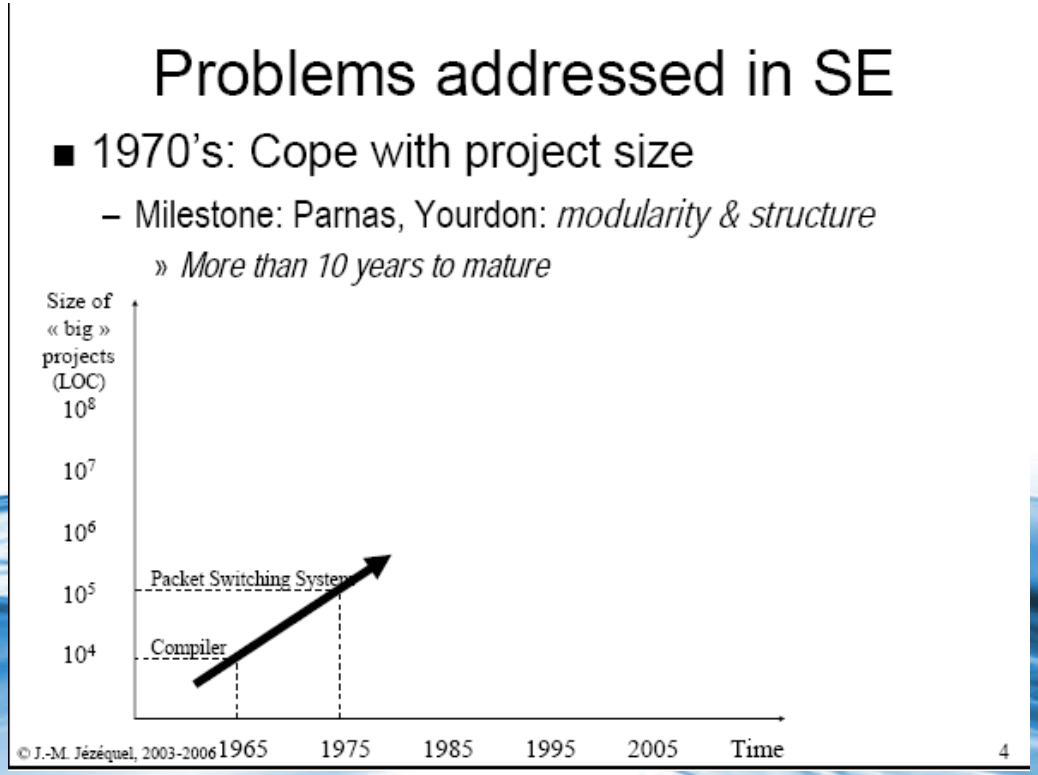
1960s

- The term « Software Engineering » is coined at the NATO Conference in 1969



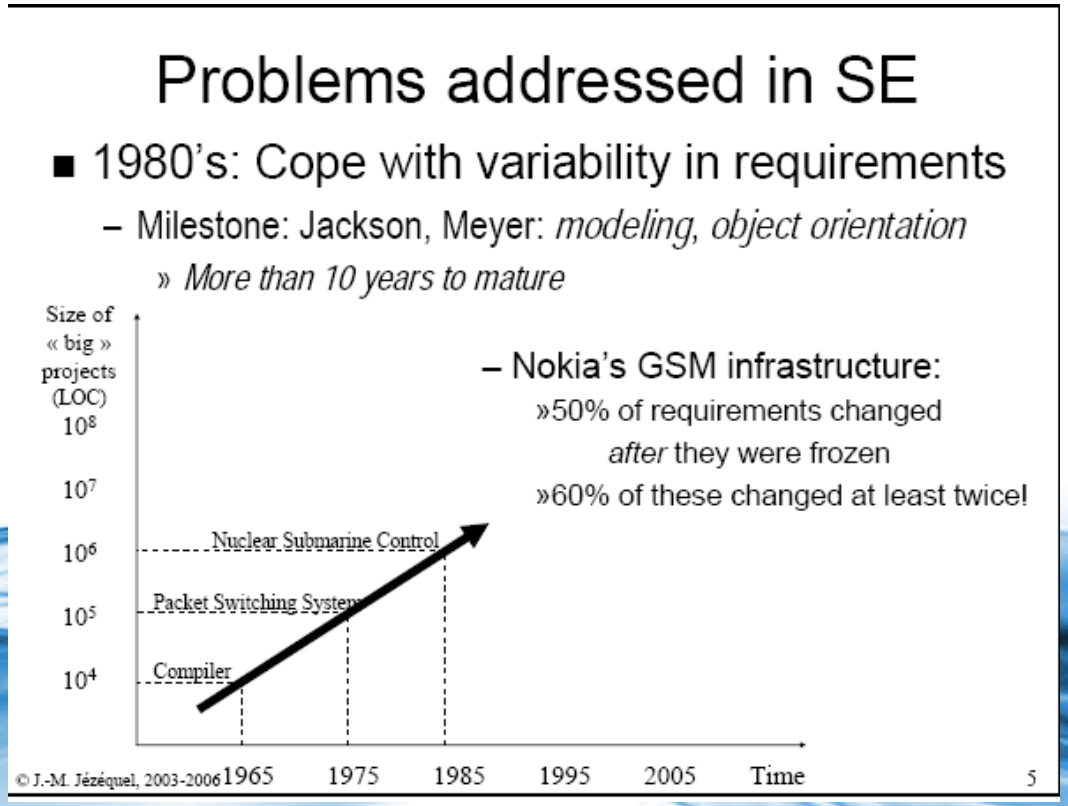
1970s

- Third generation languages such as C (Ritchie, 1972)



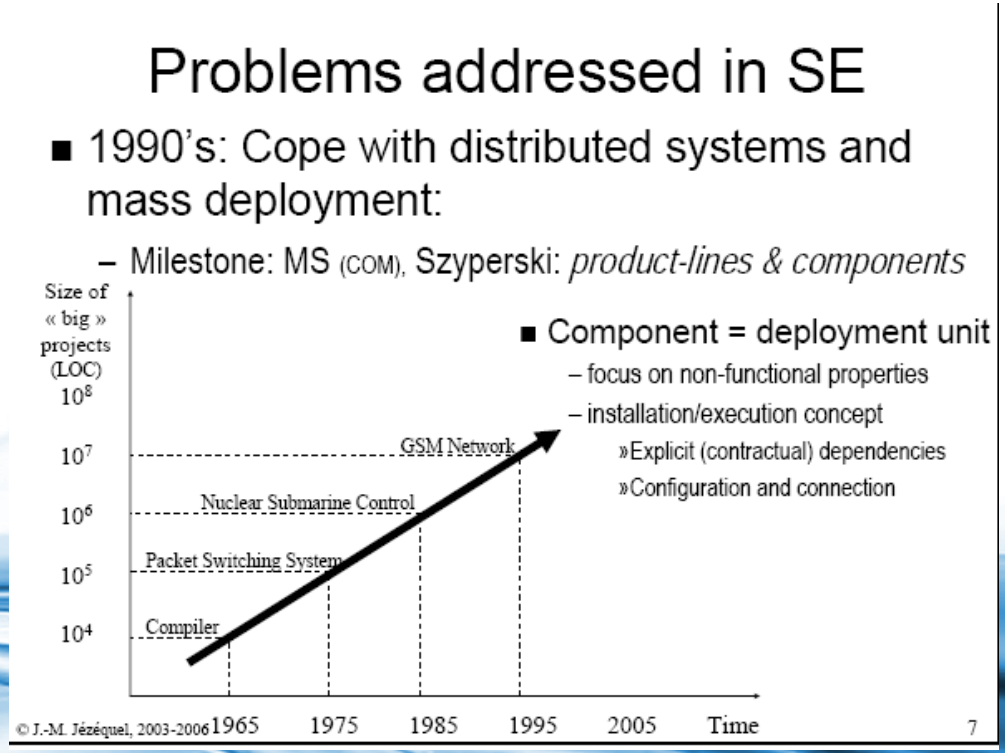
1980s

- From procedural languages to object programming, C++ (1983)



1990s

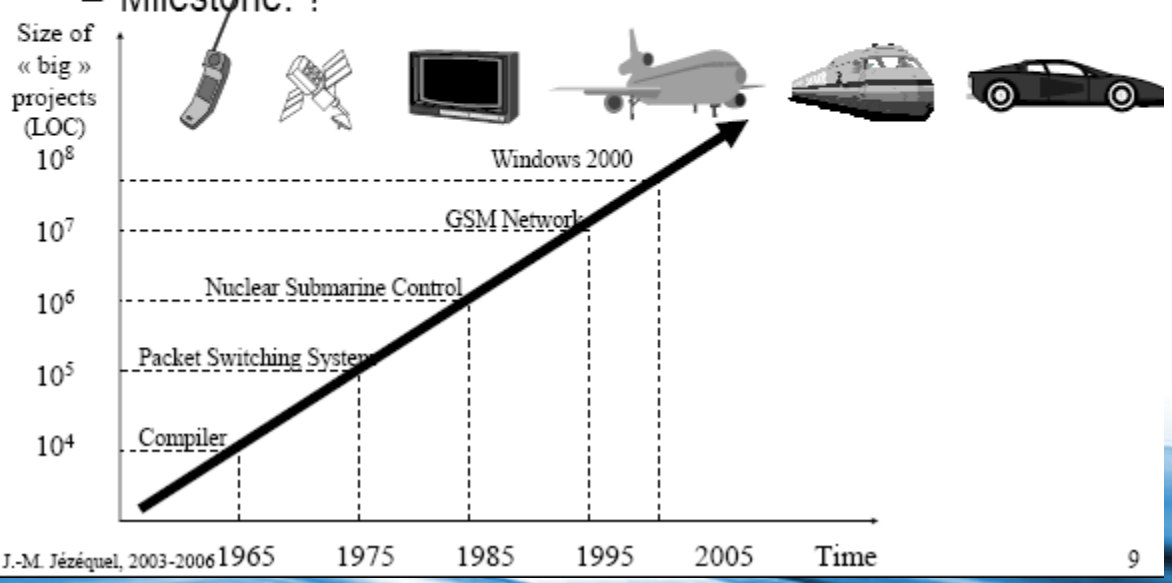
- Classes are packed into packages and components : component-based programming



2000s

Problems addressed in SE

- 2000's: pervasive software integration, accelerating technological changes (platforms)
  - Milestone: ?

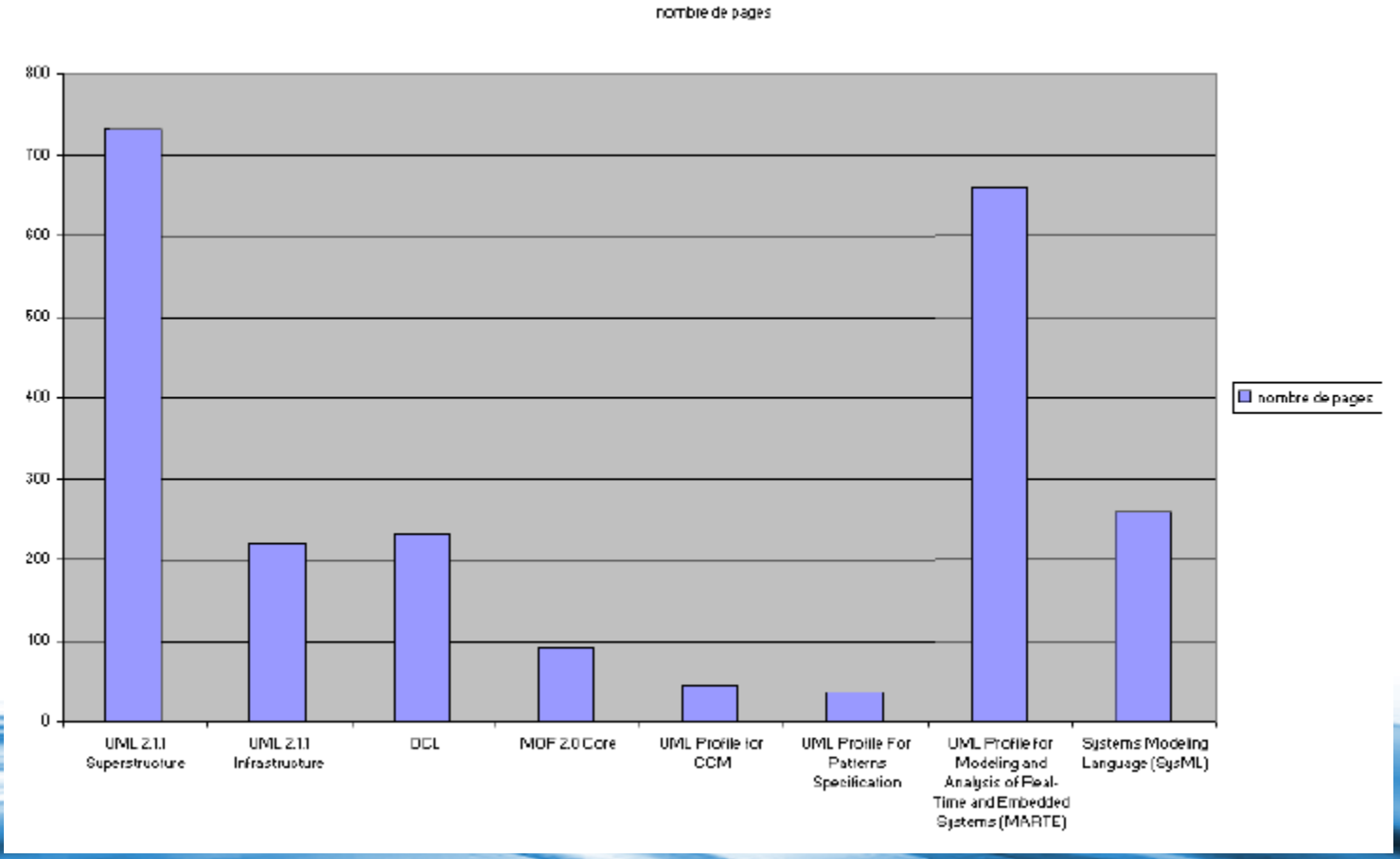


Complexity is everywhere

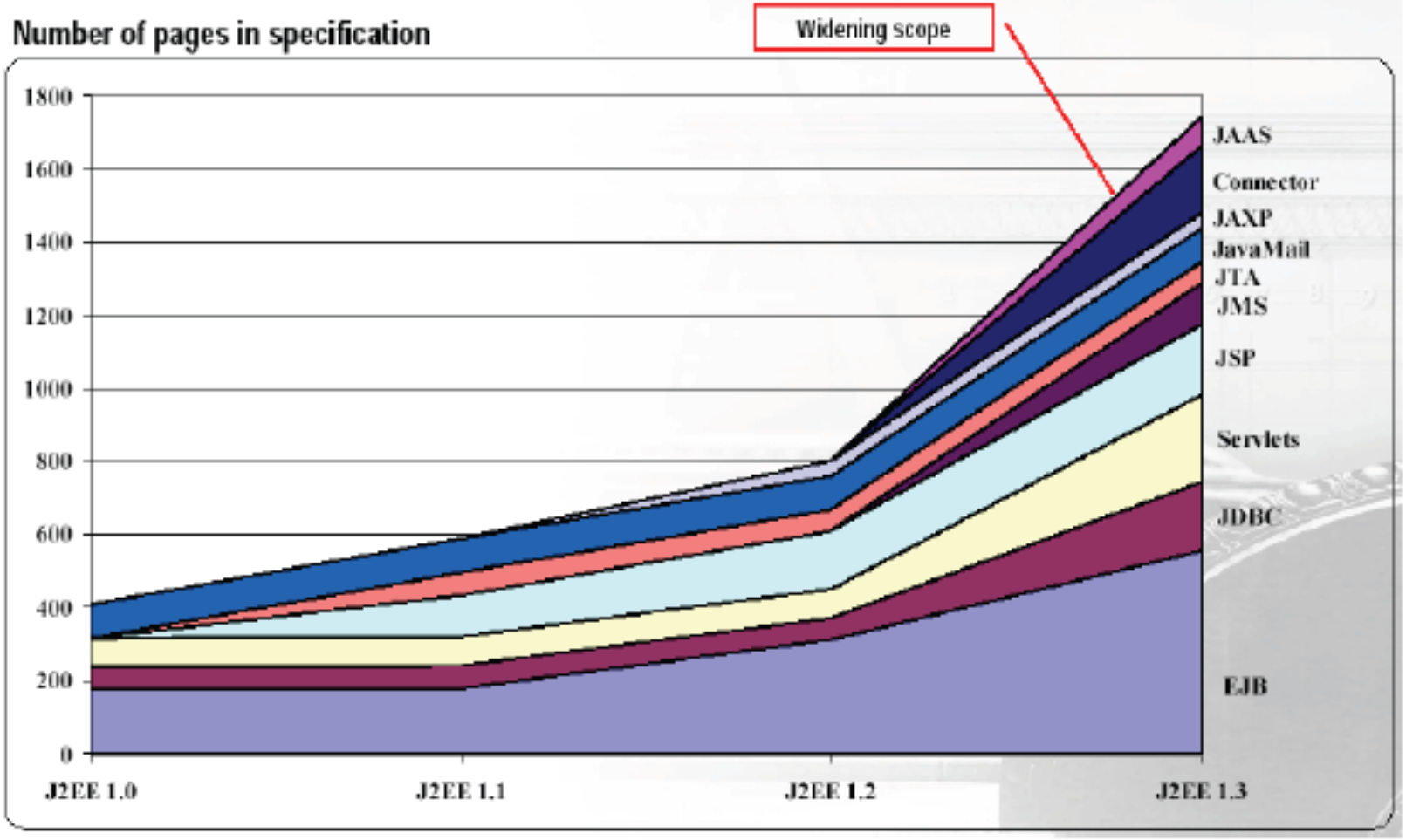
Lines of Code

- Windows NT 3.1: 5M LOC
- Windows 2000: >29M LOC
- Mac OS X 10.4: 89M LOC
- Firefox: 1M LOC

Complexity is everywhere (cont'd)



Complexity is everywhere (cont'd)



d'après Interactive-Objects

The death march of specifications

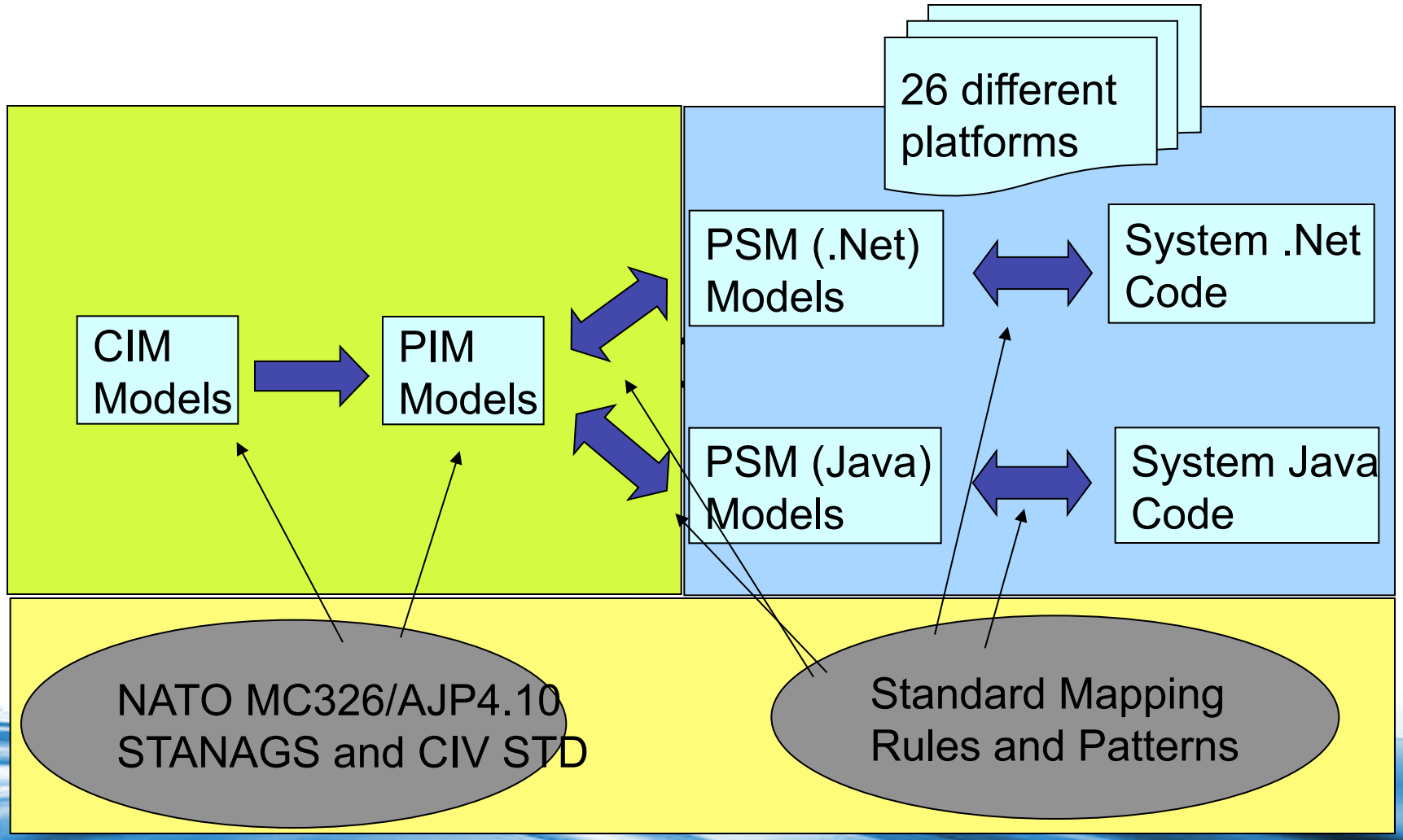
J2EE	Date of release
1.0	1999
1.1	
1.2	2000
1.3	2001
1.4	2003
5	2006
6	2010

Too many platforms

- Technological platforms
  - .NET
  - Java (SE, EE, ME)
- Hardware platforms
  - Xbox 360
  - PS3
  - Wii
- Etc.

The Military Medical Information Systems Example

- No common NATO Electronic Health Record
  - Nations' own responsibility
    - "Push" from national EHR to NATO: No semantics or syntax standard agreed upon
  - Periodic reports about medical status
    - Delayed and aggregated: Not good enough for many purposes
- 26 NATO member nations
  - 26 different Electronic Health Records (at least: one nation may have more than one system)
  - 26 different technological "platforms"?
  - 26 different "standards"?
  - Maturity of systems varies
- Multinational Integrated Medical Unit (MIMU)
  - Many nations in one medical treatment facility
  - Should (must?) share information


26 different platforms
CIM Models
PIM Models
PSM (.Net) Models
System .Net Code
PSM (Java) Models
System Java Code
NATO MC326/AJP4.10 STANAGS and CIV STD
Standard Mapping Rules and Patterns

If you were still not convinced…

From the Standish group, in 2003, on 13522 IT projects

- 34% respect the deadline and cost
- 51% respect the deadline but increase the cost
- 16% are stopped before ending

It is time to change the way to develop software

Model-Driven Development

Philosophy behind model-driven development

- Gains abstraction: adopt an eagle's eye view of software rather than a mole's one
- Gets separation of concerns between business stuff and technological one
- Modifies the leitmotive « Write once, runs everywhere » to « Model once, generate anywhere »

What is a model?

Are they the same?

Why modeling?

Modeling, in the broadest sense, is the *cost-effective use of* ***something in place of something else*** *for some cognitive purpose. It allows us to* **use something** that is ***simpler, safer or cheaper than reality*** *instead of reality* for some purpose.

A **model** represents reality for the given purpose; the model is an **abstraction of reality** in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality.

Jeff Rothenberg

One real « car »: several models depending on the point of view…
Hundreds of thousands data…

Ferrari F60

Different points of view for modelling:

- If you are Scuderia addict:
  - Drivers
  - Nb of victories
  - Etc.
- If you are an engineer:
  - Nb of horsepower
  - Maximum speed
  - Etc.

Simplified view of system and with only worthwhile data

What is a model?

[OMG MDA Guide 1.0.1]

A *model* of a system is a description or specification of that system and its environment for some certain purpose. A model is often presented as a **combination of drawings and text**. The text may be in a modeling language or in a natural language.

Tower

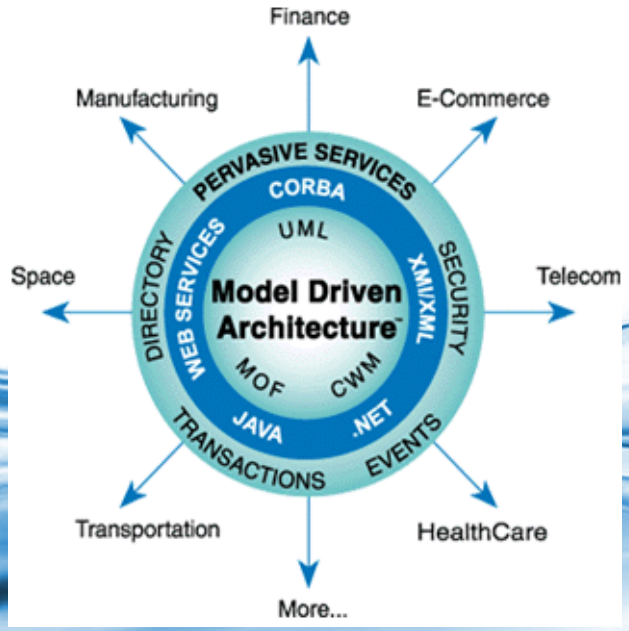
One model, two instantiations

Model

- A model is a description, a **partial** specification of a system:
  - Abstraction of a real entity with only interesting features for a given context and a given aim
  - Subjective view of a system
- The aim of a model is:
  - Ease the understanding of a system
  - Simulate a system under use
- Outstanding examples:
  - Economical models
  - Social models

« One and three chairs », Joseph Kossuth, 1965

Model-Driven Architecture (MDA)

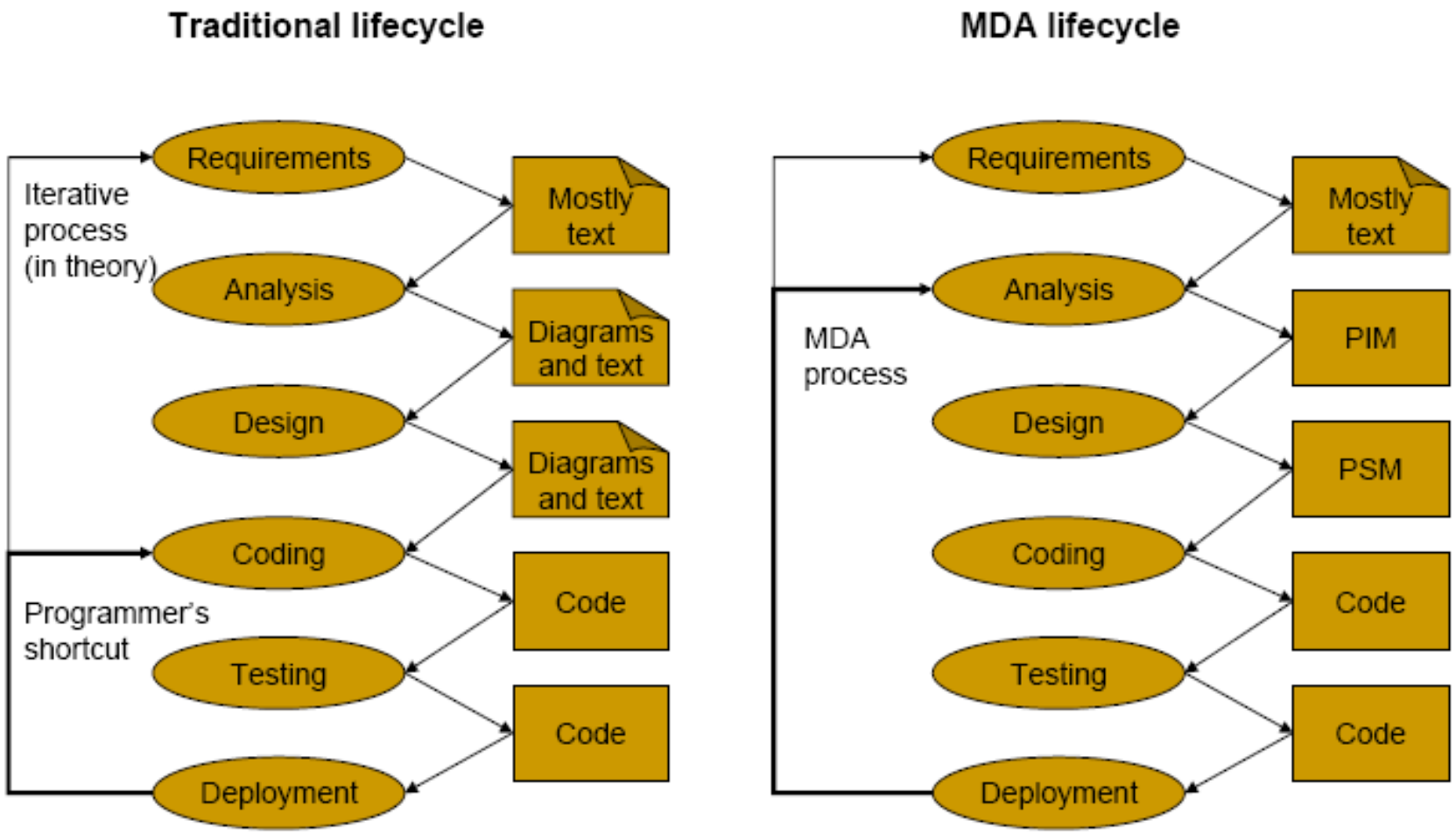


Model-Driven Architecture (MDA)

- Proposed by the Object Management Group (OMG) consortium
- Part of a collection of standards
  - UML
  - SysML
  - XMI
  - MOF
  - SPEM
  - CWM
- Uses extensively UML models

Model-driven Architecture (MDA)

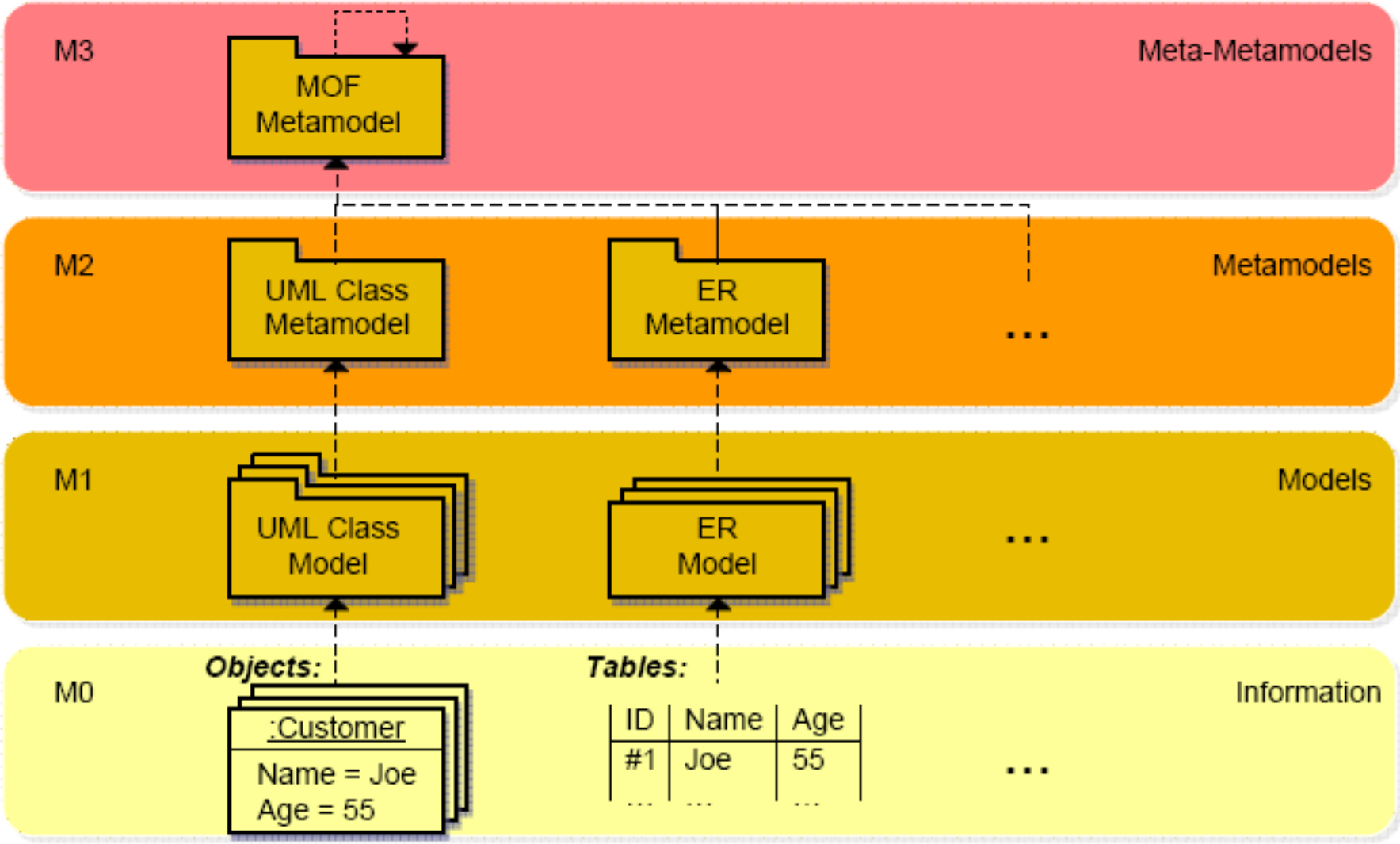
- Separation of business functions from technological implementation
  - Represent common concepts in a model independent of computation and technological platforms
    - CIM: Computation Independent Model
    - PIM: Platform Independent Model
  - Use the PIM model transformation to create new platform specific models
    - PSM: Platform Specific Model
  - Generate executable code (applications) from the PSM



Source: Kleppe et al 2003

Several models

- CIM (Computation Independent Model): this is the business model, frequently realized during requirements engineering
- PIM (Platform Independent Model): this is the instance of the application
- PSM (Platform Specific Model): this is the instance of the application for a specific platform (either technical or conceptual)



The Four Modeling Layers of the OMG

Layer M0: instances

At the M0 layer, there is the running system in which the actual (« real ») instances exist.

This is where you do the « new » in object-oriented programs

But to define these instances, we need a *form*: the M1 layer

Layer M1: the Model of the System

The M1 layer contains models, for example, a UML model of a software system. In the M1 model, for instance, the concept of *Customer* is defined with the properties of *name, street* and *city*

The M1 elements define what M0 instances look like

But defining models for instances requires to know how to model models…

Layer M2: The Model of the Model

Classes, operations, attributes and associations defined at the M1 layer need to be modeled.

The M2 layer defines how elements at the M1 layer should be defined: what is a class, what is an attribute, etc.

Layer M3: the Model of M2

- M0 defines the data for your application
- That are based on classes defined in M1
- That are based on elements that explained how to write classes/concepts: M2
- Finally, we need to explain how to write these elements: the M3 layer

Layer M3 (cont'd)

And a M4 layer ?

And a M5 layer ?

And so on ?

Hopefully not, M3 is a reflexive layer, it is used to describe itself

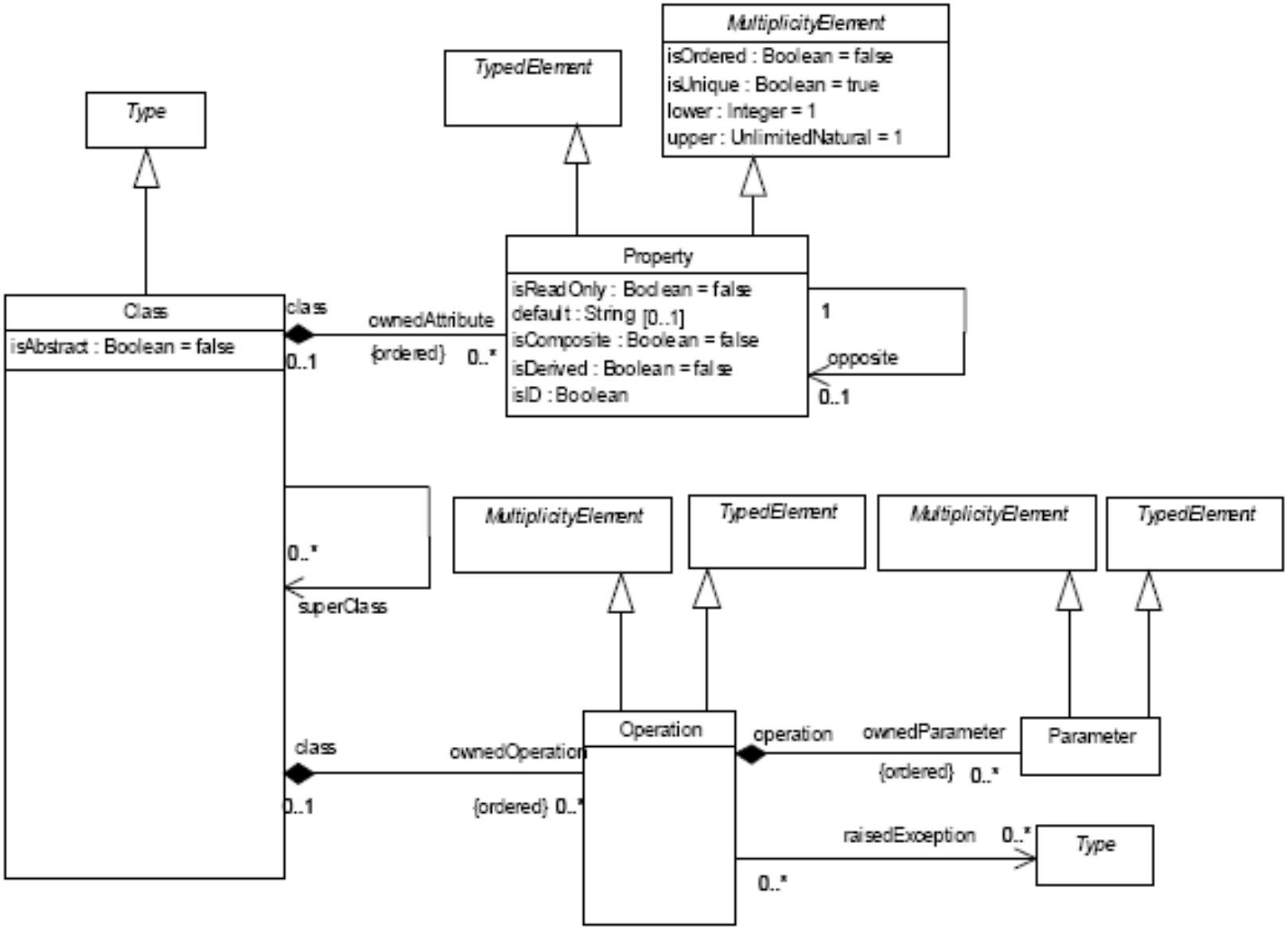
M3 Meta-metamodels

Two meta-metamodels exist to describe metamodels:

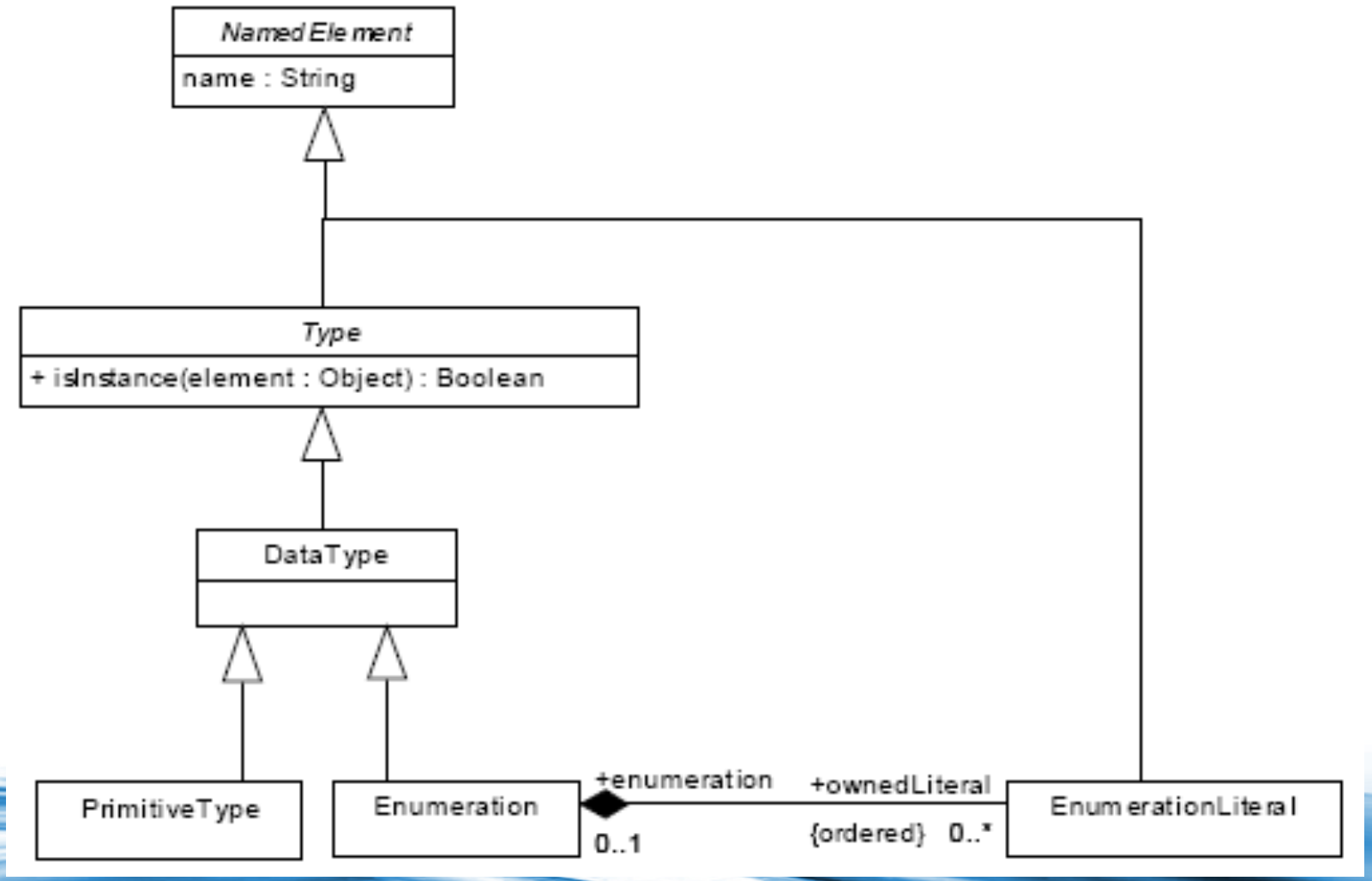
- OMG's Meta-Object Facility (MOF)
- Eclipse Modeling Framework (ecore) (discussed in MDE part)

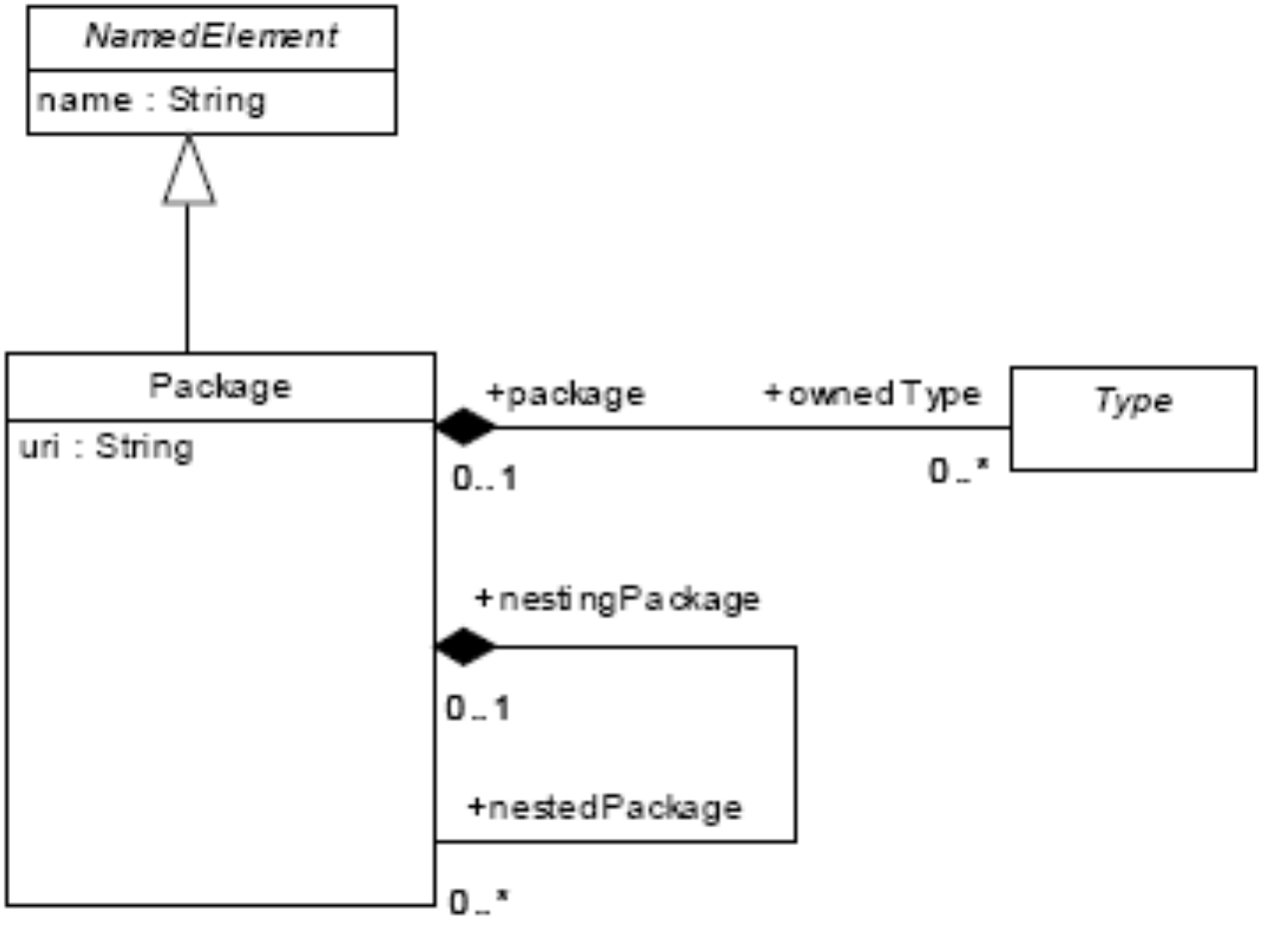
Meta-Object Facility

- Proposed by OMG as support of Model-Driven Architecture
- Comes into two flavours:
  - Essential MOF (EMOF)
  - Complete MOF (CMOF)
- Designers do not have to pay too much attention about MOF intricacies: tools directly represent classes, attributes and operations



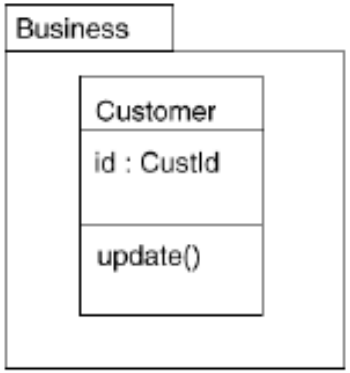
EMOF Classes


NamedElement
name : String
Type
+ isInstance(element : Object) : Boolean
DataType
PrimitiveType
Enumeration
+enumeration
0..1
+ownedLiteral
{ordered}
0..*
EnumerationLiteral

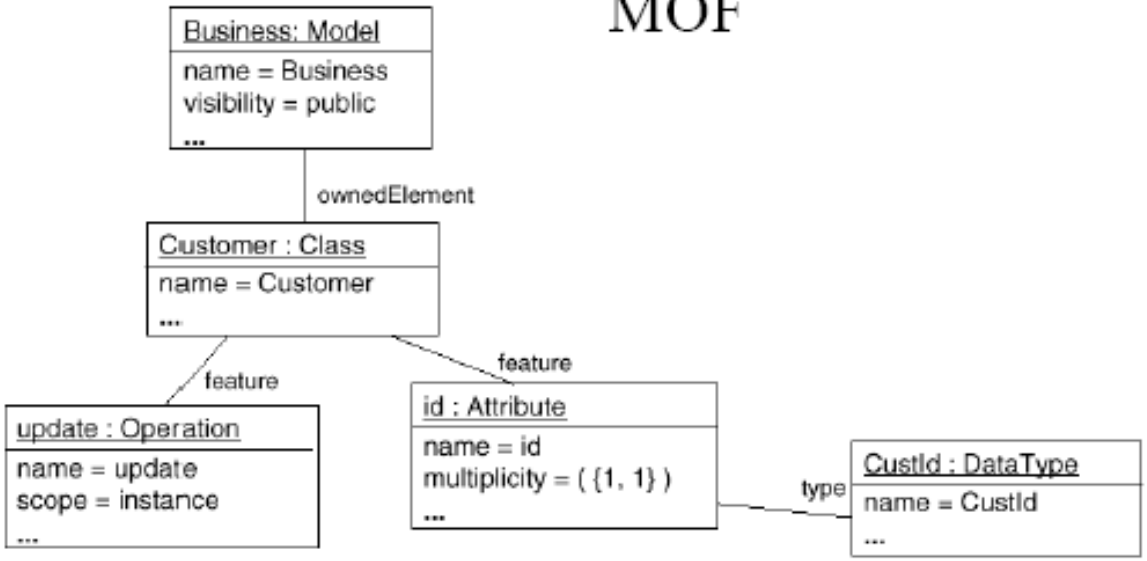

NamedElement
name : String
Package
uri : String
+package
+ownedType
Type
0..1
0..*
+nestingPackage
0..1
+nestedPackage
0..*

Essential MOF

- Sufficient to represent simple metamodels
- Complex metamodels with reflection require CMOF



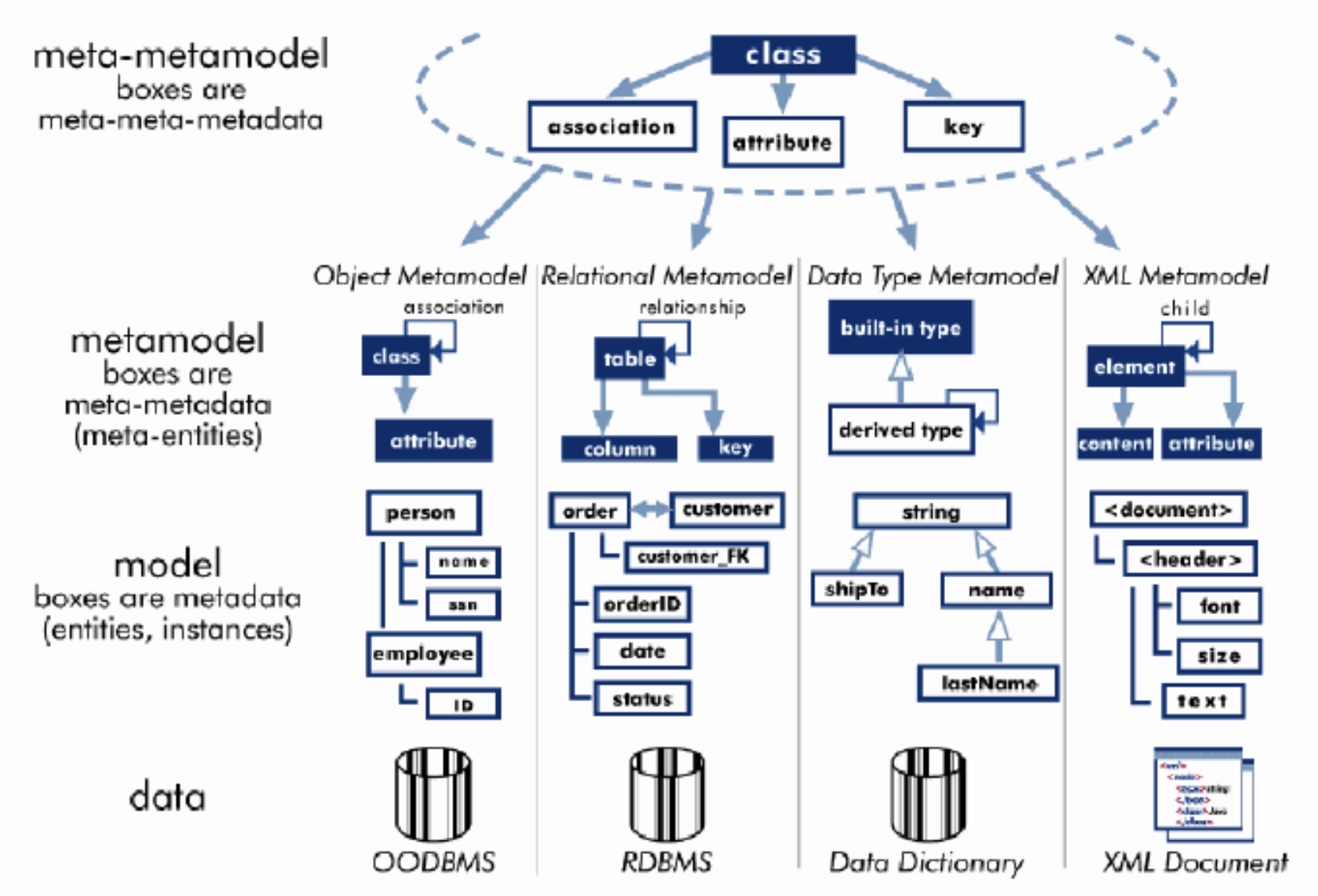
UML



MOF

```
<Model>
  <name>Business</name>
  <visibility xmi.value="public"/>
  <Class>
    <name>Customer</name>
    <feature>
      <Attribute>
        <name>id</name>
        <multiplicity>
          <XMI.field> 1</ XMI. field>
          < XMI. field> 1</ XMI. field>
        </multiplicity>
      </Attribute>
    </feature>
  </Class>
</Model>
```

XMI


meta-metamodel
boxes are
meta-meta-metadata
class
association
attribute
key
Object Metamodel
Relational Metamodel
Data Type Metamodel
XML Metamodel
metamodel
boxes are
meta-metadata
(meta-entities)
association
class
attribute
relationship
table
column
key
built-in type
derived type
child
element
content
attribute
model
boxes are metadata
(entities, instances)
person
name
ssn
employee
ID
order
customer
customer_FK
orderID
date
status
string
shipTo
name
lastName
<document>
<header>
font
size
text
data
OODBMS
RDBMS
Data Dictionary
XML Document

XMI

- XML Metadata Interchange (XMI)
- Part of OMG specifications
- Responsible to save MOF (and Ecore) objects as an XML file: the M2, M1 and M0 layers
- Used to exchange models between UML tools

XMI (cont'd)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns="Families">
  <Family lastName="March">
    <father firstName="Jim"/>
    <mother firstName="Cindy"/>
    <sons firstName="Brandon"/>
    <daughters>
       <firstName>Brenda</firstName>
    </daughters>
  </Family>
  <Family lastName="Sailor">
    <father firstName="Peter"/>
    <mother firstName="Jackie"/>
    <sons firstName="David"/>
    <sons firstName="Dylan"/>
    <daughters firstName="Kelly"/>
  </Family>
  <Book>
   <title> "Easy ATL" </title>
  </Book>
</xmi:XMI>
```

M0 layer example

XMI (cont'd)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0"
   xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore">
  <ecore:EPackage name="Families">
   <eClassifiers xsi:type="ecore:EClass" name="Family">
     <eStructuralFeatures xsi:type="ecore:EAttribute" name="lastName" ordered="false"
        unique="false" lowerBound="1" eType="#/1/String"/>
     <eStructuralFeatures xsi:type="ecore:EReference" name="father" ordered="false"
        lowerBound="1" eType="#/0/Member" containment="true" eOpposite="#/0/Member/
      familyFather"/>
     <eStructuralFeatures xsi:type="ecore:EReference" name="mother" ordered="false"
        lowerBound="1" eType="#/0/Member" containment="true" eOpposite="#/0/Member/
      familyMother"/>
     <eStructuralFeatures xsi:type="ecore:EReference" name="sons" ordered="false"
        upperBound="-1" eType="#/0/Member" containment="true" eOpposite="#/0/Member/familySon"/
      >
     <eStructuralFeatures xsi:type="ecore:EReference" name="daughters" ordered="false"
        upperBound="-1" eType="#/0/Member" containment="true" eOpposite="#/0/Member/
      familyDaughter"/>
   </eClassifiers>
</xmi:XMI>
```

XMI (cont'd)

```
<eClassifiers xsi:type="ecore:EClass" name="Member">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="firstName" ordered="false"
        unique="false" lowerBound="1" eType="#/1/String"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="familyFather"
    ordered="false"
        eType="#/0/Family" eOpposite="#/0/Family/father"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="familyMother"
    ordered="false"
        eType="#/0/Family" eOpposite="#/0/Family/mother"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="familySon"
    ordered="false"
        eType="#/0/Family" eOpposite="#/0/Family/sons"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="familyDaughter"
    ordered="false"
        eType="#/0/Family" eOpposite="#/0/Family/daughters"/>
  </eClassifiers>
</ecore:EPackage>
```

XMI (cont'd)

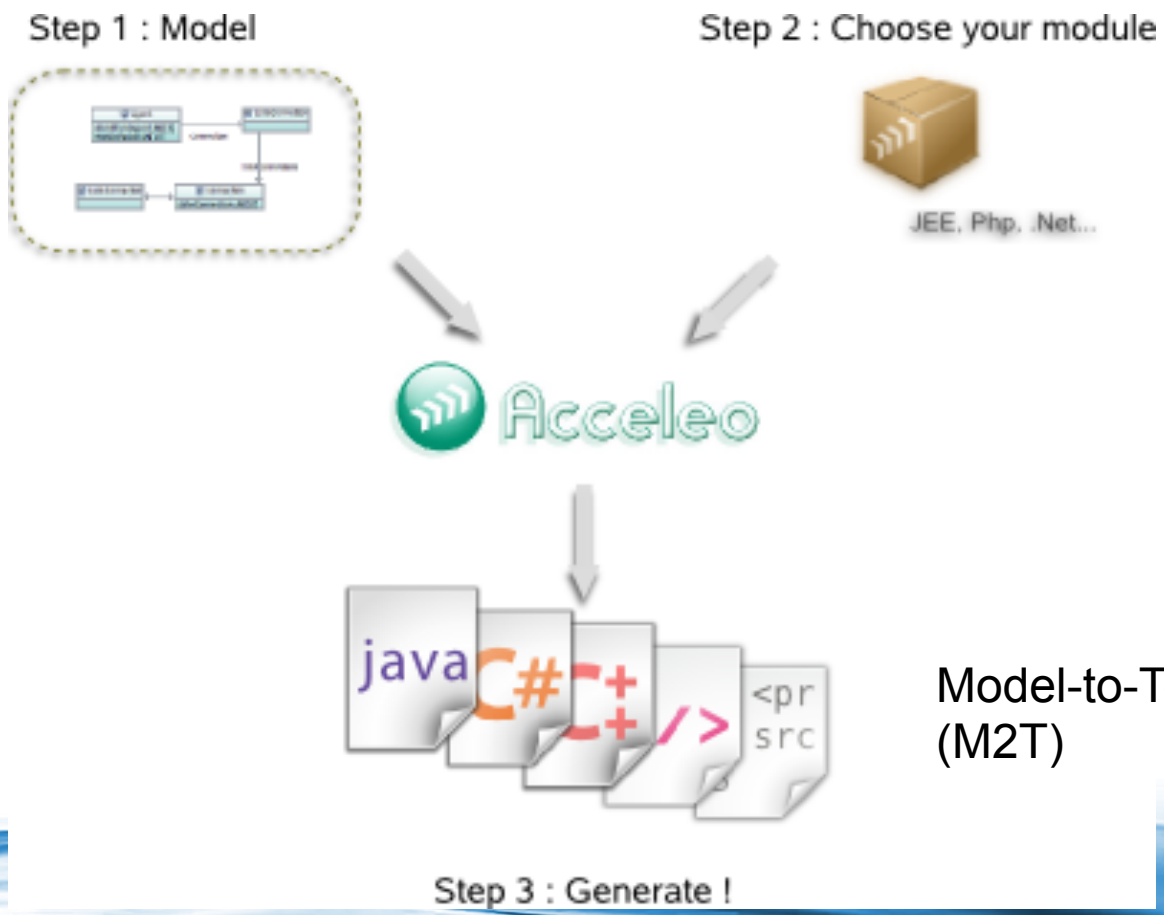
- Hopefully for you, you do not have to read XMI...

Tools for MDA

- UML-based tools: Code generation is available and for some of them, it is possible to modify the templates
  - Poseidon (Gentleware AG), Omondo, Rational Rose (IBM), etc.
- Specific MDA tools:
  - Commercial:
    - Mia Software (Mia Software), MetaEdit+ (Metacase)
  - Open Source:
    - Acceleo, OpenArchitectureWare (retired), Kermeta (even if it is a MDE tool)

MDA in practice

Acceleo



Model-to-Text tool (M2T)

Available at: http://www.acceleo.org

Installation

1. Download All-in-one Eclipse Modeling Tools (http://www.eclipse.org/downloads/packages/eclipse-modeling-tools-includes-incubating-components/galileosr1)
2. Unzip the archive to a directory
3. Execute Eclipse
4. Check in File>New you have Eclipse Modeling Framework and UML 2.1

Installation (cont'd)

1. Go to Help>Install new software
2. Click on Button *Add* and this new site: http://www.acceleo.org/update
3. Then select the Acceleo site in the combo *Work with*
4. Check the Acceleo proposal below as shown on Figure

Install

Available Software

Check the items that you wish to install.

Work with: http://www.acceleo.org/update

Add...

Find more software by working with the 'Available Software Sites' preferences.

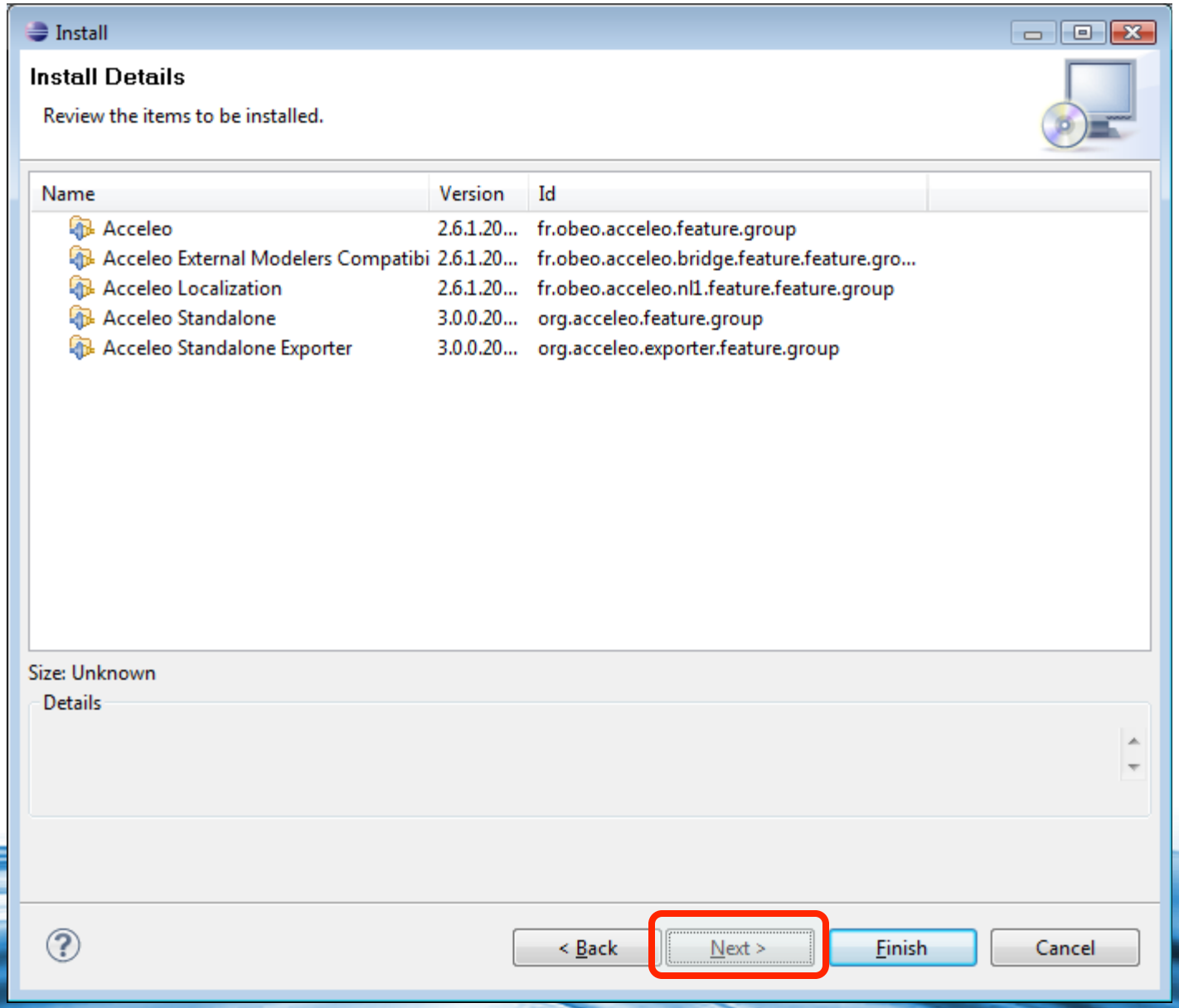
type filter text

Name	Version
Acceleo	
Acceleo	2.6.1.200909102231
Acceleo External Modelers Cor	2.6.1.200909102231
Acceleo Localization	2.6.1.200909102231
Acceleo Standalone	3.0.0.200909102301
Acceleo Standalone Exporter	3.0.0.200909102301

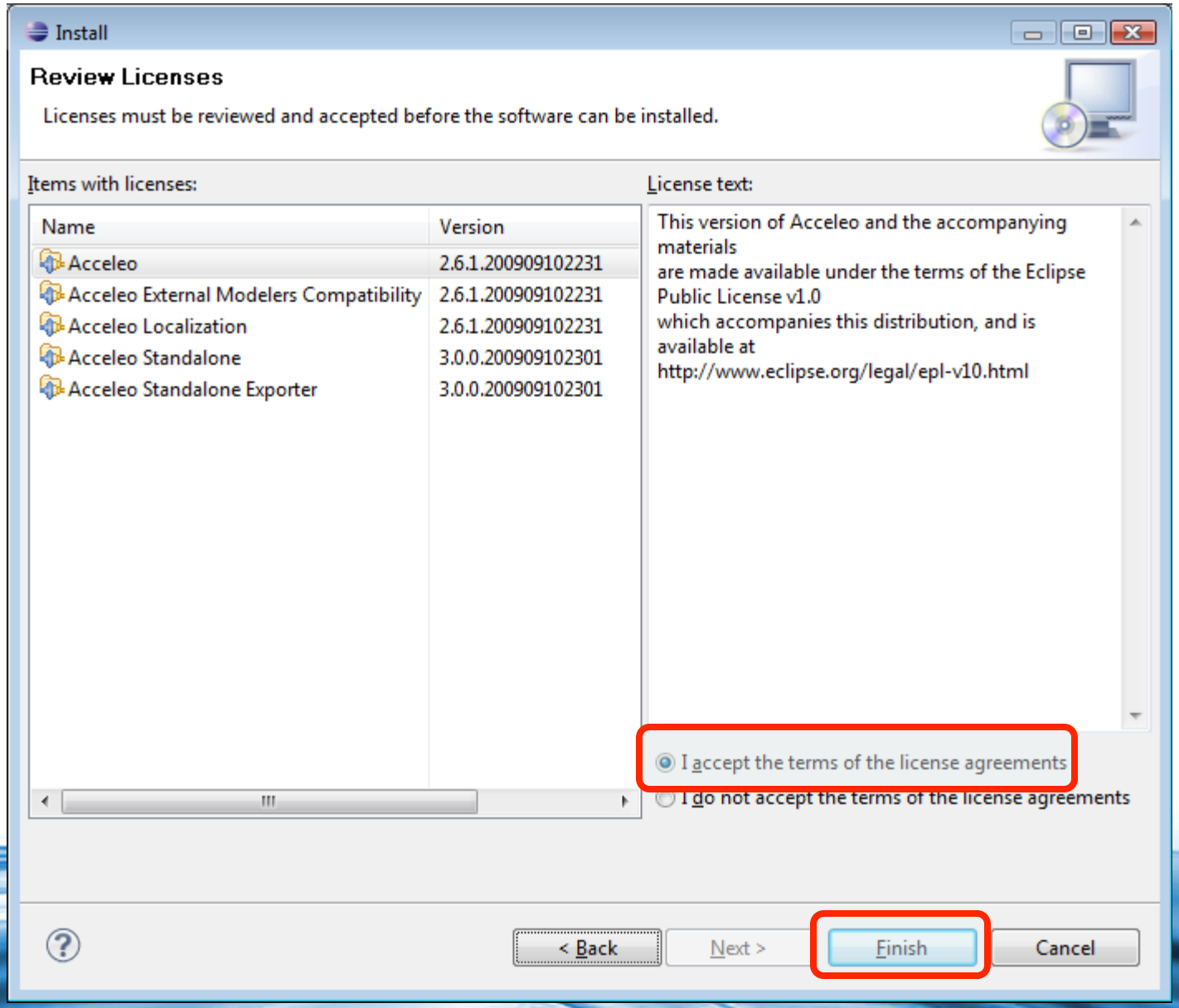
Details

- [x] Show only the latest versions of available software
- [ ] Hide items that are already installed
- [x] Group items by category
- What is already installed?
- [x] Contact all update sites during install to find required software

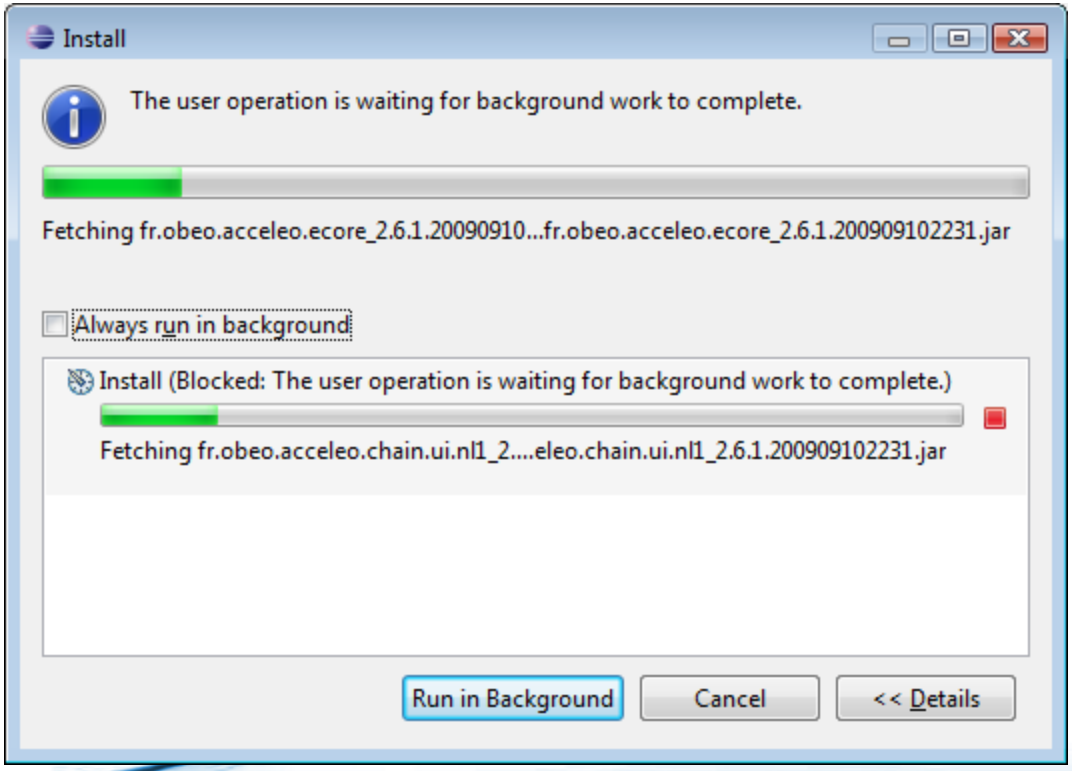
< Back | Next > | Finish | Cancel



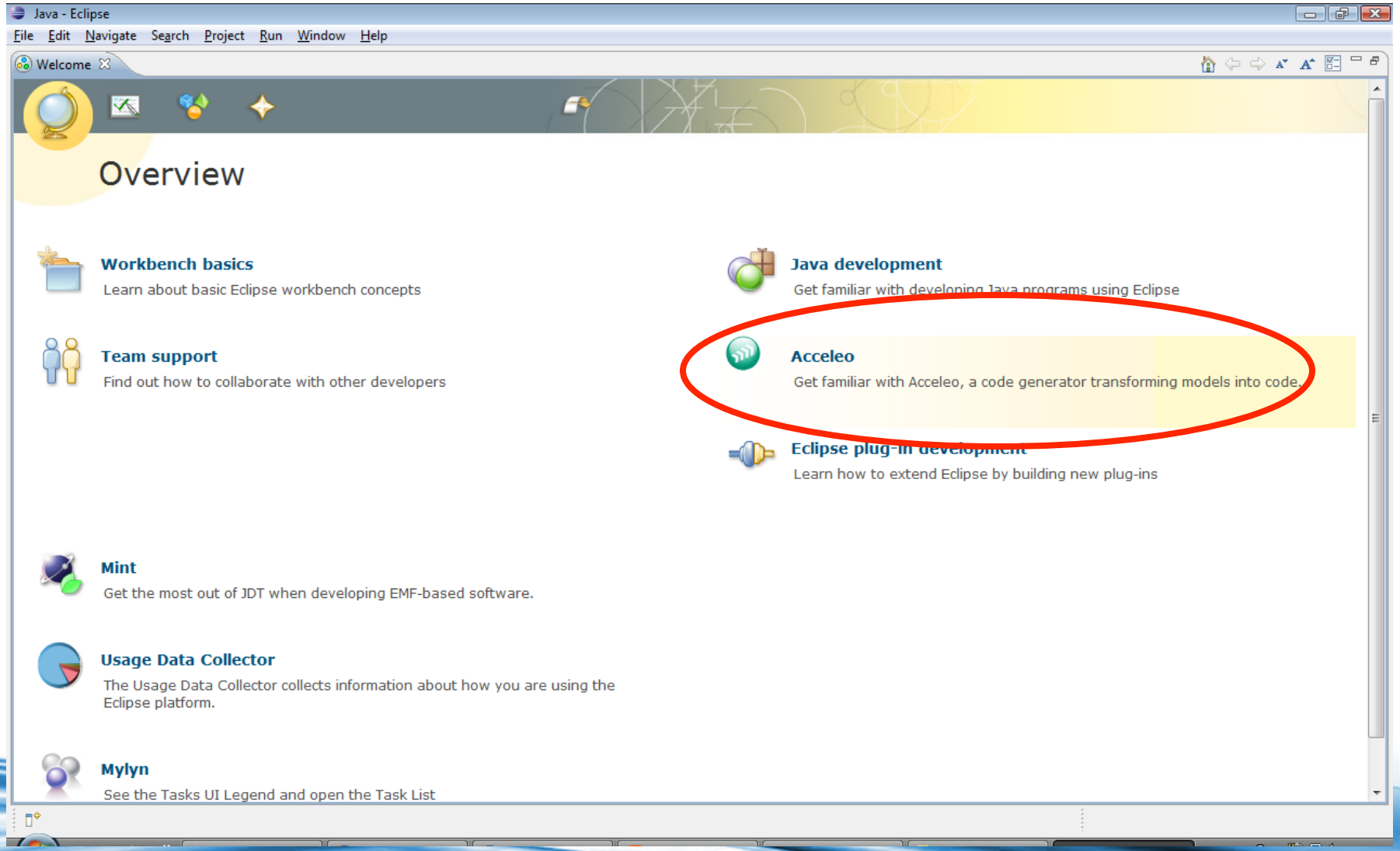
After clicking *Next* button



Accept the license agreements and Finish



Eclipse is updating the platform with Acceleo features



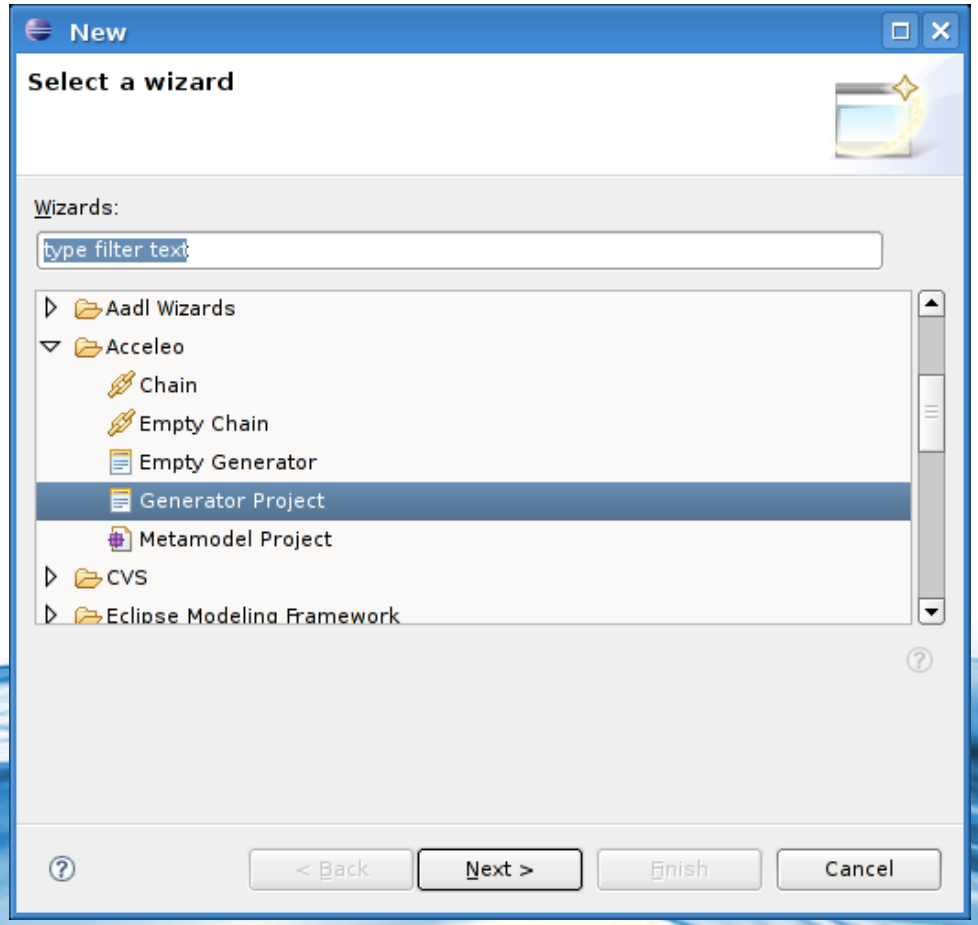
Acceleo is now ready for use

Acceleo in use

- Change perspective to the Acceleo one (the second in the list of Acceleo perspective)
- Development following two steps:
  1. Create the metamodel as UML class diagram
  2. Create the script
  3. Generate the target model

Metamodel creation

- File>New>Acceleo>Generator project



Metamodel creation

- Name your project, here *PhoneDirectory*
- Then, create your metamodel via UML 2 plugin

Acceleo - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer | Hierarchy

- PhoneDirectory
  - src
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties
  - plugin.properties

New

Select a wizard

Créer un nouveau projet de génération

Wizards:

type filter text

- Plug-in Development
- Tasks
- UML 2.1 Diagrams
  - Activity Diagram
  - Class Diagram
  - Component Diagram
  - Composite Structures Diagram
  - Deployment Diagram
  - Profile Definition Diagram
  - Sequence Diagram
  - State Machine Diagram
  - Use Case Diagram

< Back | Next > | Finish | Cancel

Outline

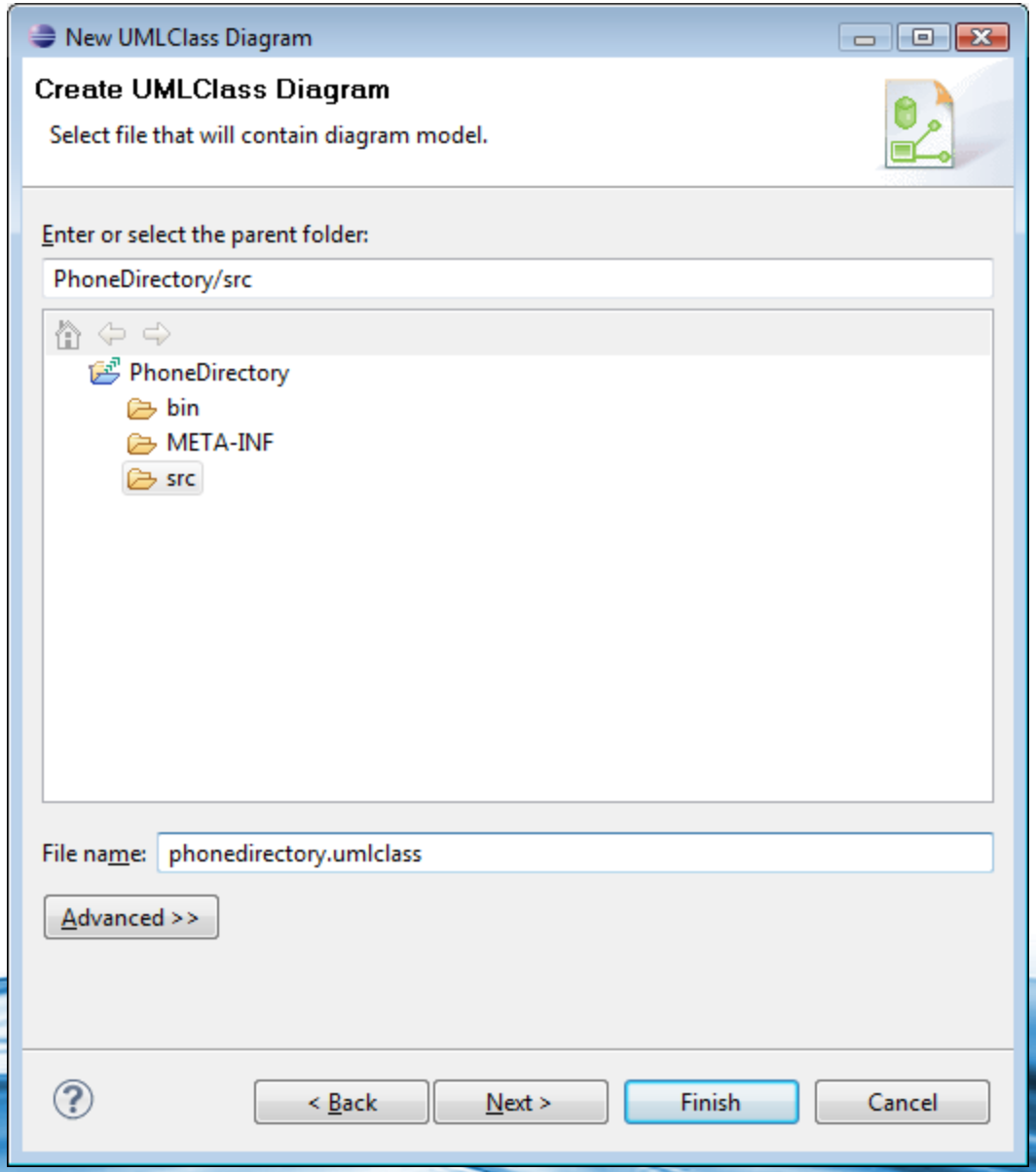
An outline is not available.

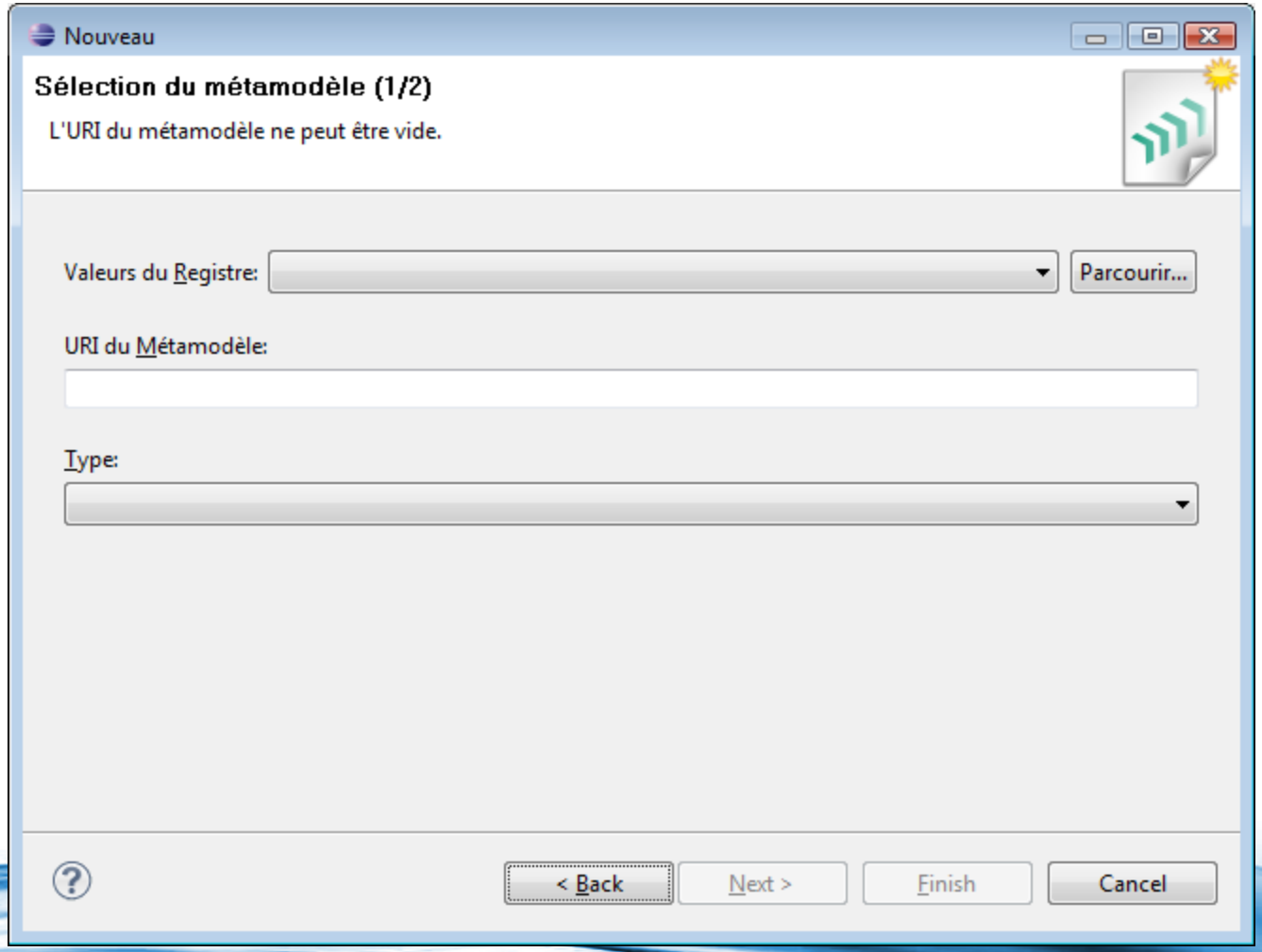
Problems

0 errors, 1 warning, 0 others

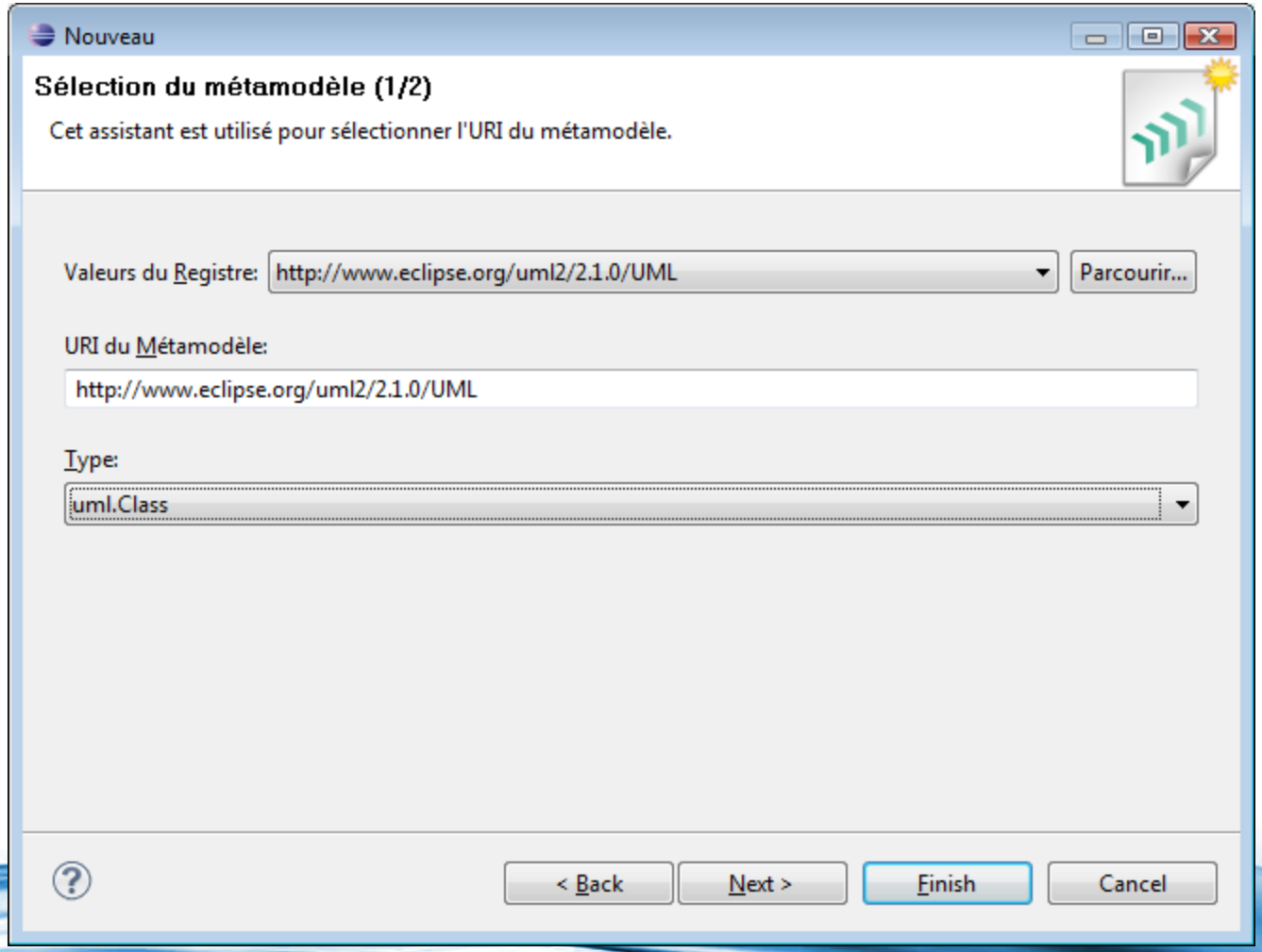
Description	Resource	Path
Warnings (1 item)		

src - PhoneDirectory

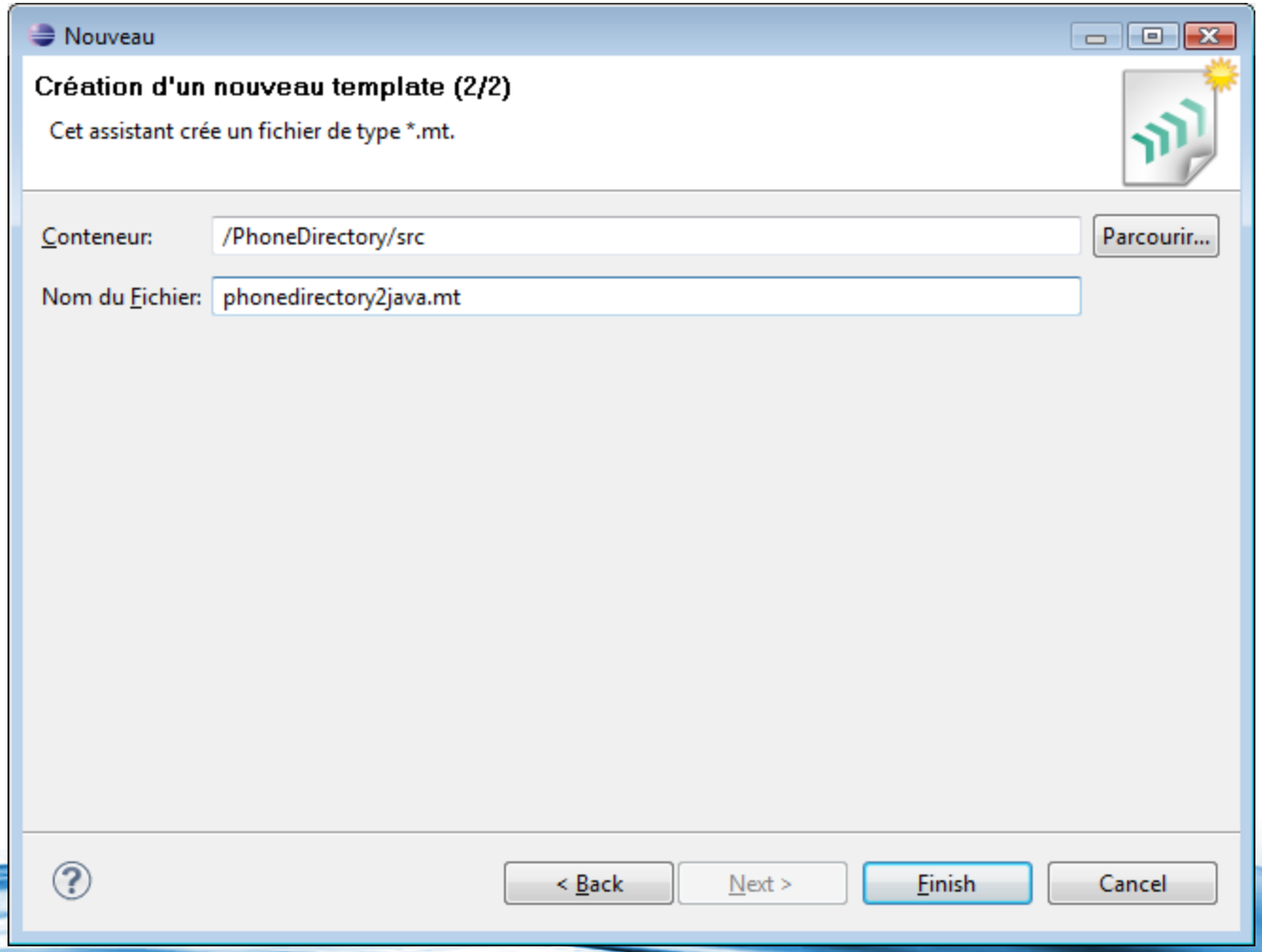

New UMLClass Diagram
Create UMLClass Diagram
Select file that will contain diagram model.
Enter or select the parent folder:
PhoneDirectory/src
PhoneDirectory
bin
META-INF
src
File name:
phonedirectory.umlclass
Advanced >>
< Back
Next >
Finish
Cancel

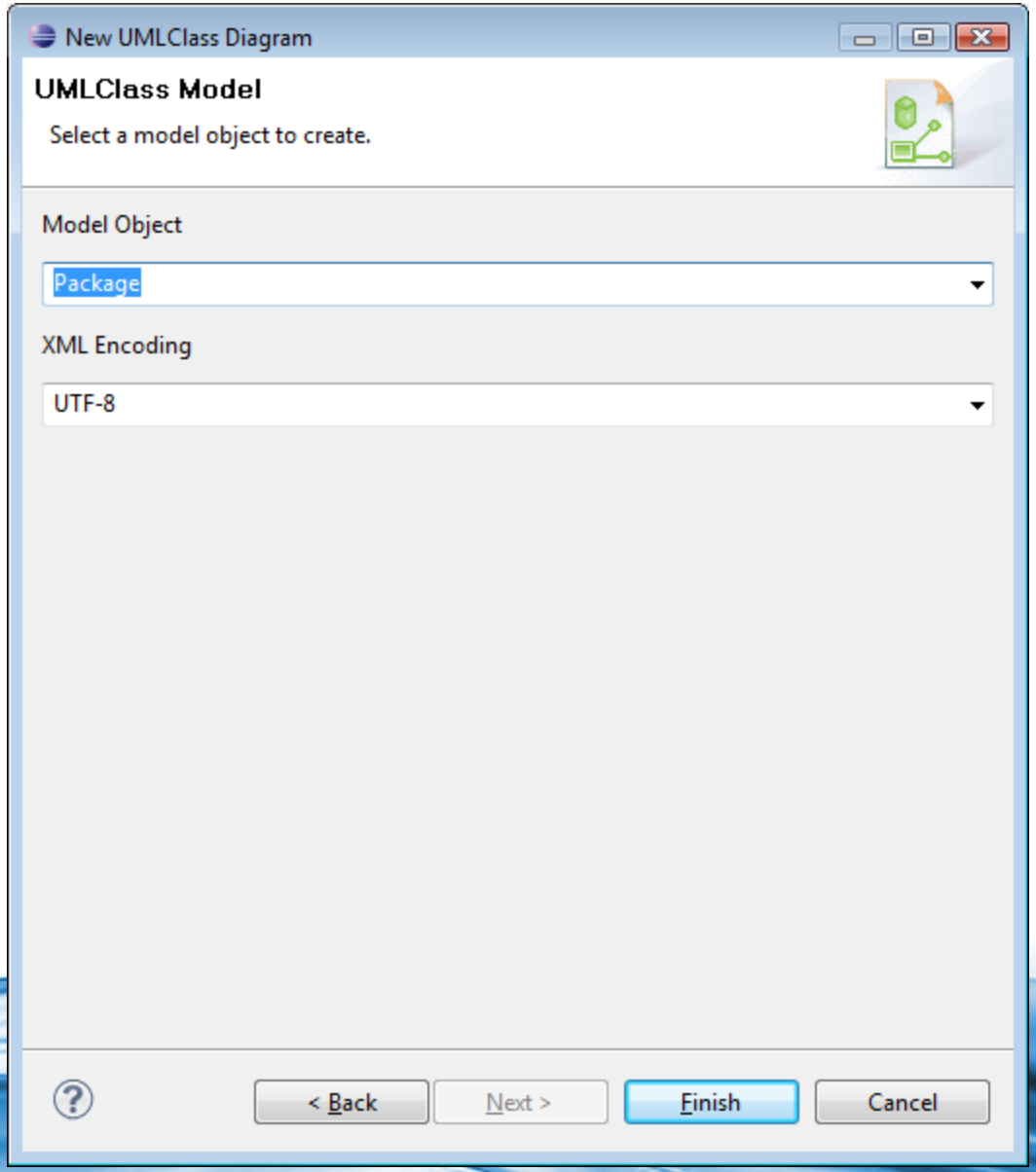

Nouveau
Sélection du métamodèle (1/2)
L'URI du métamodèle ne peut être vide.
Valeurs du Registre:
Parcourir...
URI du Métamodèle:
Type:
< Back
Next >
Finish
Cancel

Complete the dialog window as follows

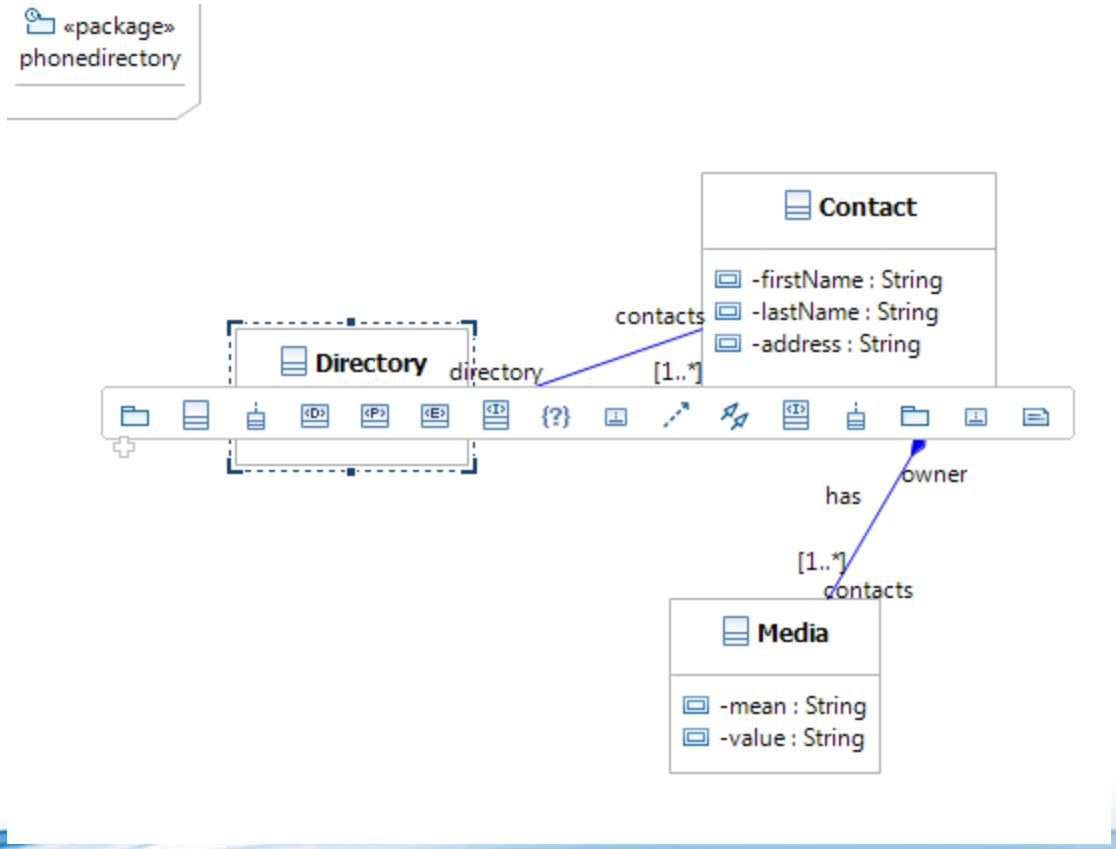


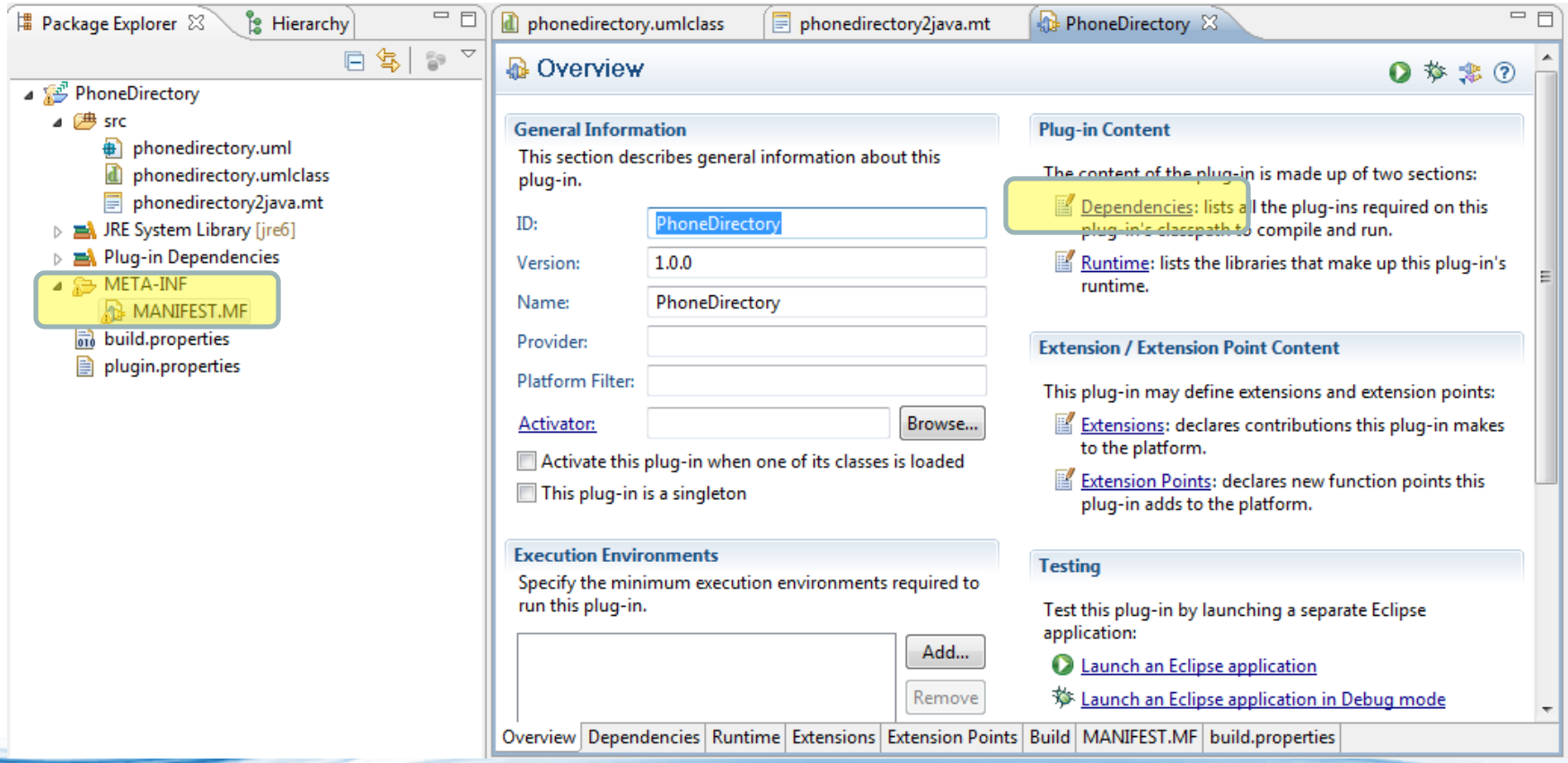
Rename the file as follows




New UMLClass Diagram
UMLClass Model
Select a model object to create.
Model Object
Package
XML Encoding
UTF-8
< Back
Next >
Finish
Cancel

Define the model as follows





We need to modify the dependencies so as to consider UML diagrams when generating code

phonedirectory.umlclass | phonedirectory2java.mt

Dependencies

Required Plug-ins

Specify the list of plug-ins required for the operation of this plug-in.

- fr.obeo.acceleo.gen

Add... | Remove | Up | Down | Properties...

Total: 1

Automated Management of Dependencies

Overview | Dependencies | Runtime | Extensions | Extension Points

Plug-in Selection

Select a Plug-in:

uml

Matching items:

- fr.obeo.acceleo.uml13 (2.6.1.200909102231)
- fr.obeo.acceleo.uml14 (2.6.1.200909102231)
- fr.obeo.acceleo.uml14.mof (2.6.1.200909102231)
- fr.obeo.acceleo.uml14.ui (2.6.1.200909102231)
- org.eclipse.m2m.atl.drivers.uml24atl (3.0.1.v200909150941)
- org.eclipse.ocl.uml (2.0.0.v200905271400)
- org.eclipse.ocl.uml.source (2.0.0.v200905271400)
- org.eclipse.uml2 (3.0.0.v200905041045)
- org.eclipse.uml2.codegen.ecore (1.5.0.v200905151700)
- org.eclipse.uml2.codegen.ecore.source (1.5.0.v200905151700)
- org.eclipse.uml2.codegen.ecore.ui (1.5.0.v200905041045)
- org.eclipse.uml2.codegen.ecore.ui.source (1.5.0.v200905041045)
- org.eclipse.uml2.common (1.5.0.v200905041045)
- org.eclipse.uml2.common.edit (1.5.0.v200905041045)
- org.eclipse.uml2.common.edit.source (1.5.0.v200905041045)

org.eclipse.uml2

OK | Cancel

phonedirectory.umlclass | phonedirectory2java.mt | PhoneDirectory

Dependencies

Required Plug-ins

Specify the list of plug-ins required for the operation of this plug-in.

- fr.obeo.acceleo.gen
- org.eclipse.uml2 (3.0.0)

Add... | Remove | Up | Down | Properties...

Total: 2

Imported Packages

Specify packages on which this plug-in depends without explicitly identifying their originating plug-in.

Add... | Remove | Properties...

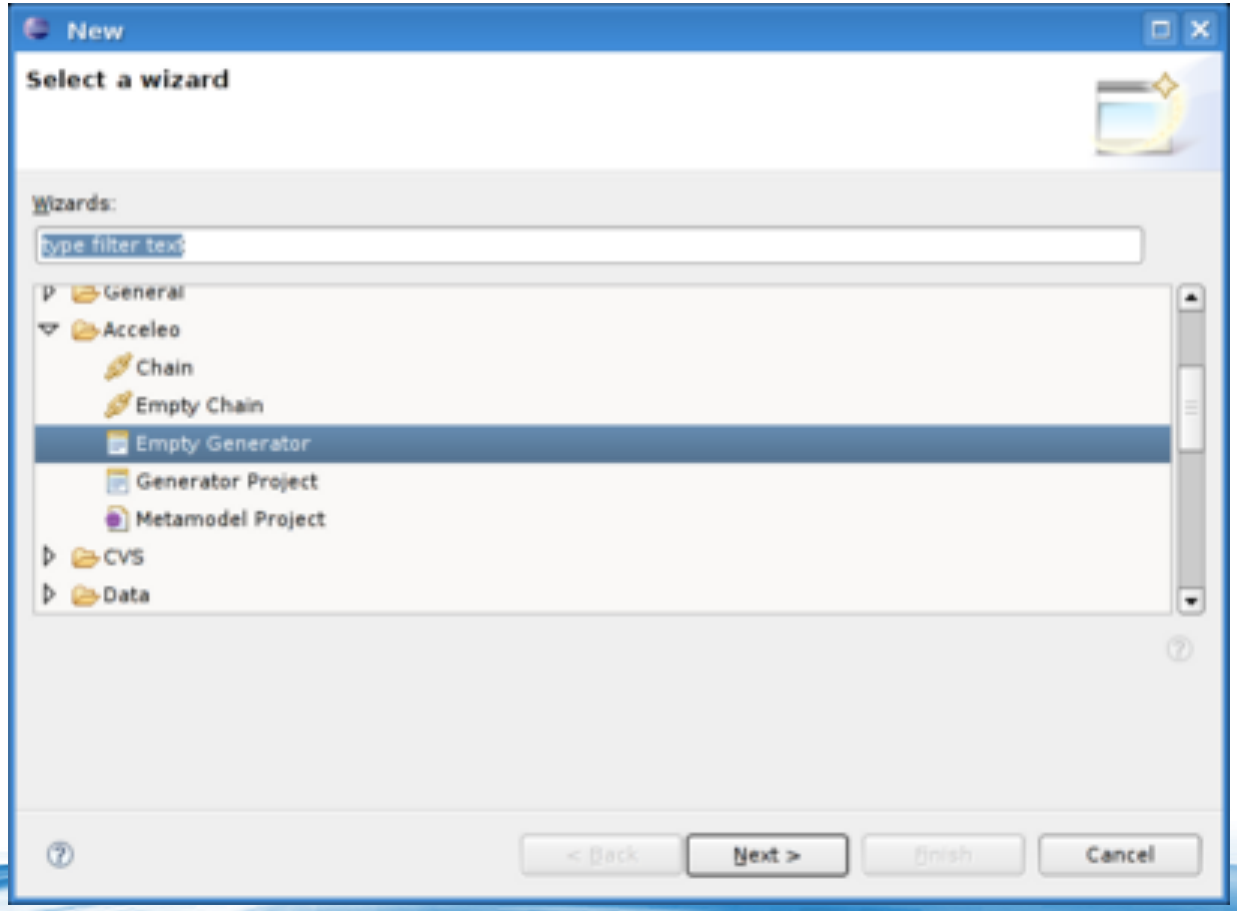
Total: 0

Automated Management of Dependencies

Dependency Analysis

Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | build.properties

Create an empty generator

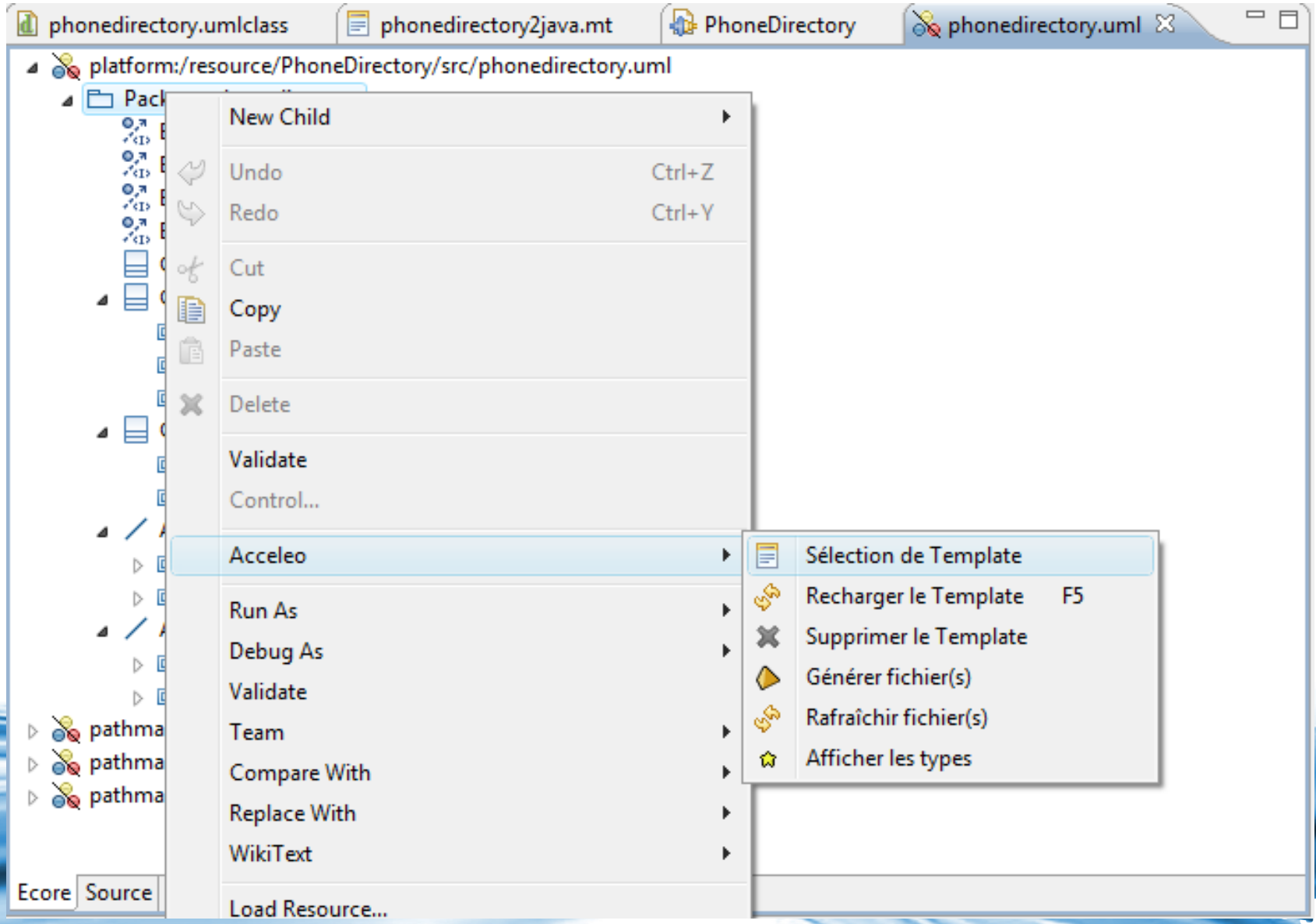


phonedirectory.umlclass | phonedirectory2java.mt | PhoneDirectory

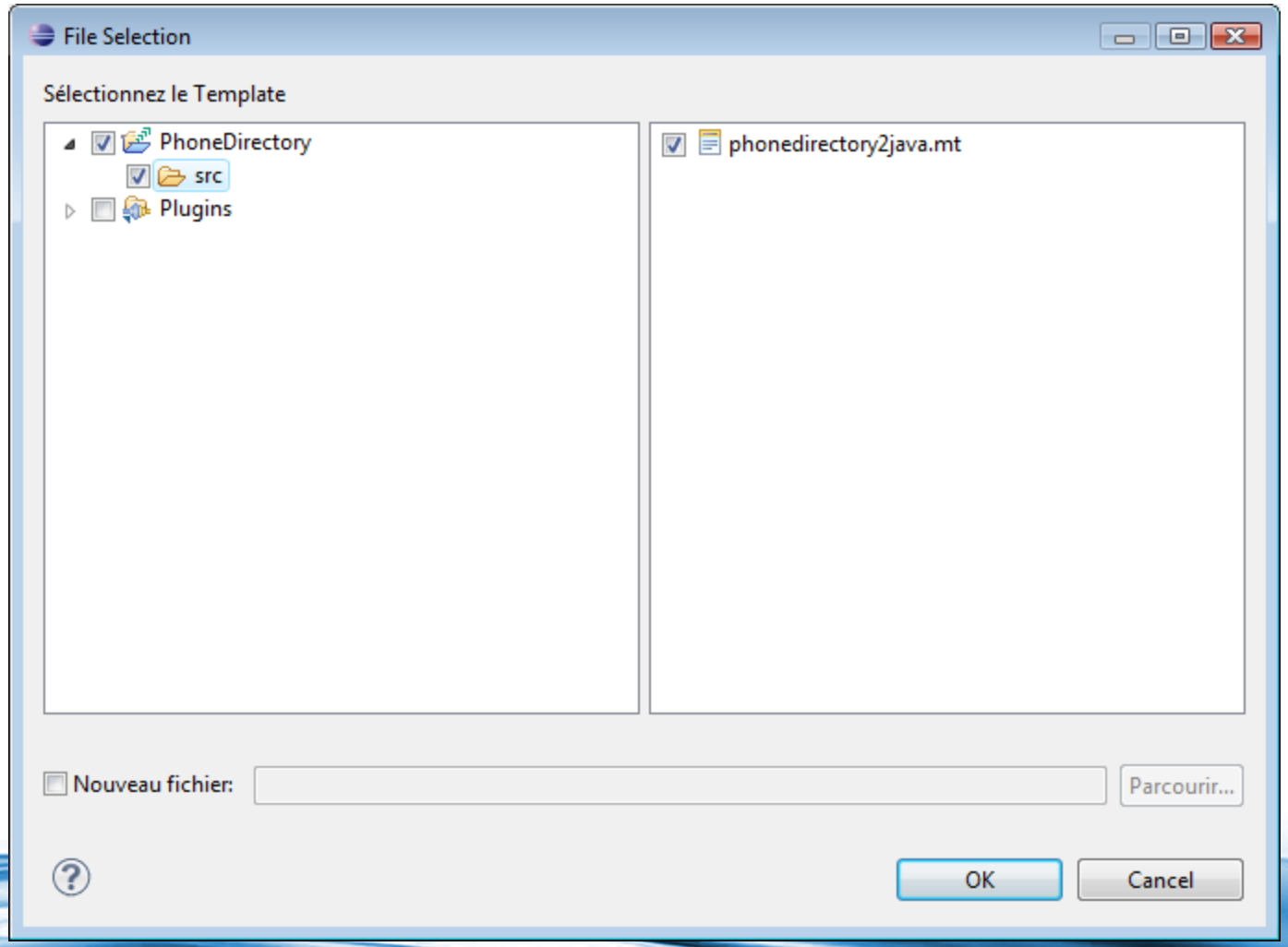
```
<%
metamodel http://www.eclipse.org/uml2/2.1.0/UML
%>

<%script type="uml.Class" name="phonedirectory2java" file="<%name%>.txt"%>
```

This template will be filled further

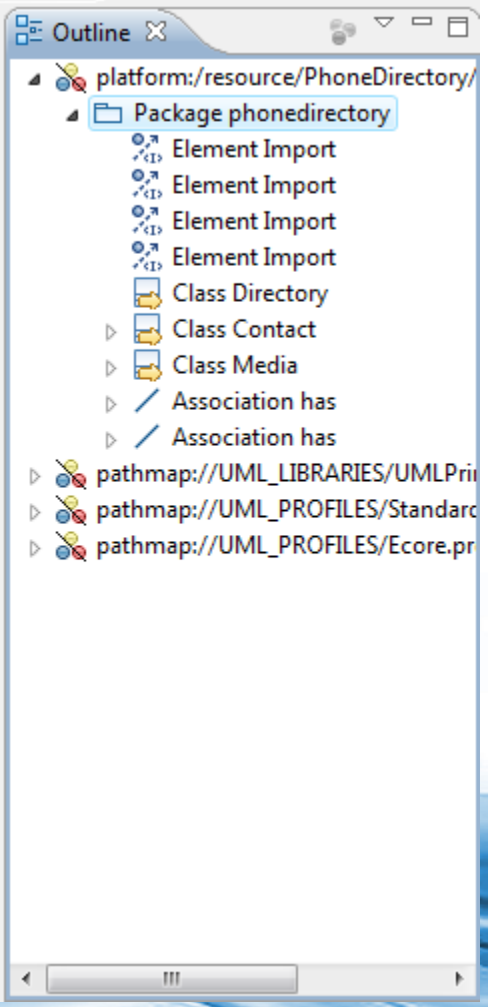
We need to associate the model to the template



The interface is bugged and we need to click on src to see the template…



If template is correctly selected, a marker appears on classes (and elements that will be considered for generation)



Modifies the template as follows

phonedirectory2java. | PhoneDirectory | phonedirectory.uml | Contact.java | 1

```
<%
metamodel http://www.eclipse.org/uml2/2.1.0/UML
%>

<%script type="uml.Class" name="phonedirectory2java" file="<%name%>.java"%>

public class <%name%> {

    <%for (attribute){%>
    private <%type.name%> <%name%>;
    <%}%>

    /*getter and setter*/
    <%for (attribute){%>
    public <%type.name%> get<%name%>() {
        return <%name%>;
    }

    public void set<%name%>(<%type.name%> a<%name%>) {
        <%name%> = a<%name%>;
    }
    <%}%>


}
```

File generation

The last stage is generating files, right click on generator project>Acceleo>Generate files

Here we are…

phonedirectory2java. | PhoneDirectory | phonedirectory.uml | Contact.java

```java
public class Contact {

        private String firstName;
        private String lastName;
        private String address;

    /*getter and setter*/
        public String getfirstName() {
            return firstName;
        }

        public void setfirstName(String afirstName) {
            firstName = afirstName;
        }
        public String getlastName() {
            return lastName;
        }

        public void setlastName(String alastName) {
            lastName = alastName;
        }
        public String getaddress() {
            return address;
        }

        public void setaddress(String address) {
```

Acceleo

Advantages

- Easy to use interface
- Models are manipulated as UML class diagrams
- Scripting ease file generation

Drawbacks

- This is not a 100% compliant tool:
  - No distinction between models and instances
  - Inability to change the metamodel
  - No transformation rules
- Documentation is difficult to find

But it works and this is all we need…
When requirements are low…

Labs on Acceleo (MDA)

- Install Acceleo on your machine
- Optional (just to get your feet wet)
  - Do the example given in previous slides
- Prepare a metamodel corresponding to a University (students, lectures, rooms, teaching staff)
- Create scripts to generate files
- Generate files

Model-Driven Engineering (MDE)

Introduction

- MDA proves its usefulness when generating code **directly** (this is what we saw with Acceleo)
- MDA is a reality in UML commercial and open source tools: we can define modules and generate code for a specific platform
- MDA is unfortunately too narrow and dedicated to UML

Model-Driven Engineering

- MDE extends MDA so as not to restrict to UML metamodel and MOF meta-metamodel
- MDE keeps the philosophy of MDA but enlarges it:
  - No predefined meta-metamodel
  - Metamodel can be defined
  - The four modeling layers of OMG really exist
  - Transformation rules are used

Model-driven Engineering (cont'd)

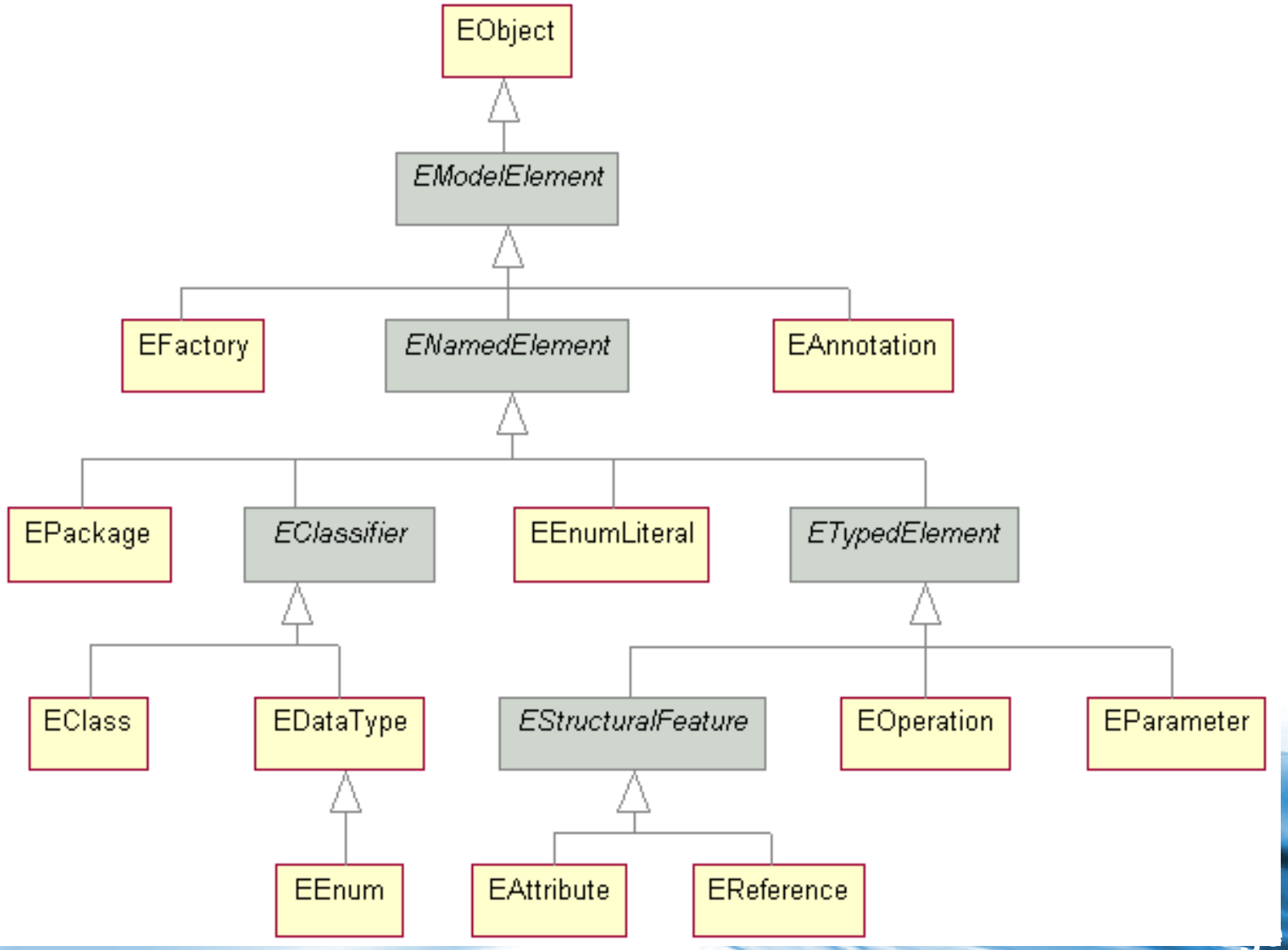
What you will consider in MDE:

- MOF or Ecore
- Eclipse Modeling Framework
- M2M via ATL
- M2T via template engine

Ecore

- Proposed in the context of Eclipse Modeling Framework (EMF)
- Another meta-metamodel
- Strong similarity with MOF except:
  - Simpler
  - No association between classes, associations are defined as attributes within classes

Ecore


EObject
EModelElement
EFactory
ENamedElement
EAnnotation
EPackage
EClassifier
EEnumLiteral
ETypedElement
EClass
EDataType
EStructuralFeature
EOperation
EParameter
EEnum
EAttribute
EReference

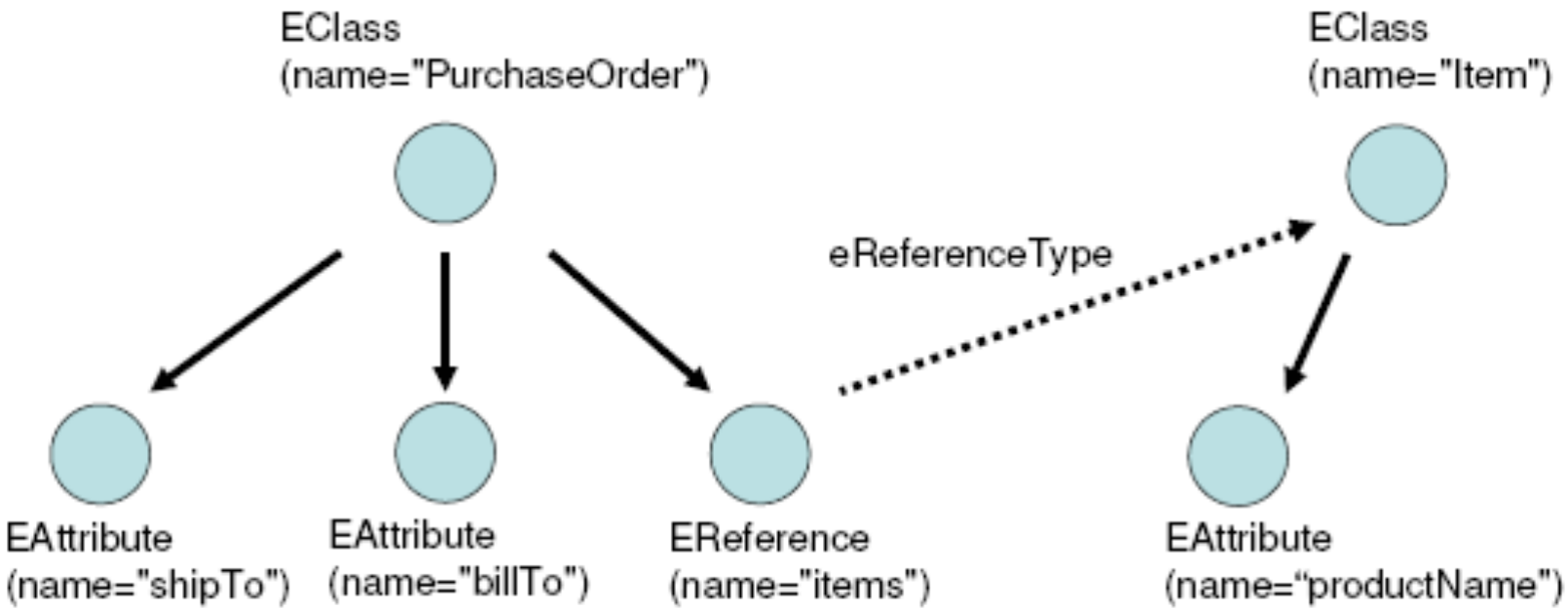
Ecore

- Every concept is prefixed with an ‘E’
- We find the usual notions of UML class diagrams:
  - Packages with EPackage
  - Classes with EClass
  - Attributes with EAttribute
  - Operations with EOperation
  - Associations with EReference

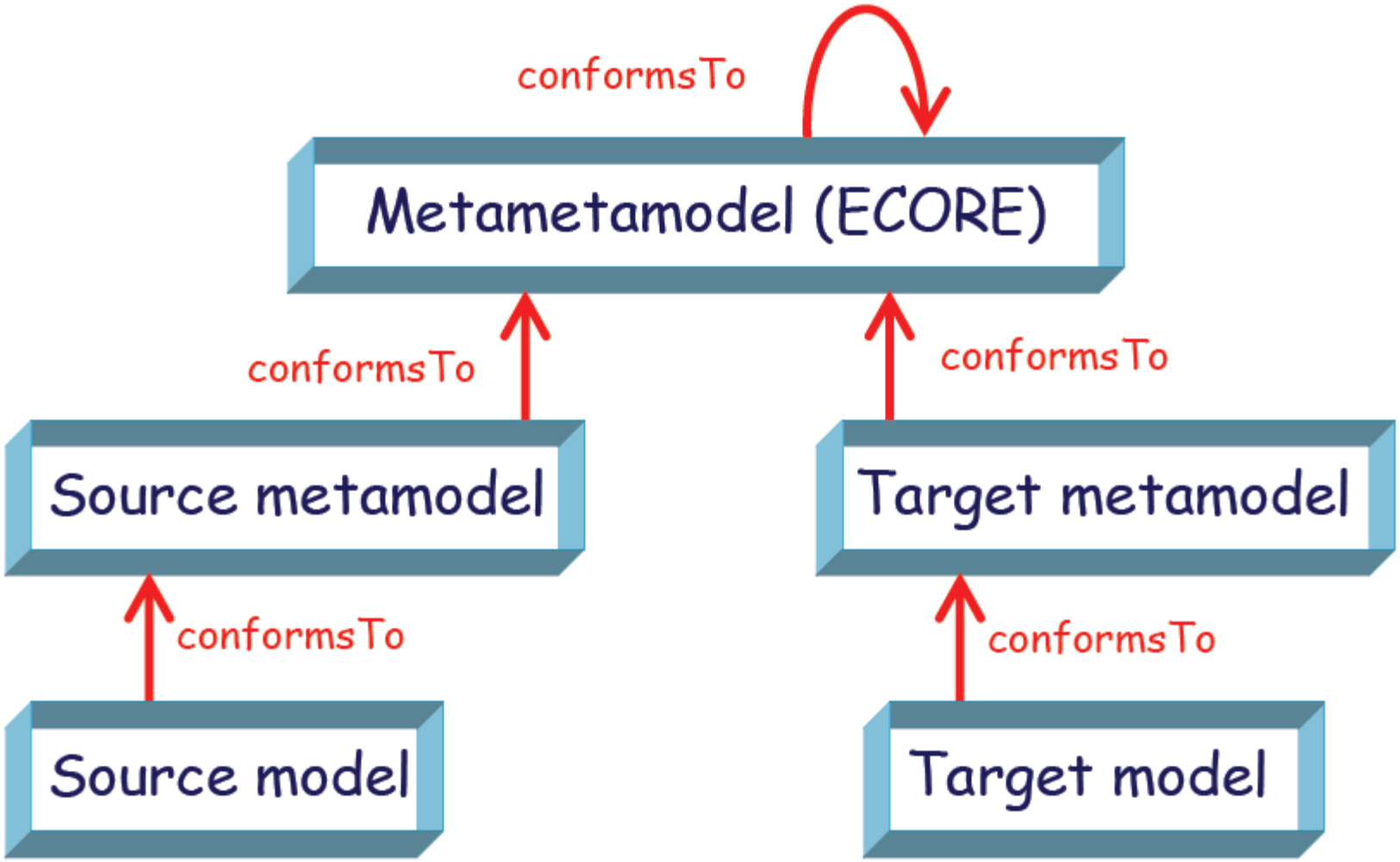
Ecore

- EFactory and EObject are used for the M0 layer: instances

Ecore example



The general picture

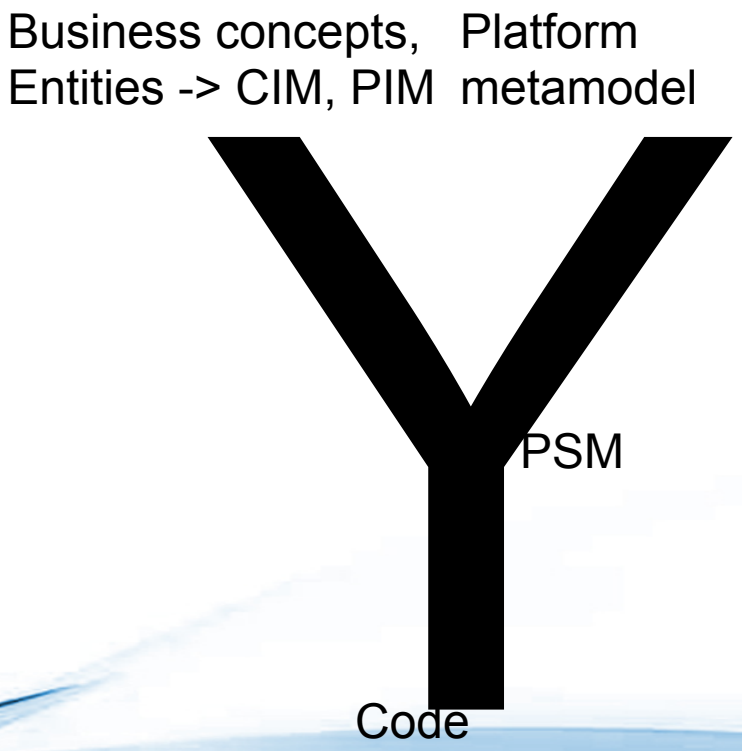


MDE Process

1. Create the **source metamodel** via Eclipse EMF
2. Generate an editor to create the **source model** (instance)
3. Create the **source model**
4. Create the **target metamodel**
5. Create the transformation rules
6. Apply the transformation rules and get the **target model** (instance)

We will use this approach in Kermeta

The Y development cycle



MDE Process in several flavors

- Pure MDE Process:
  - Source and target metamodels with Ecore or MOF
  - Transformation rules with ATL
  - Source and target models as an XMI file

  Only three modeling layers in this case (M3, M2, M0)

Tool: Kermeta

MDE Process in several flavors (cont'd)

- Extreme Purity:
  - Source and target metamodels with Ecore or MOF
  - Source and target models with Ecore or MOF
  - Source and target instances with XMI
  - Transformation rules with ATL

The four modeling layers in this approach

Tool: EMF

MDE Process in several flavors (cont'd)

- Flexible attitude:
  - Source and target metamodels with Ecore or MOF
  - *Source and target models with Ecore or MOF*
  - Source and target instances with XMI
  - Transformation rules with Java

Tool: Dynamic EMF + Velocity + Java

Tools for MDE

- Kermeta (http://www.kermeta.org)
- OpenArchitectureWare: retired (http://openarchitectureware.org)
- OpenEmbedd (http://openembedd.inria.fr)
- TopCased (http://www.topcased.org)

OpenArchitectureWare

- Tools proposed around OAW are now part of Eclipse Modeling tools
- Retired, tools are now used in Eclipse EMF

OpenEmbedd

- MDE platform for real-time and embedded systems
- Based on Eclipse and Eclipse modeling tools
- Strong industrial tool with academic partners specialist in Model-Driven Engineering

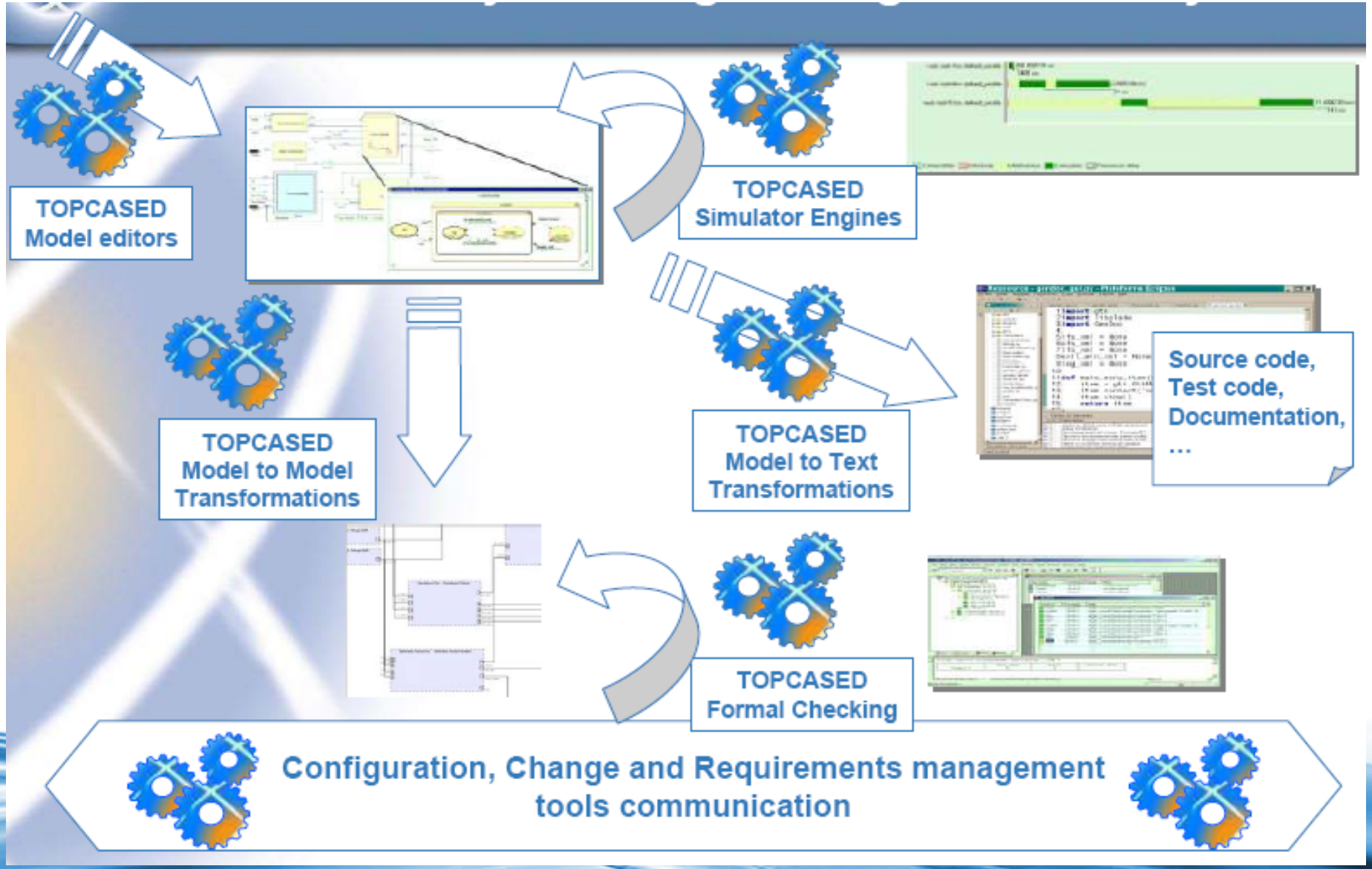
The different tools used in OpenEmbedd are the ones in Kermeta and EMF

TopCased

- MDE platform for critical systems
- Strong support from industry
- Based on Eclipse (Eclipse RCP, Eclipse Modeling tools)
- Specific tools from partners: model checking, analysis, etc.

The different tools used in OpenEmbedd are the ones in Kermeta and EMF

TOPCASED
Toolkit in Open-Source for Critical Application & Systems Development


TOPCASED
Model editors
TOPCASED
Simulator Engines
TOPCASED
Model to Model
Transformations
TOPCASED
Model to Text
Transformations
Source code,
Test code,
Documentation,
…
TOPCASED
Formal Checking
Configuration, Change and Requirements management
tools communication

MDE in practice

Kermeta

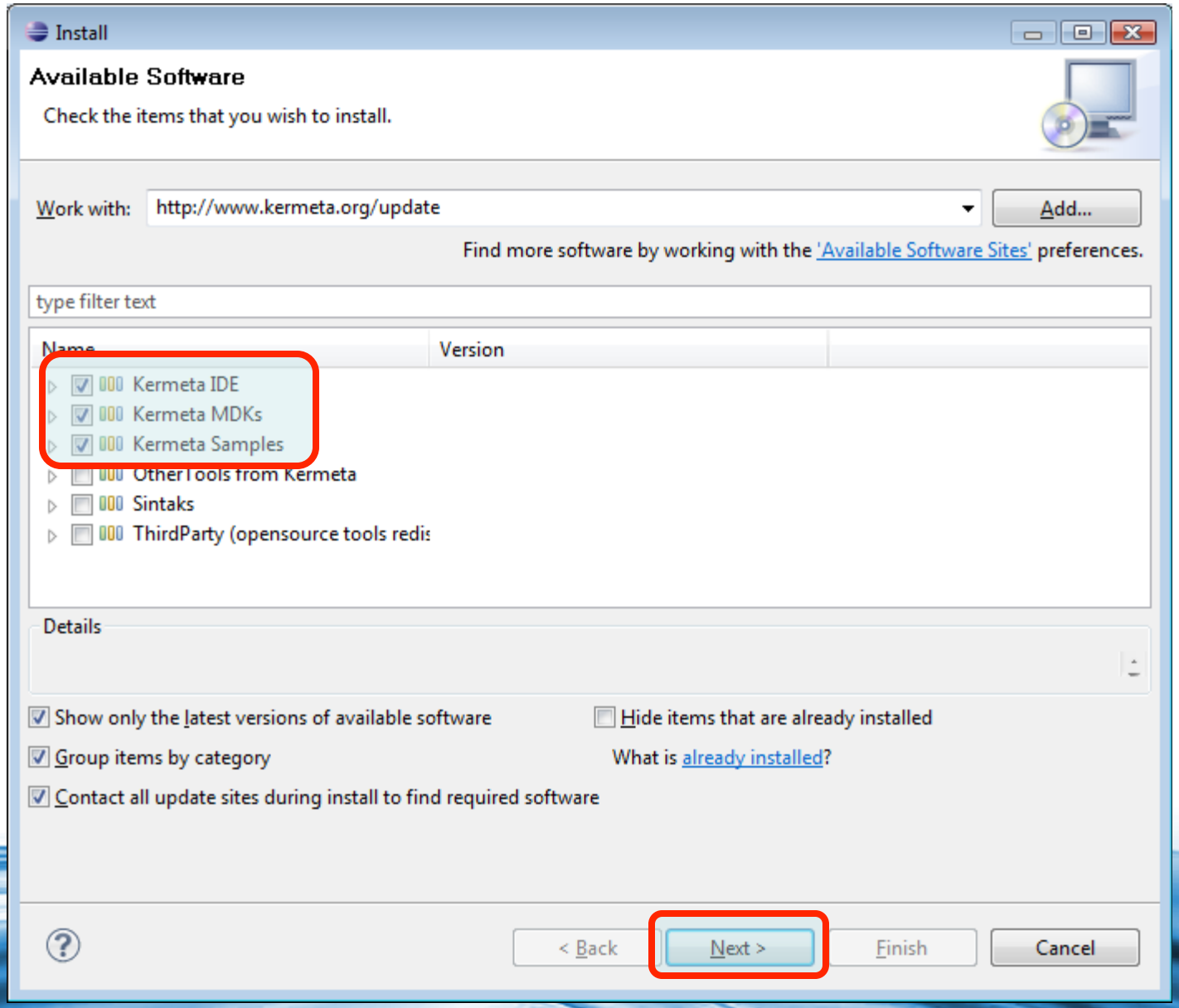
- MDE platform
- Based on Eclipse
- Kermeta = Metamodeling + Behavior

Installation

- Download All-in-one Eclipse Modeling Tools (http://www.eclipse.org/downloads/packages/eclipse-modeling-tools-includes-incubating-components/galileosr1)
- Unzip the archive to a directory
- Execute Eclipse
- Check in File>New you have Eclipse Modeling Framework and UML 2.1 Diagrams

Installation (cont'd)

- Go to Help>Install new software
- Click on Button *Add* and this new site: http://www.kermeta.org/update
- Then select the Kermeta site in the combo *Work with*
- Check the Kermeta proposal below as shown on Figure


Install
Available Software
Check the items that you wish to install.
Work with:
http://www.kermeta.org/update
Add...
Find more software by working with the 'Available Software Sites' preferences.
type filter text
Name
Version
Kermeta IDE
Kermeta MDKs
Kermeta Samples
OtherTools from Kermeta
Sintaks
ThirdParty (opensource tools redis
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

Install

Install Details

Review the items to be installed.

Name	Version	Id
Finite State Machine Kermeta sample	1.3.0	fr.irisa.triskell.kermeta.samples.fsm.featur...
Kermeta compiler	1.3.2	org.kermeta.compiler.feature.feature.group
Kermeta Emitter Templates	1.3.2	fr.irisa.triskell.kermeta.ket.feature.feature....
Kermeta Graphical Editor	1.3.2	fr.irisa.triskell.kermeta.graphical.feature.gr...
Kermeta MDK for Ecore	1.3.2	fr.irisa.triskell.kermeta.ecore.feature.group
Kermeta MDK for OCL	1.3.2	fr.irisa.triskell.kermeta.ocl.feature.group
Kermeta MDK for Traceability	1.3.2	fr.irisa.triskell.traceability.feature.group
Kermeta MDK for UML2	1.3.2	org.kermeta.uml2.feature.group
Kermeta tutorial on Aspect Programm	1.3.2	org.kermeta.tutorial.aspects.feature.group
Kermeta Workbench	1.3.2	fr.irisa.triskell.kermeta.feature.group
Kompose	0.0.5	org.kermeta.kompose.feature.feature.group
KTR Feature	1.0.1	fr.irisa.triskell.ktr.feature.feature.group
Logo language Kermeta sample	1.3.2	org.kermeta.kmlogo.feature.feature.group
Marte profile for UML MDK	0.1.0	org.kermeta.uml2.profile.marte.feature.gr...
Moncom process support	1.0.0	fr.moncom.wfr.feature.feature.group

Size: 49 182 KB

Details

< Back | Next > | Finish | Cancel

Install

Review Licenses

Licenses must be reviewed before the software can be installed. This includes licenses for software required to complete the install.

Items with licenses:

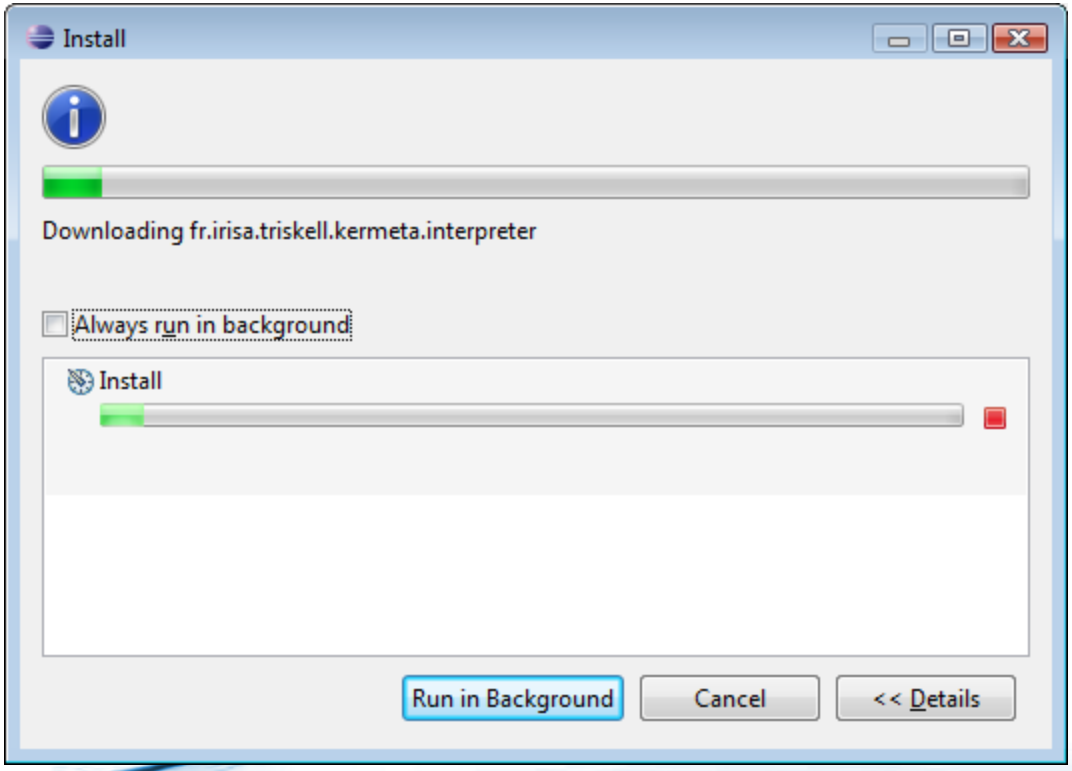
Name	Version
Finite State Machine Kermeta sample	1.3.0
Kermeta compiler	1.3.2
Kermeta Emitter Templates	1.3.2
Kermeta Graphical Editor	1.3.2
Kermeta MDK for Ecore	1.3.2
Kermeta MDK for OCL	1.3.2
Kermeta MDK for Traceability	1.3.2
Kermeta MDK for UML2	1.3.2
Kermeta tutorial on Aspect Programmi...	1.3.2
Kermeta Workbench	1.3.2
Kompose	0.0.5
KTR Feature	1.0.1
Log4j Feature	1.2.0
Logo language Kermeta sample	1.3.2
Marte profile for UML MDK	0.1.0
Mopcom process support	1.0.0
OpenEmbeDD graphical modeler for L...	1.0.0

License text:

ECLIPSE FOUNDATION SOFTWARE USER AGREEMENT
March 17, 2005
Usage Of Content
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR
OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT").
USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF
THIS
AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS
OR
NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT,
YOU
AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT

- (selected) I accept the terms of the license agreements
- I do not accept the terms of the license agreements

< Back | Next > | Finish | Cancel


Install
Downloading fr.irisa.triskell.kermeta.interpreter
Always run in background
Install
Run in Background
Cancel
<< Details

Java - Eclipse

File Edit Navigate Search Project Run Window Help

Welcome

Overview

Kermeta

Find out how to enhance your model experience using executable metamodeling.

Workbench basics

Learn about basic Eclipse workbench concepts

Team support

Find out how to collaborate with other developers

Mint

Get the most out of JDT when developing EMF-based software.

Usage Data Collector

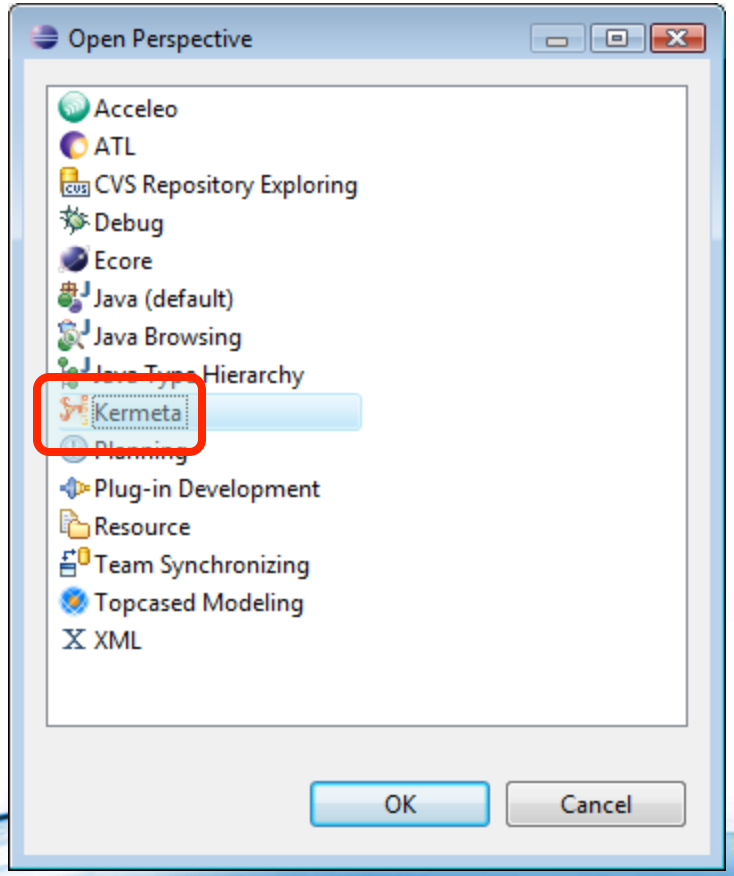
The Usage Data Collector collects information about how you are using the Eclipse platform.

Java development

Get familiar with developing Java programs using Eclipse

Eclipse plug-in development

Learn how to extend Eclipse by building new plug-ins



Change the perspective to the Kermeta one

Creating the source metamodel

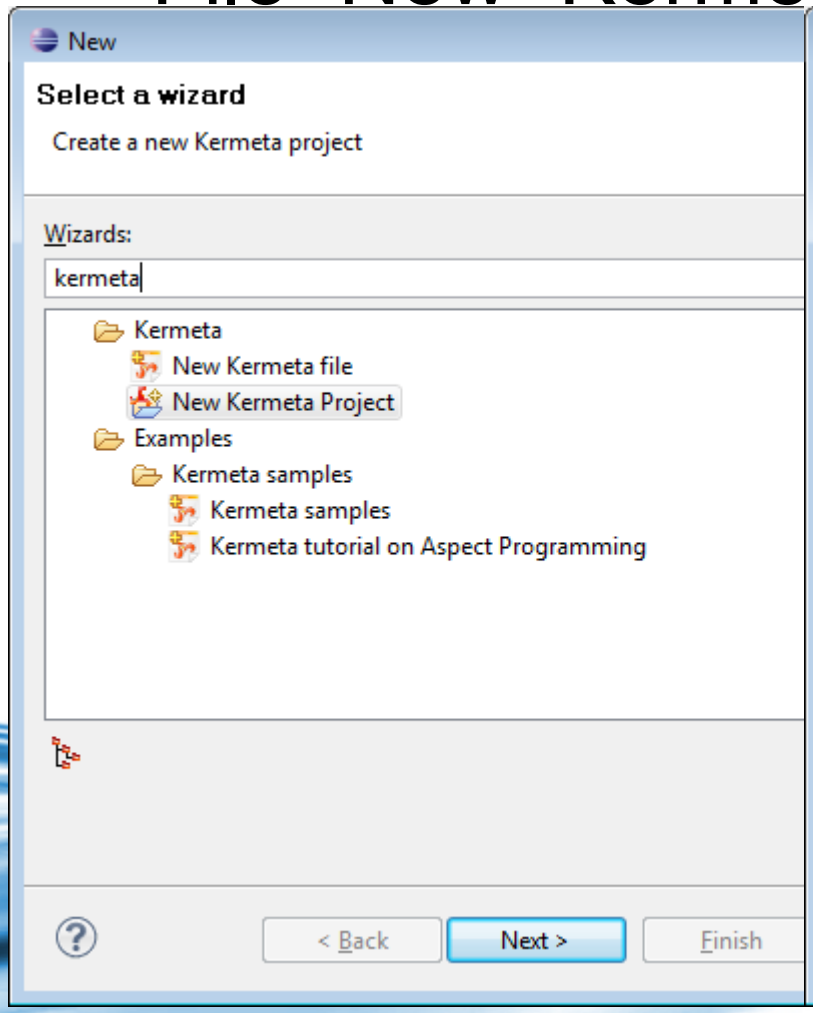
Two possibilities:

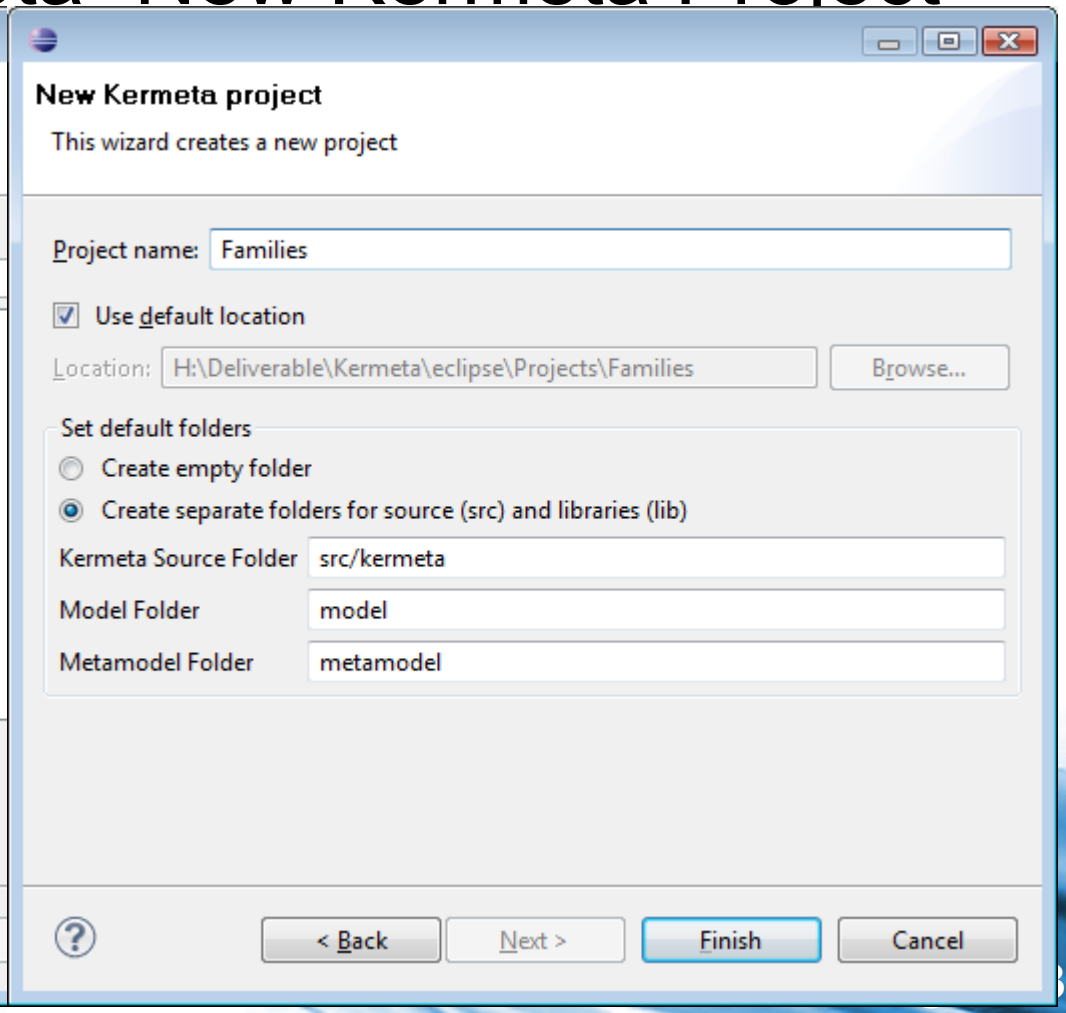
- From a file using the Kermeta notation (KM3)
- From an Ecore file

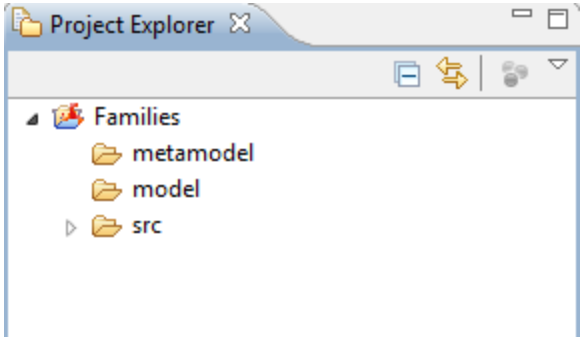
Or you could be more exotic with annotated Java files, UML class diagrams, or Rational Rose models

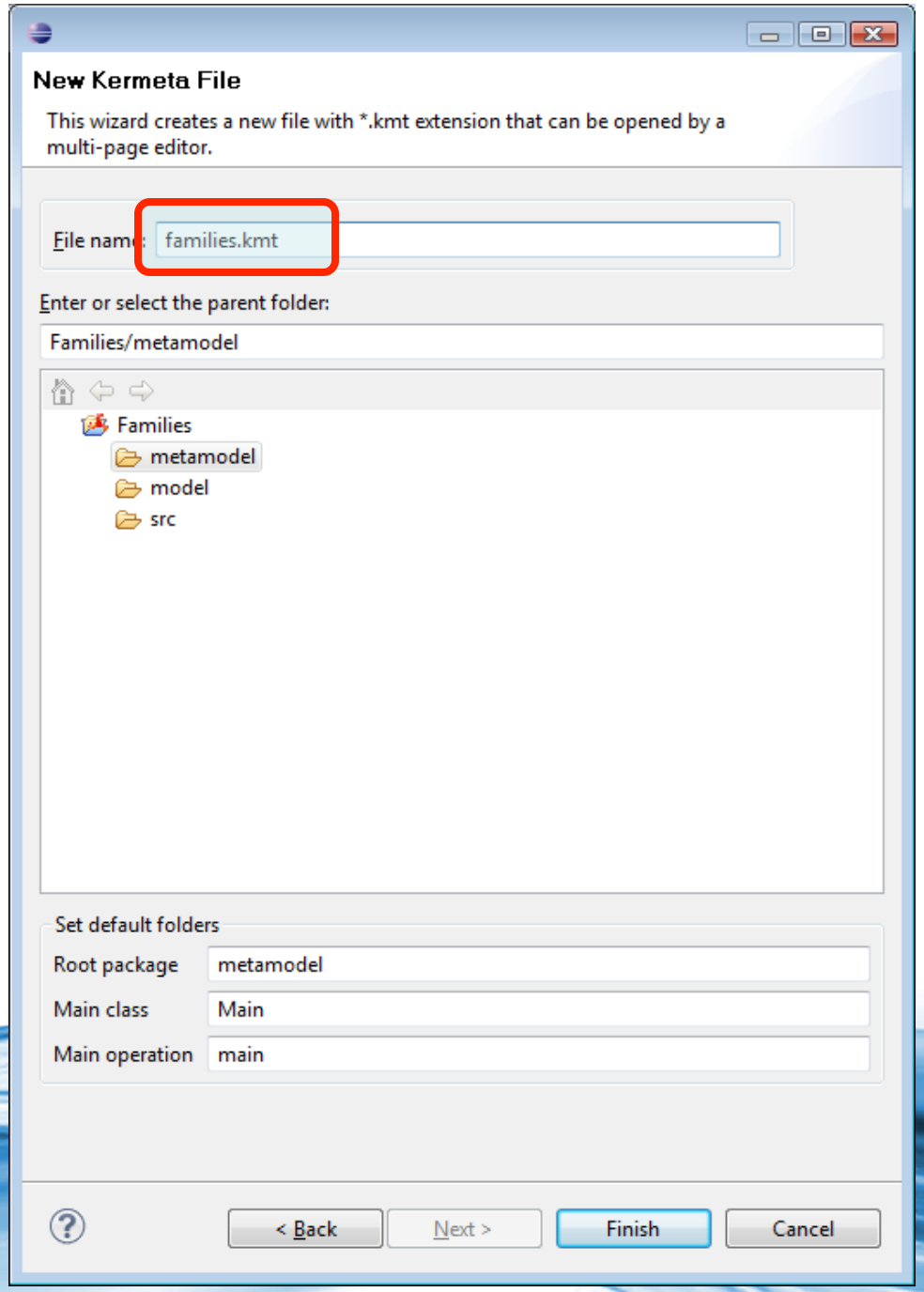
Creating from a Kermeta file

Create an empty project from File>New>Kermeta>New Kermeta Project






Project Explorer
Families
metamodel
model
src

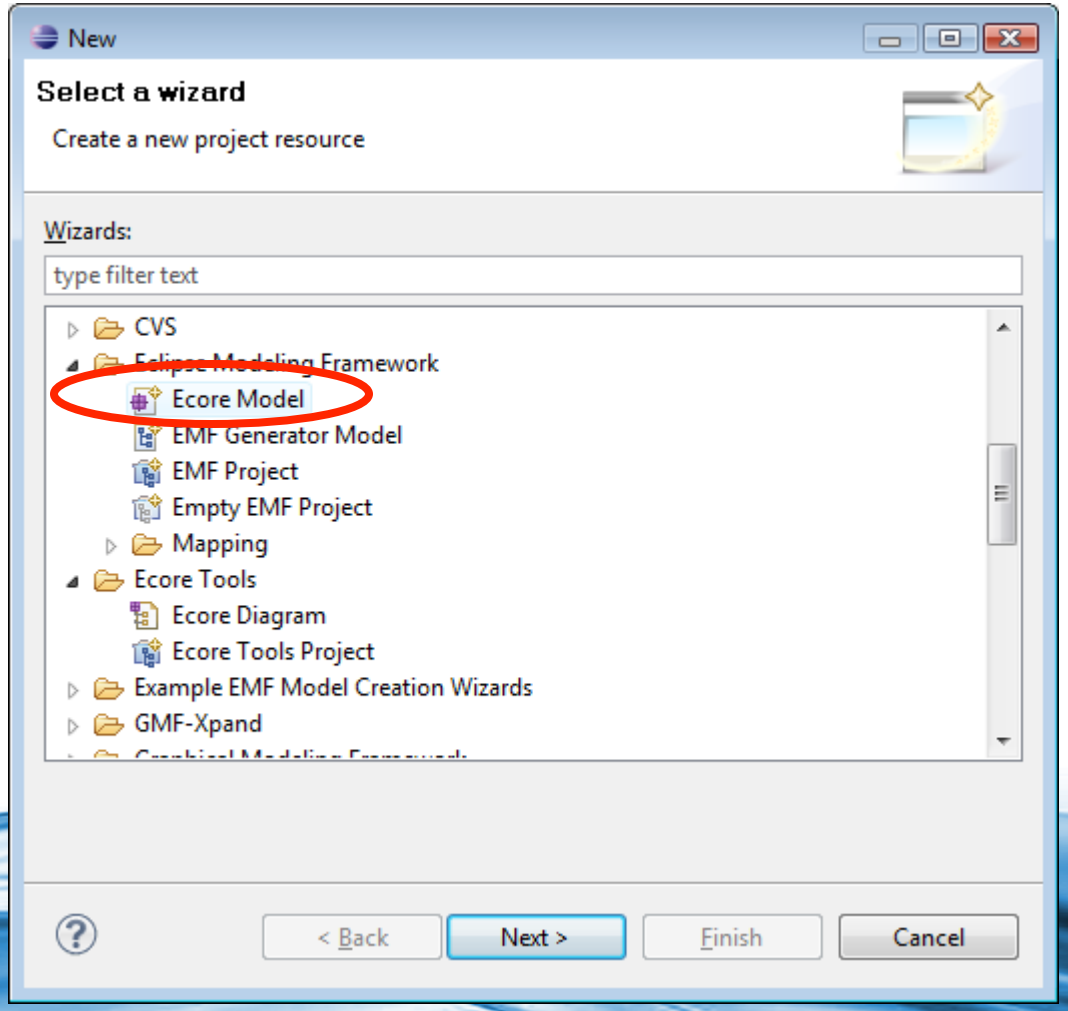

New Kermeta File
This wizard creates a new file with *.kmt extension that can be opened by a multi-page editor.
File name:
families.kmt
Enter or select the parent folder:
Families/metamodel
Families
metamodel
model
src
Set default folders
Root package
metamodel
Main class
Main
Main operation
main
< Back
Next >
Finish
Cancel

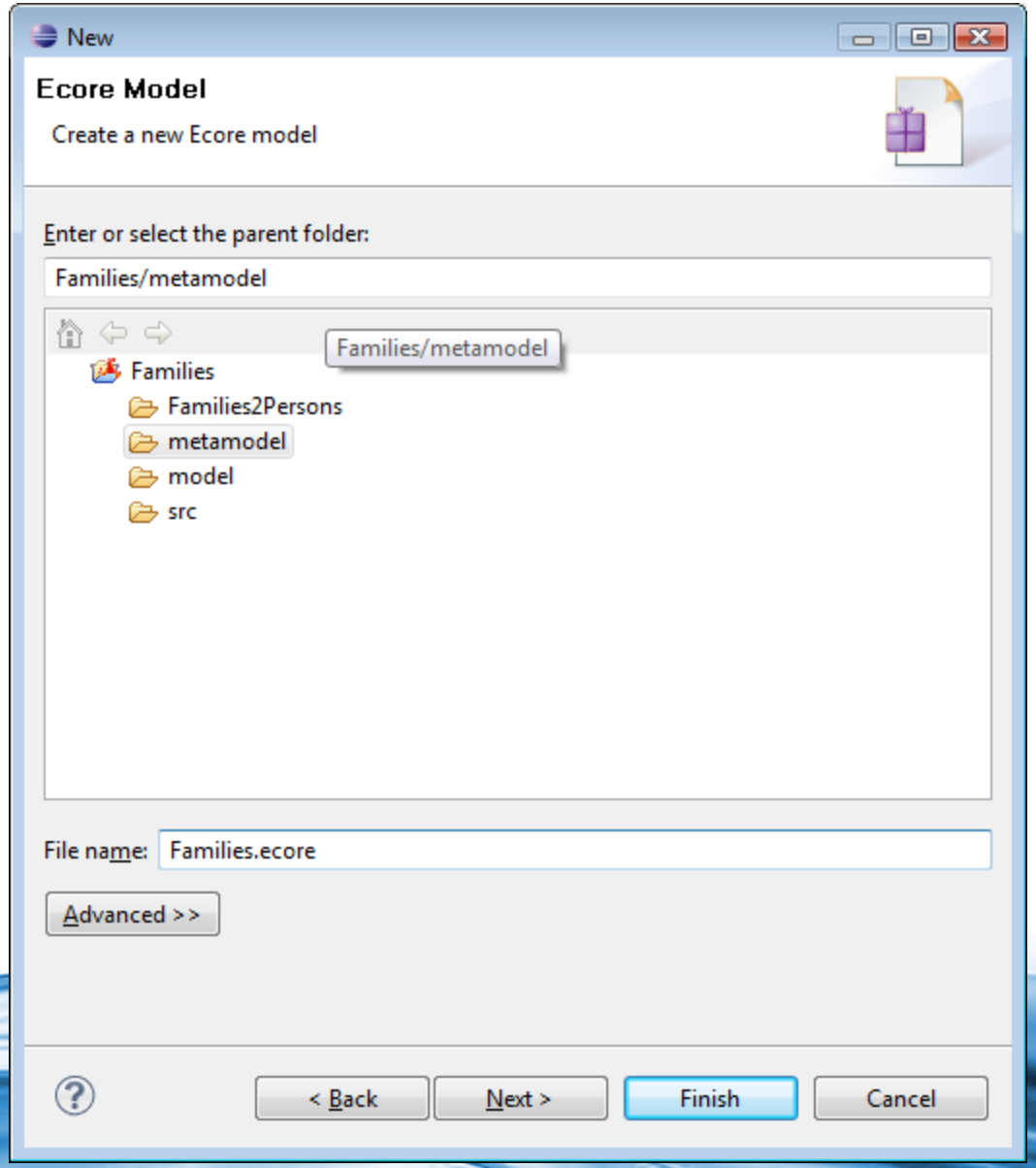
families.kmt

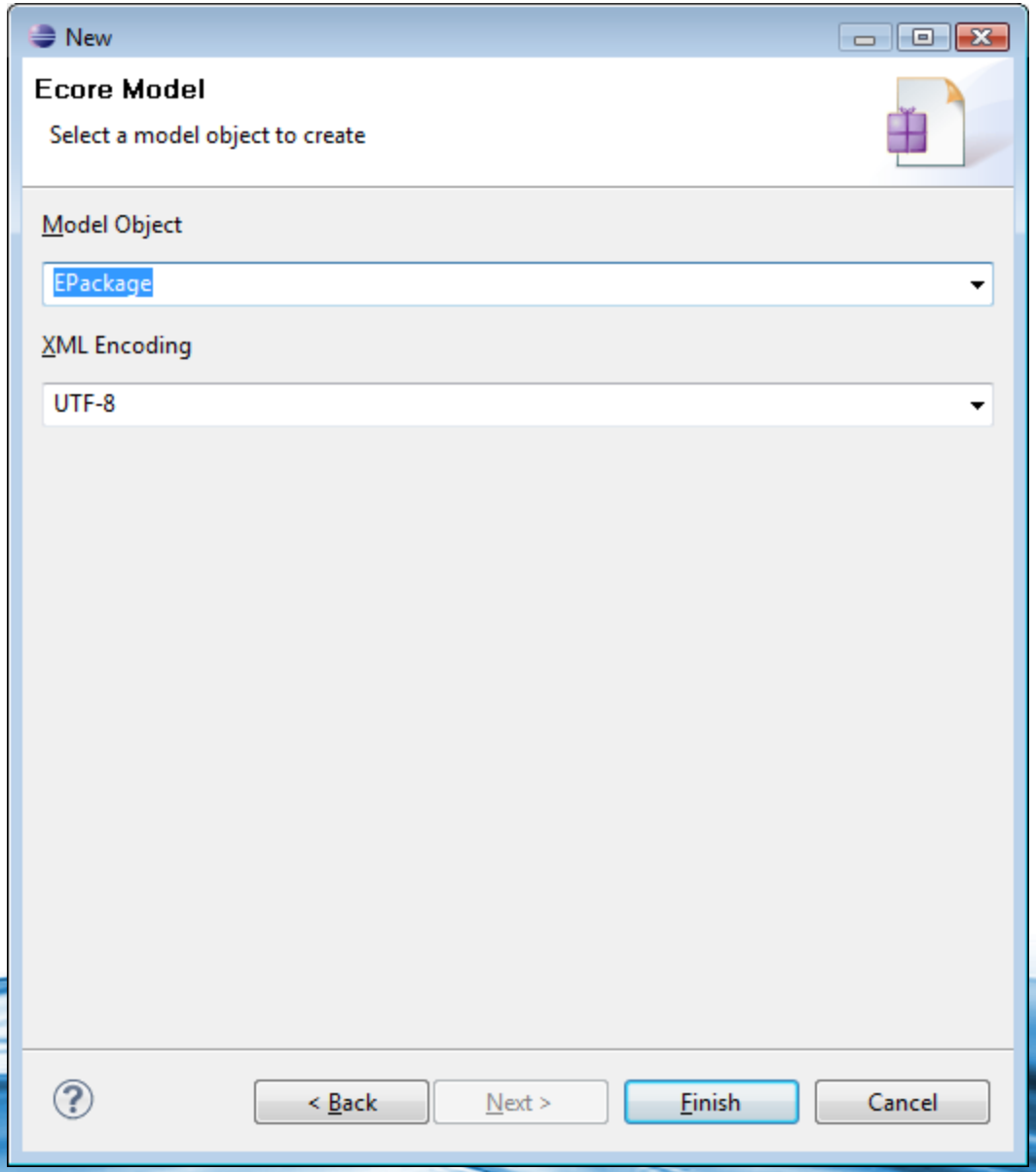
```
package metamodel;

require kermeta
using kermeta::standard

class Member {
    attribute firstName : String
    reference familyFather : Family
    reference familyMother : Family
    reference familySons : Family
    reference familyDaughters : Family
}

class Family {
    attribute lastName : String
    reference father : Member
    reference mother : Member
    reference sons : set Member[0..*]
    reference daughters : set Member[0..*]
}
```

Translate this Kermeta file into an Ecore file
It is preferable to use an Ecore for our specific needs

Create from an Ecore file




New
Ecore Model
Create a new Ecore model
Enter or select the parent folder:
Families/metamodel
Families/metamodel
Families
Families2Persons
metamodel
model
src
File name:
Families.ecore
Advanced >>
< Back
Next >
Finish
Cancel


New
Ecore Model
Select a model object to create
Model Object
EPackage
XML Encoding
UTF-8
< Back
Next >
Finish
Cancel

ATL - Families/metamodel/Families.ecore - Eclipse

File Edit Navigate Search Project Run Sample Ecore Editor Compatibility Window Help

Project Explorer

- Families
  - metamodel
    - Families.ecore
    - families.kmt
  - model
  - src

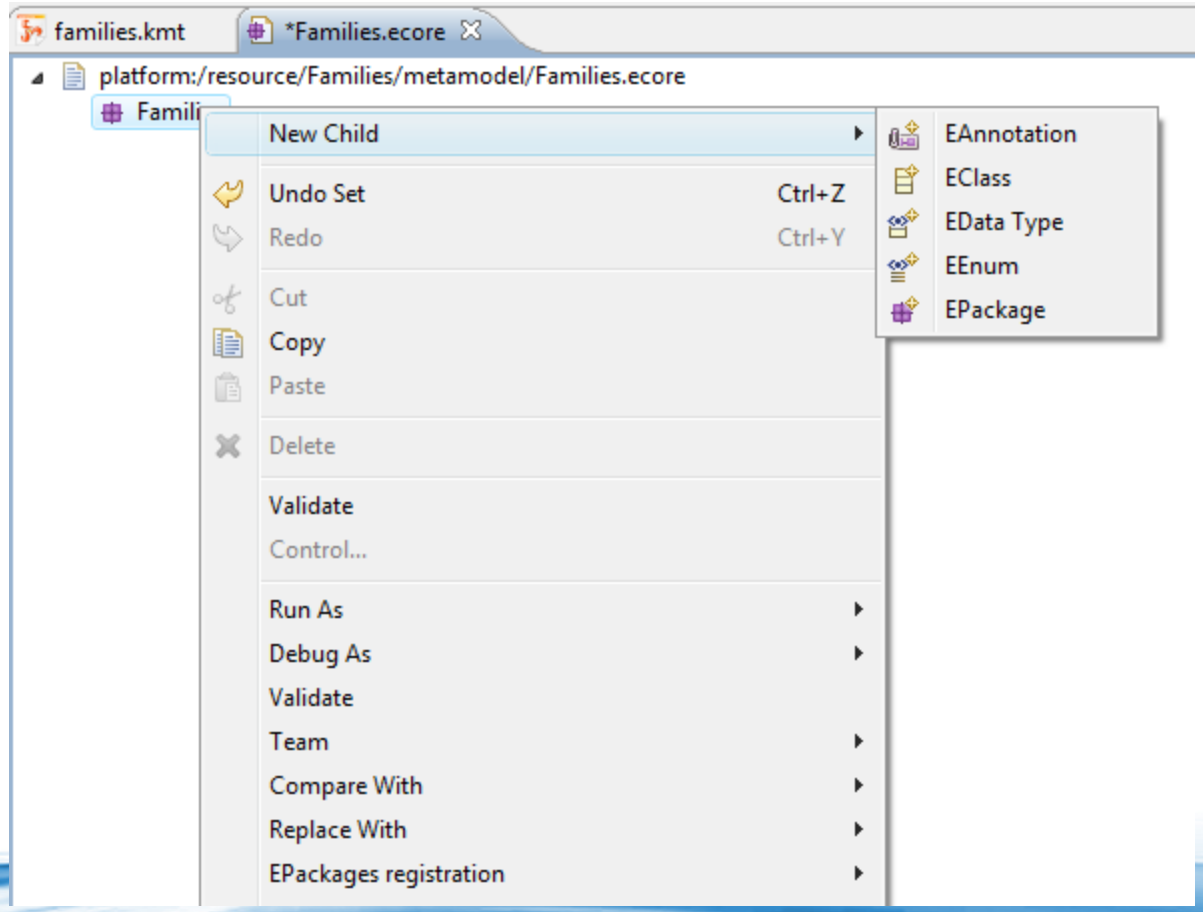
families.kmt | *Families.ecore

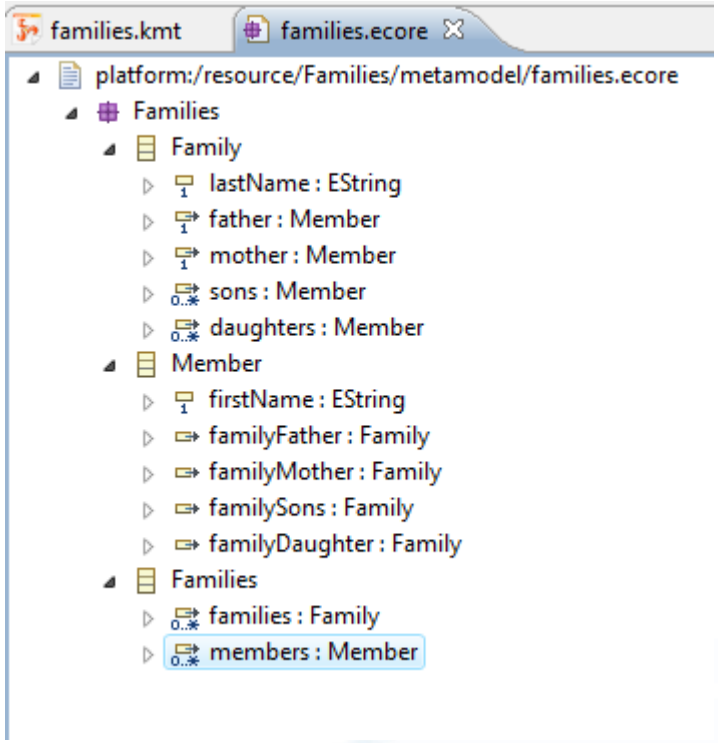
platform:/resource/Families/metamodel/Families.ecore

Families

Problems | Console | Error Log | Properties

Property	Value
Name	Families
Ns Prefix	Families
Ns URI	http://Families.ecore


families.kmt
*Families.ecore
platform:/resource/Families/metamodel/Families.ecore
Famili
New Child
Undo Set Ctrl+Z
Redo Ctrl+Y
Cut
Copy
Paste
Delete
Validate
Control...
Run As
Debug As
Validate
Team
Compare With
Replace With
EPackages registration
EAnnotation
EClass
EData Type
EEnum
EPackage


families.kmt
families.ecore
platform:/resource/Families/metamodel/families.ecore
Families
Family
lastName : EString
father : Member
mother : Member
sons : Member
daughters : Member
Member
firstName : EString
familyFather : Family
familyMother : Family
familySons : Family
familyDaughter : Family
Families
families : Family
members : Member

- Families
  - Family
    - lastName : EString
    - father : Member
    - mother : Member
    - sons : Member
    - daughters : Member
  - Member
    - firstName : EString
    - familyFather : Family
    - familyMother : Family
    - familySons : Family
    - familyDaughter : Family
  - Families
    - families : Family
    - members : Member

Problems | Console | Error Log | Properties

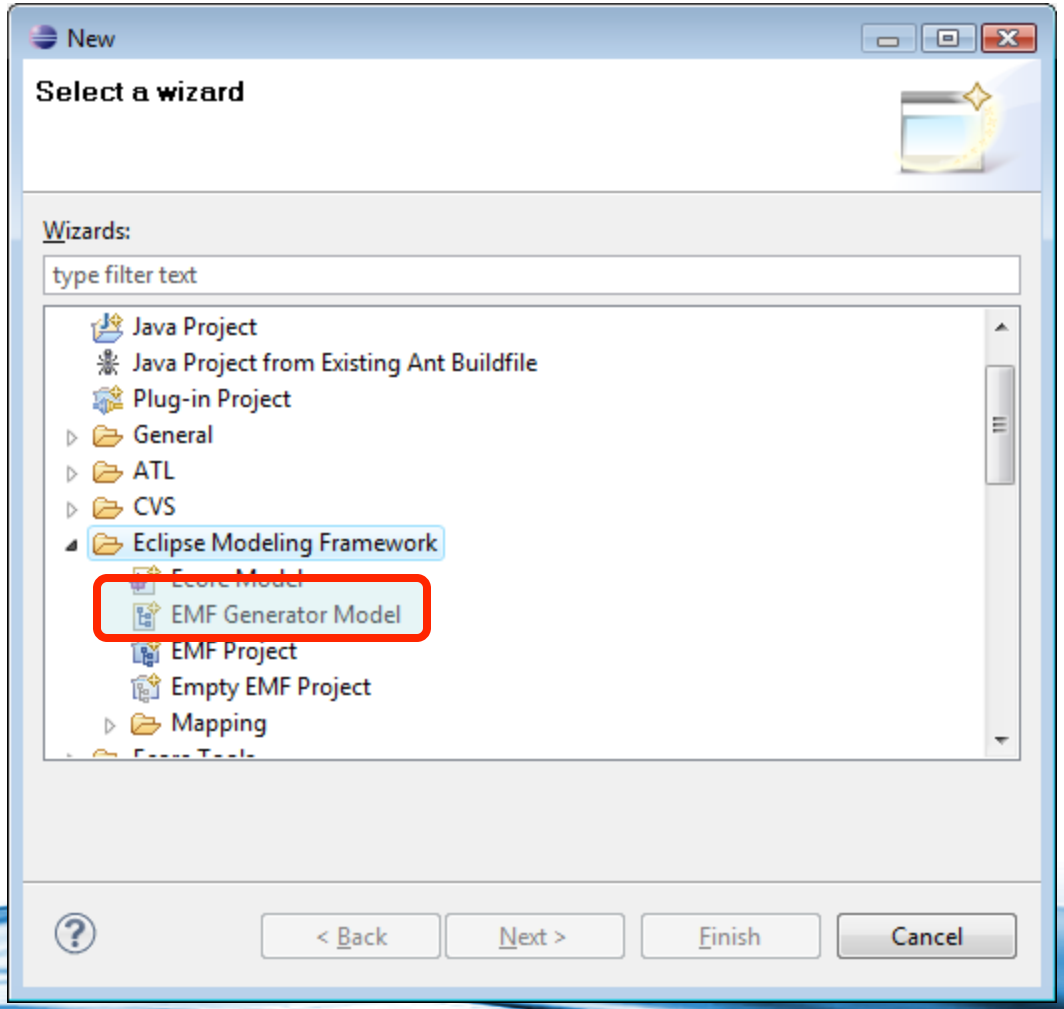
Property	Value
Derived	false
EKeys	
EOpposite	familyFather : Family
EType	
Lower Bound	

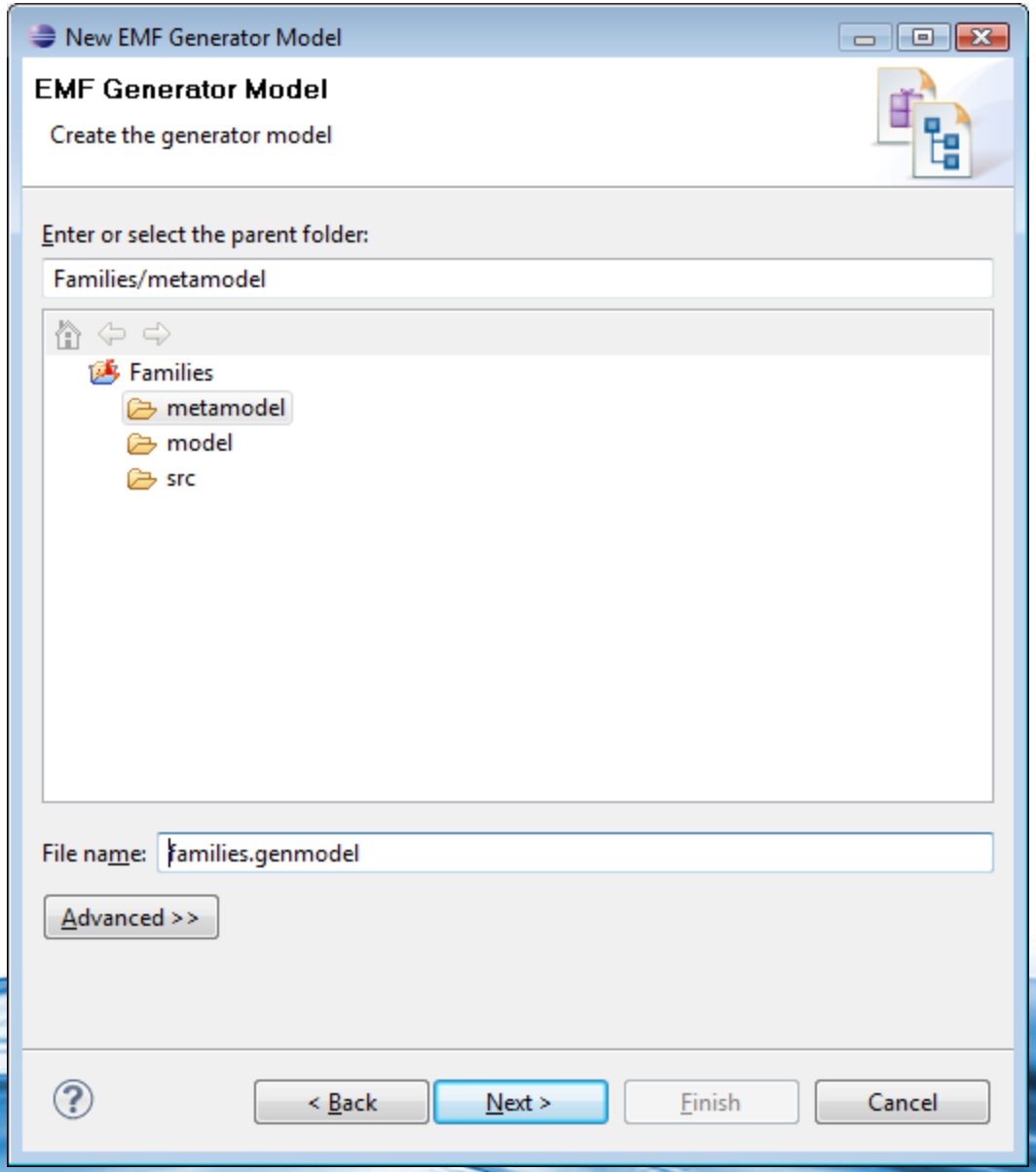
familyDaughter : Family
familyFather : Family
familyMother : Family
familySons : Family

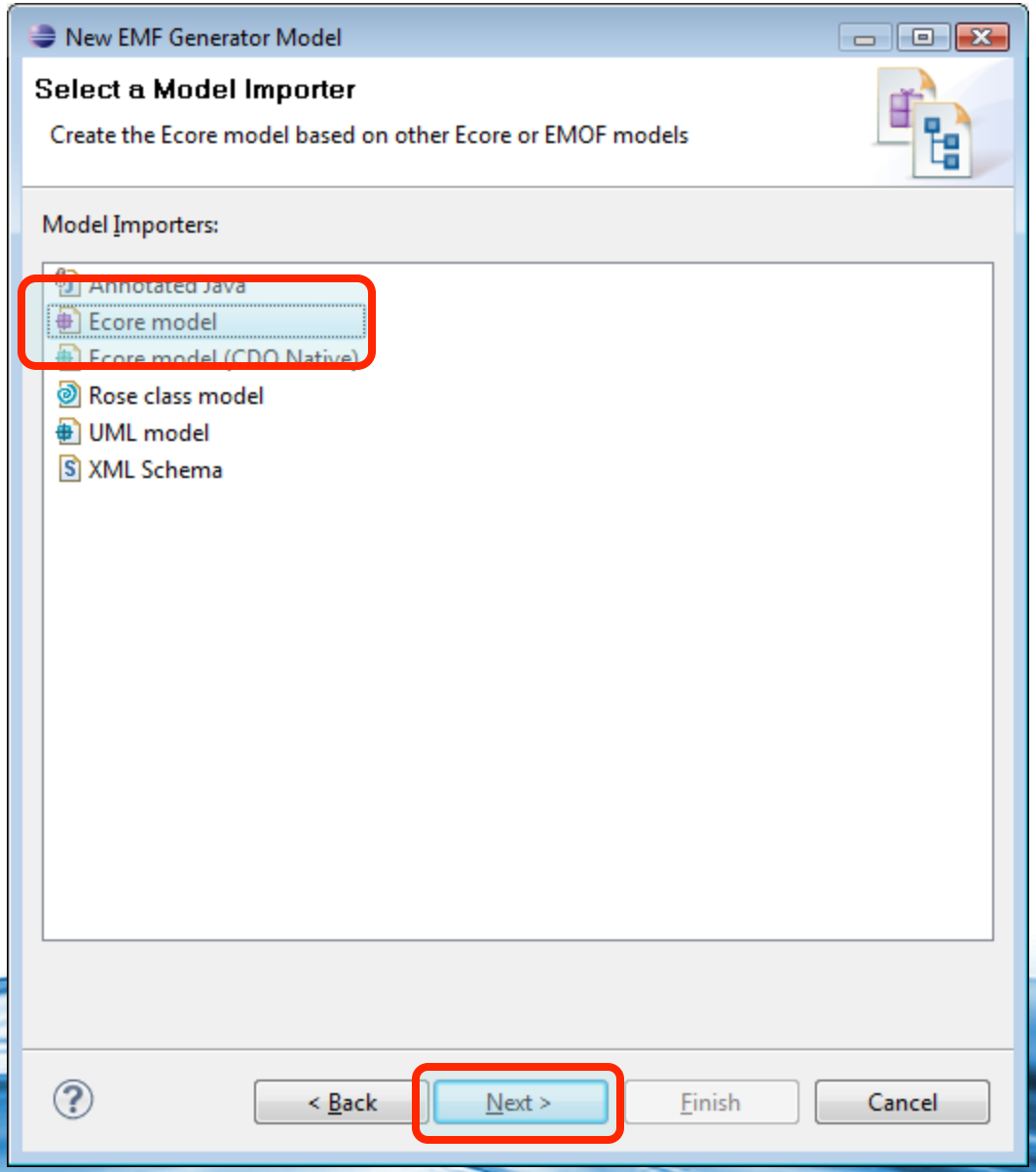
al opposite of this reference

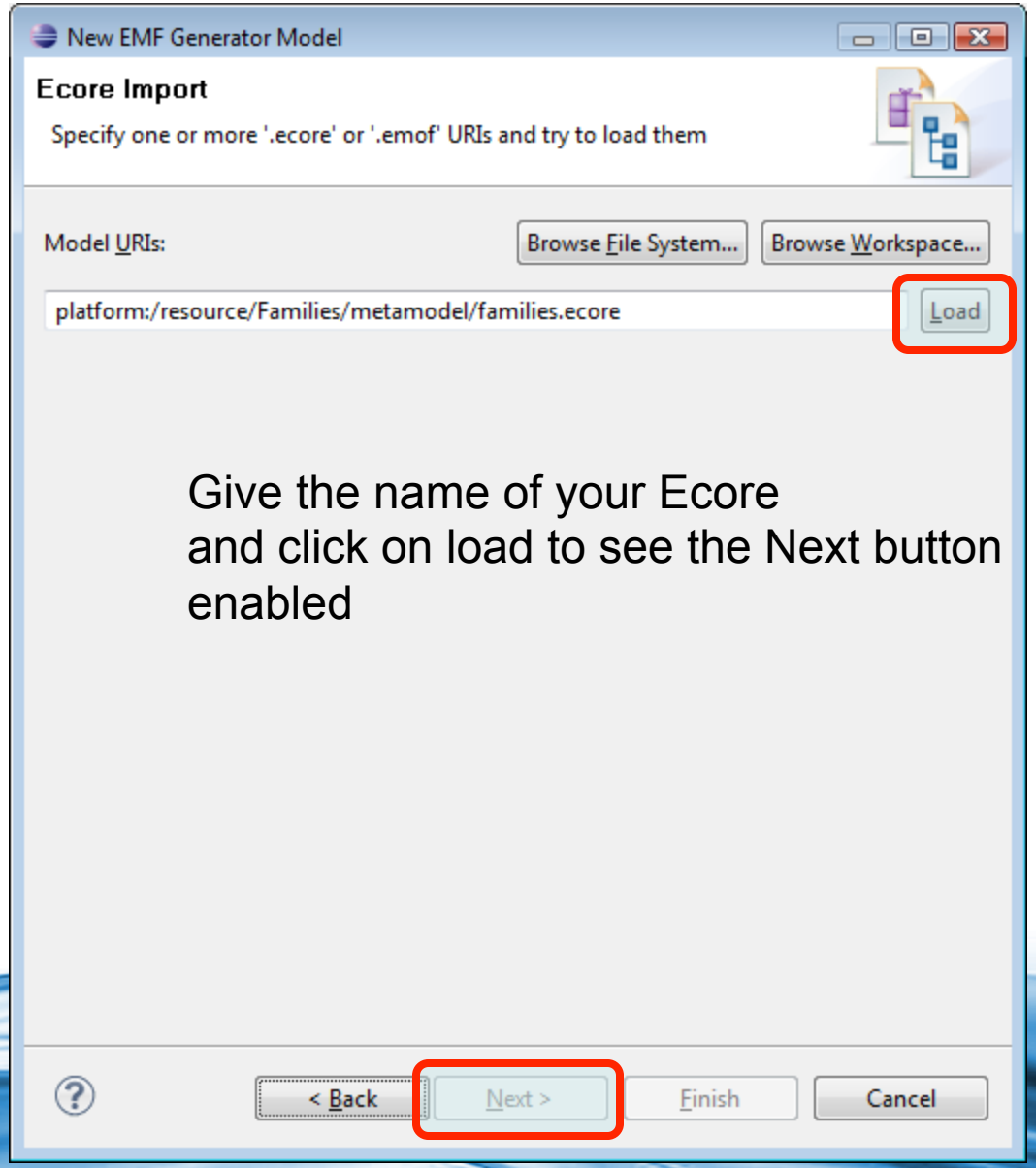
Generate the editor for the model

- Right click on families.ecore in Project Explorer>New>EMF Generator Model

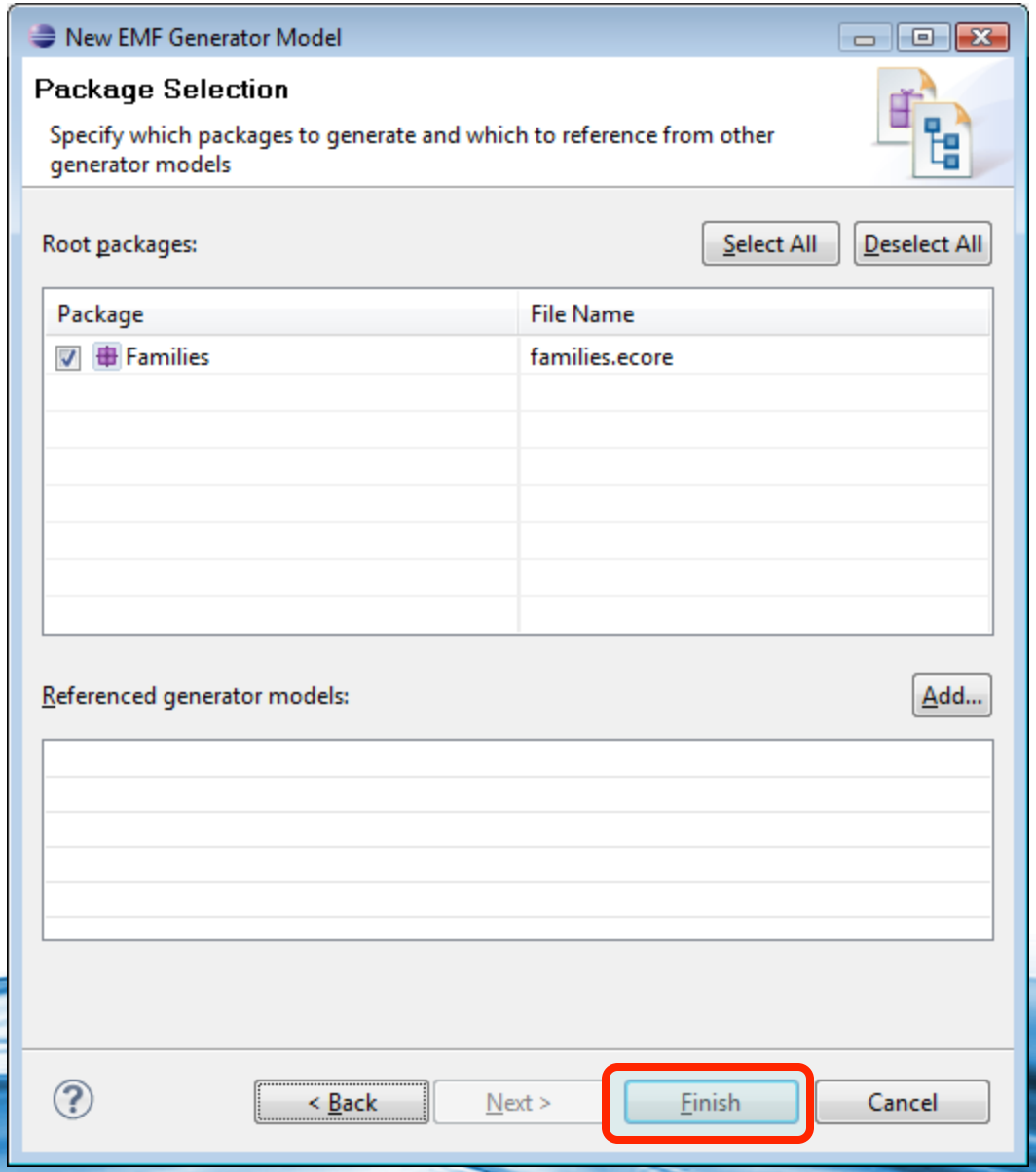

New
Select a wizard
Wizards:
type filter text
Java Project
Java Project from Existing Ant Buildfile
Plug-in Project
General
ATL
CVS
Eclipse Modeling Framework
EMF Generator Model
EMF Project
Empty EMF Project
Mapping
< Back
Next >
Finish
Cancel

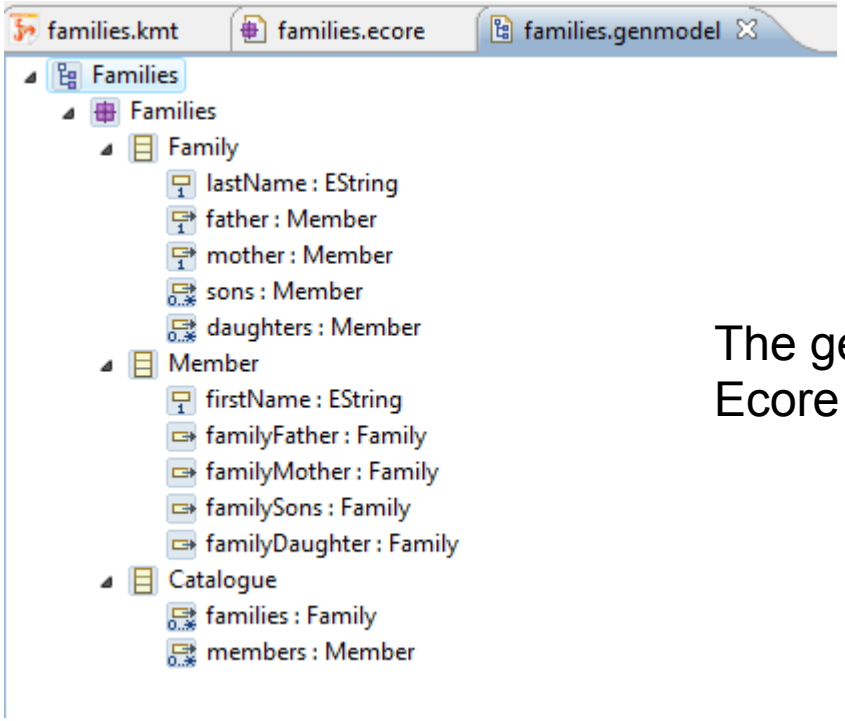

New EMF Generator Model
EMF Generator Model
Create the generator model
Enter or select the parent folder:
Families/metamodel
Families
metamodel
model
src
File name:
families.genmodel
Advanced >>
< Back
Next >
Finish
Cancel


New EMF Generator Model
Select a Model Importer
Create the Ecore model based on other Ecore or EMOF models
Model Importers:
Annotated Java
Ecore model
Ecore model (CDO Native)
Rose class model
UML model
XML Schema
< Back
Next >
Finish
Cancel



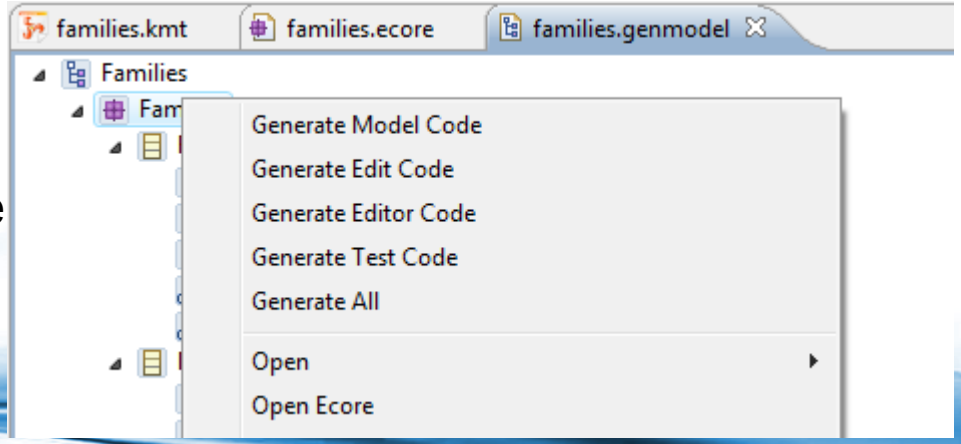
Give the name of your Ecore
and click on load to see the Next button
enabled

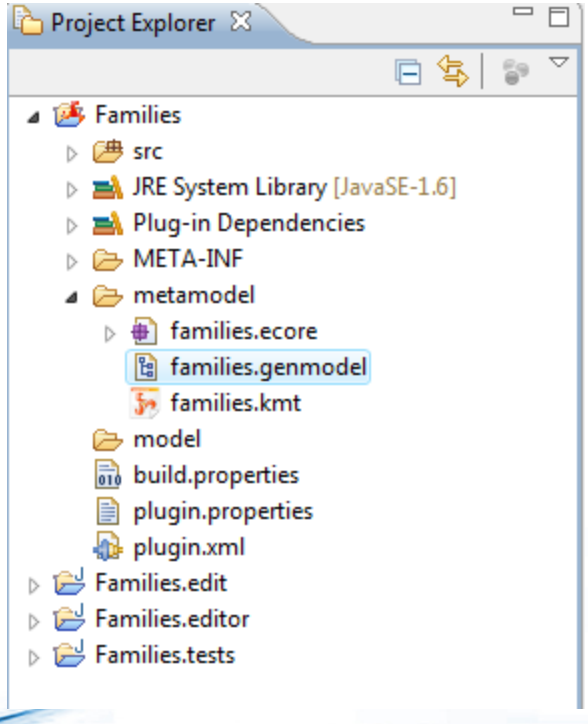

New EMF Generator Model
Package Selection
Specify which packages to generate and which to reference from other generator models
Root packages:
Select All
Deselect All
Package
File Name
Families
families.ecore
Referenced generator models:
Add...
< Back
Next >
Finish
Cancel



The generator model is equivalent to the Ecore file but is prepared to build an editor

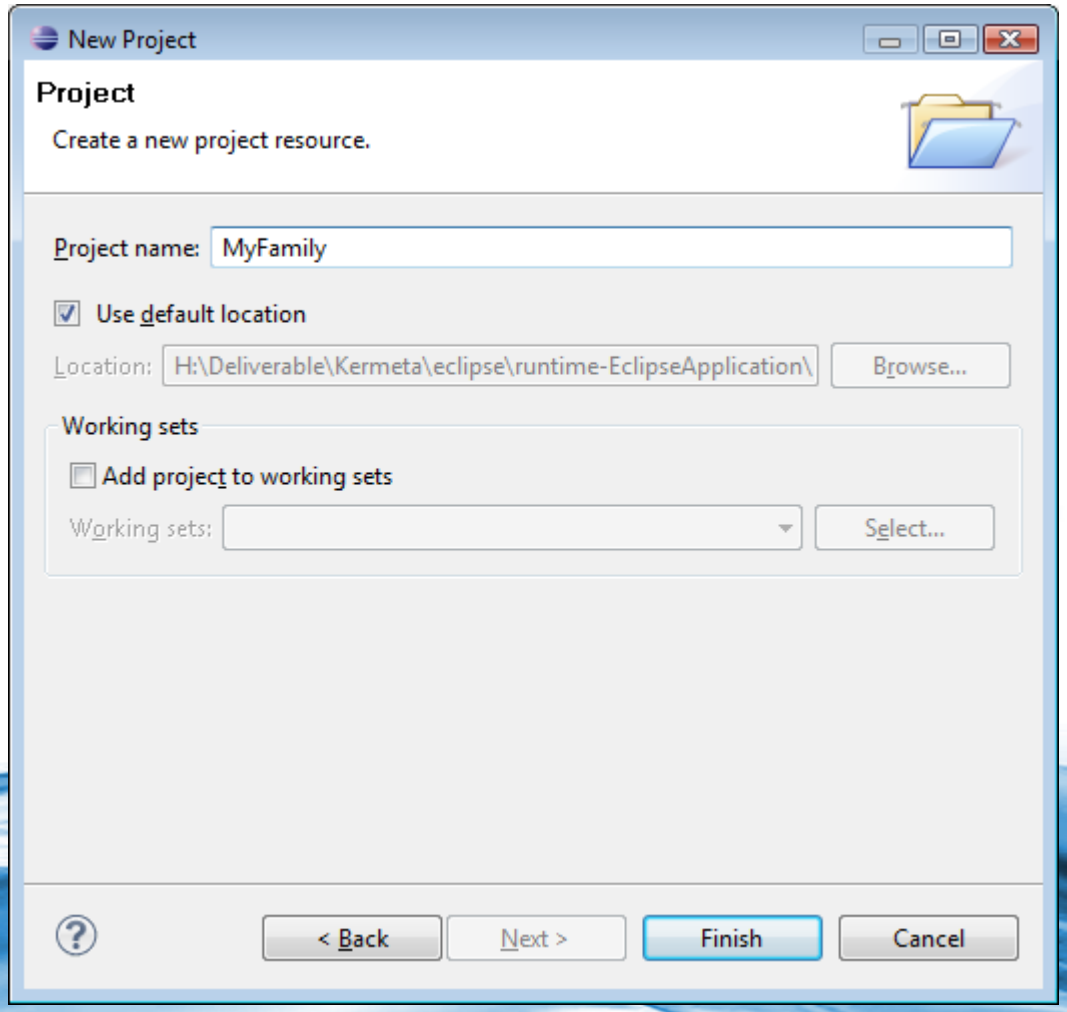
Right-click on the package and click on Generate all



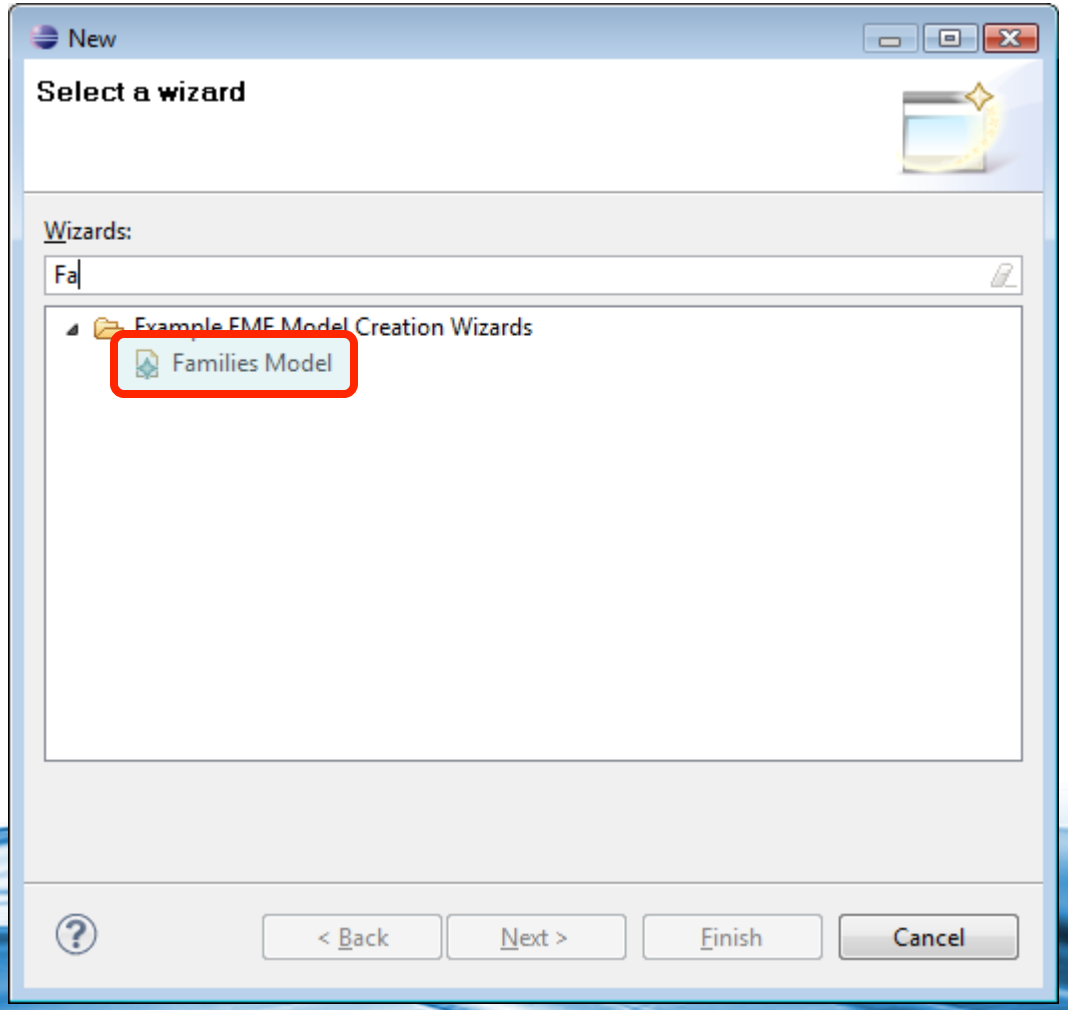


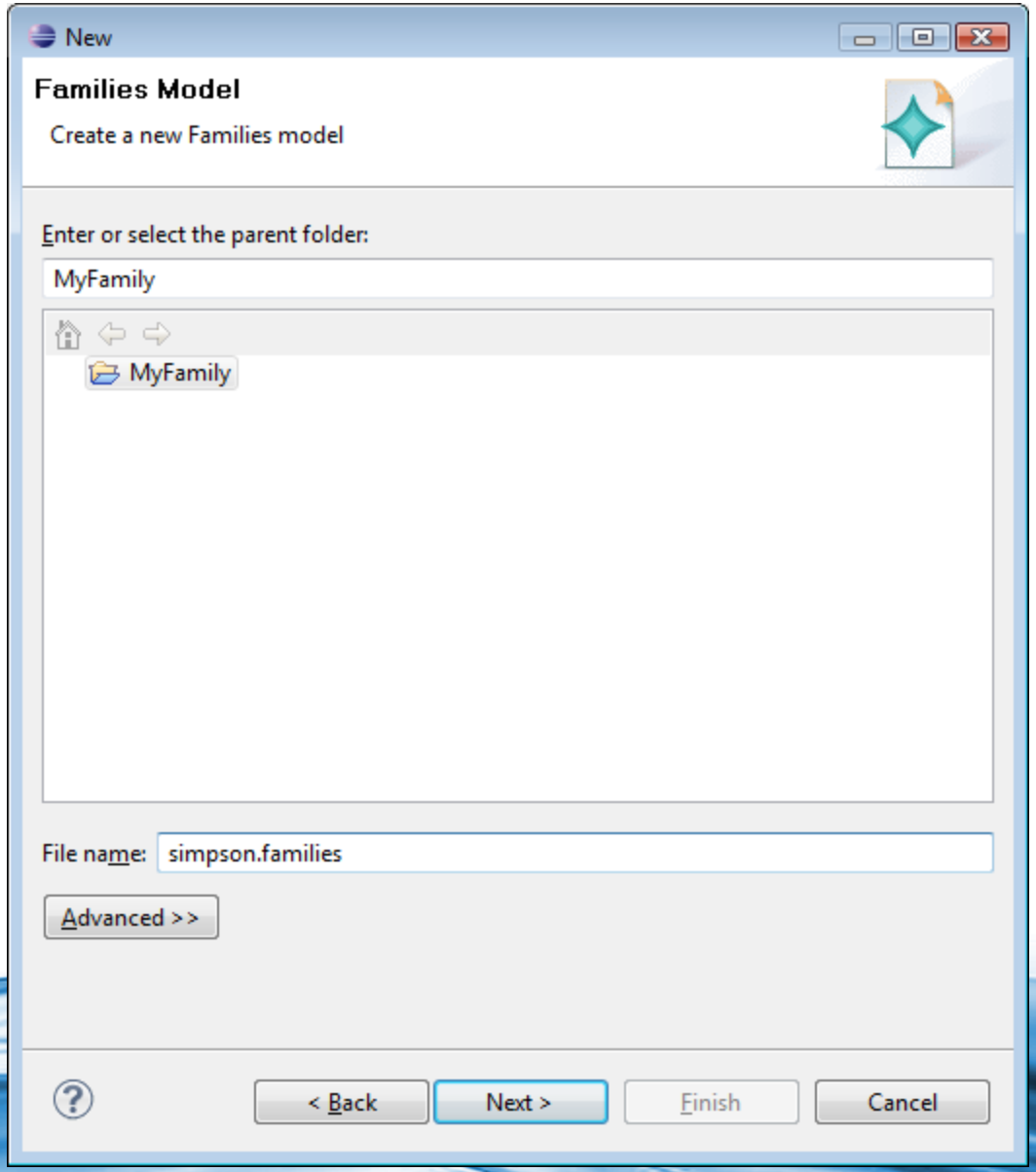
After generation, you can find some new projects in your workspace. A plugin (editor + wizard) is created for your metamodel

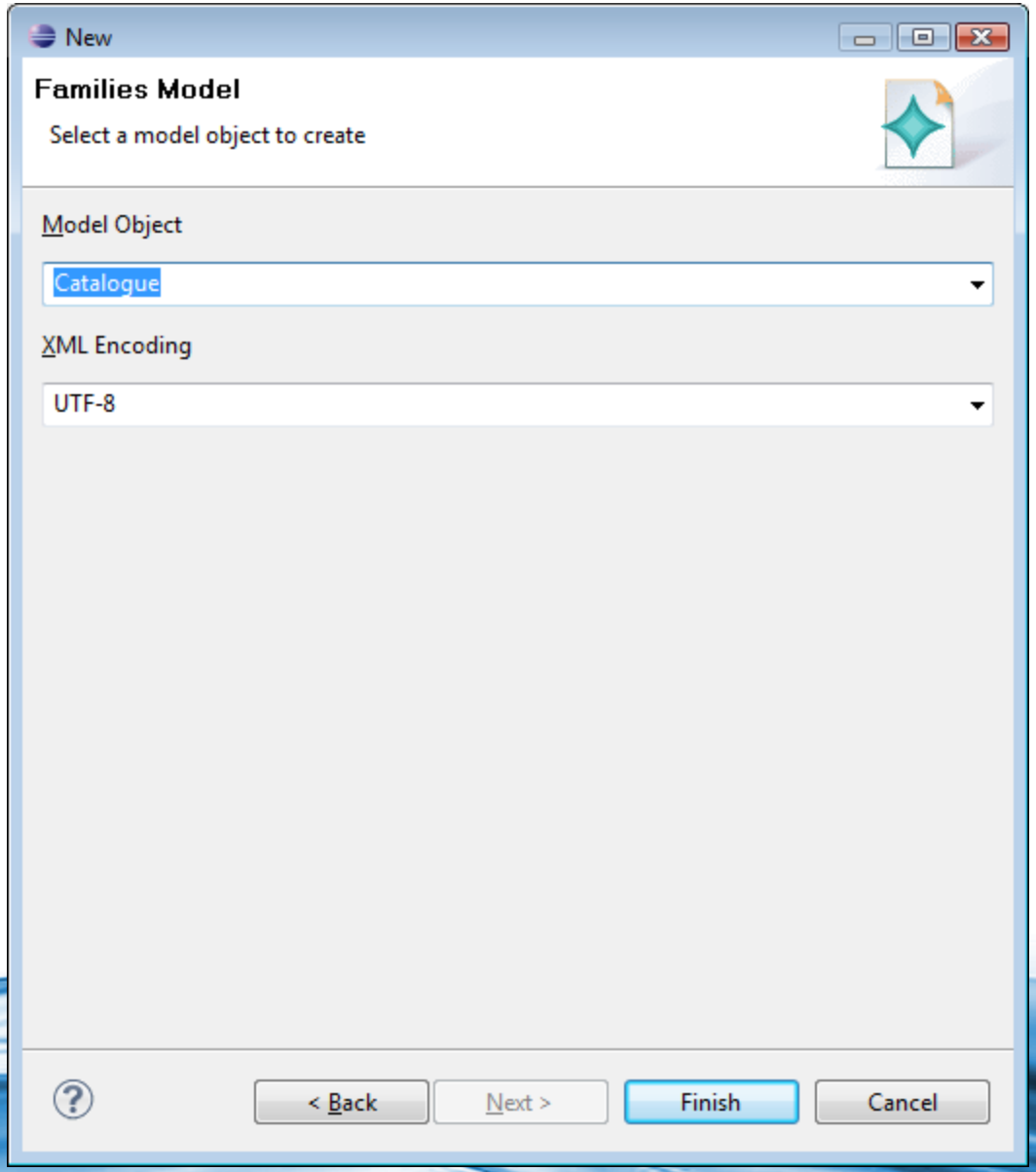
Run a new Eclipse from the one you are, and File>New>Project

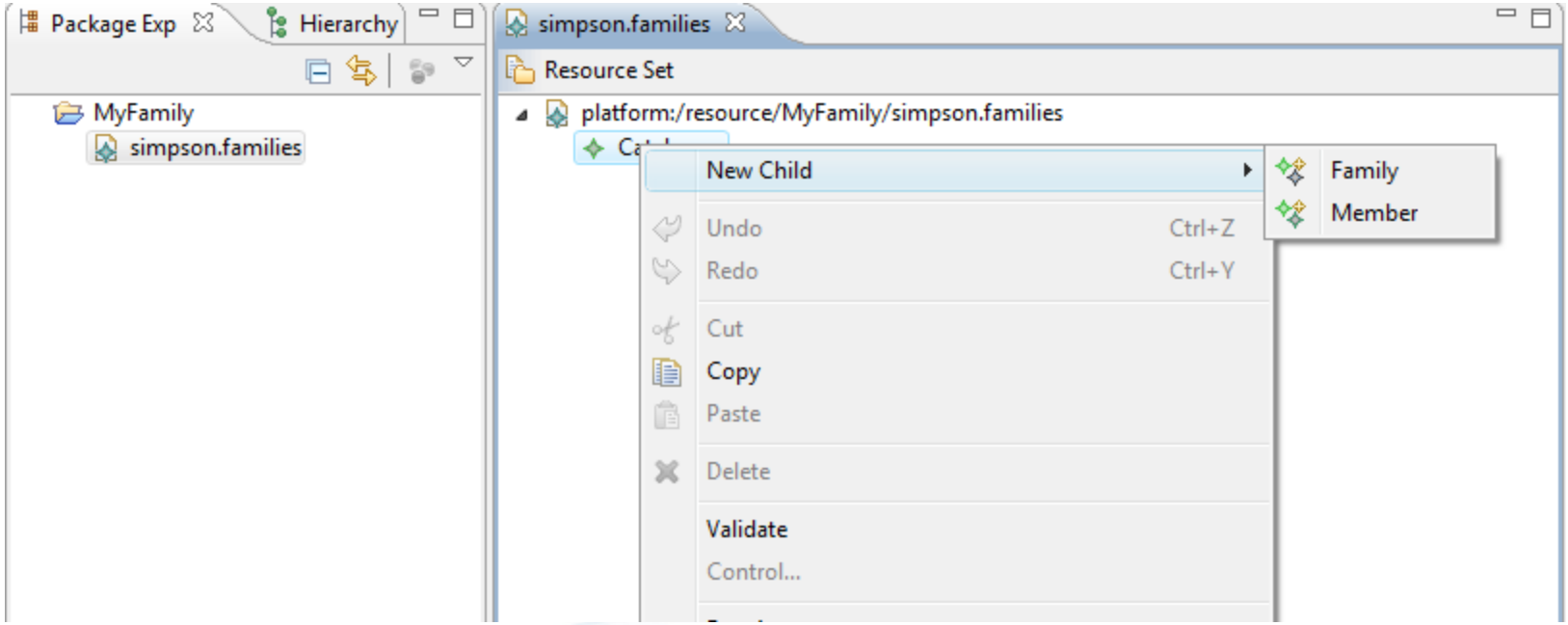


On the project, right click New>Other




New
Families Model
Create a new Families model
Enter or select the parent folder:
MyFamily
MyFamily
File name:
simpson.families
Advanced >>
< Back
Next >
Finish
Cancel


New
Families Model
Select a model object to create
Model Object
Catalogue
XML Encoding
UTF-8
< Back
Next >
Finish
Cancel


Package Exp
Hierarchy
MyFamily
simpson.families
simpson.families
Resource Set
platform:/resource/MyFamily/simpson.families
New Child
Undo Ctrl+Z
Redo Ctrl+Y
Cut
Copy
Paste
Delete
Validate
Control...
Family
Member

Java - MyFamily/simpson.families - Eclipse Platform

File Edit Navigate Search Project Run Families Editor Window Help

Package Exp | Hierarchy

- MyFamily
  - simpson.families

simpson.families

Resource Set

- platform:/resource/MyFamily/simpson.families
  - Catalogue
    - Family Simpson

Selection | Parent | List | Tree | Table | Tree with Columns

Task List

Find | All | Activat...

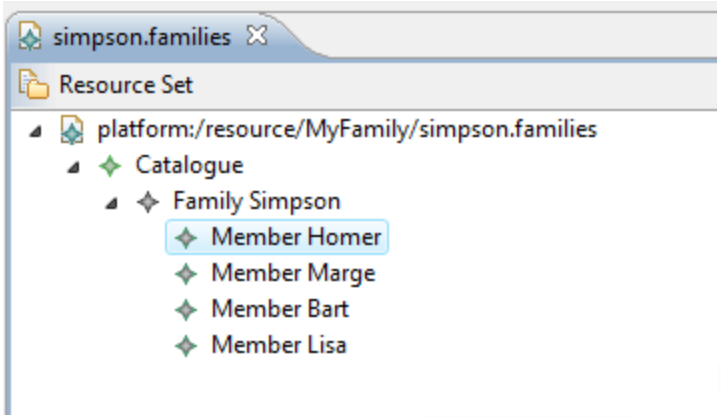
Uncategorized

Outline

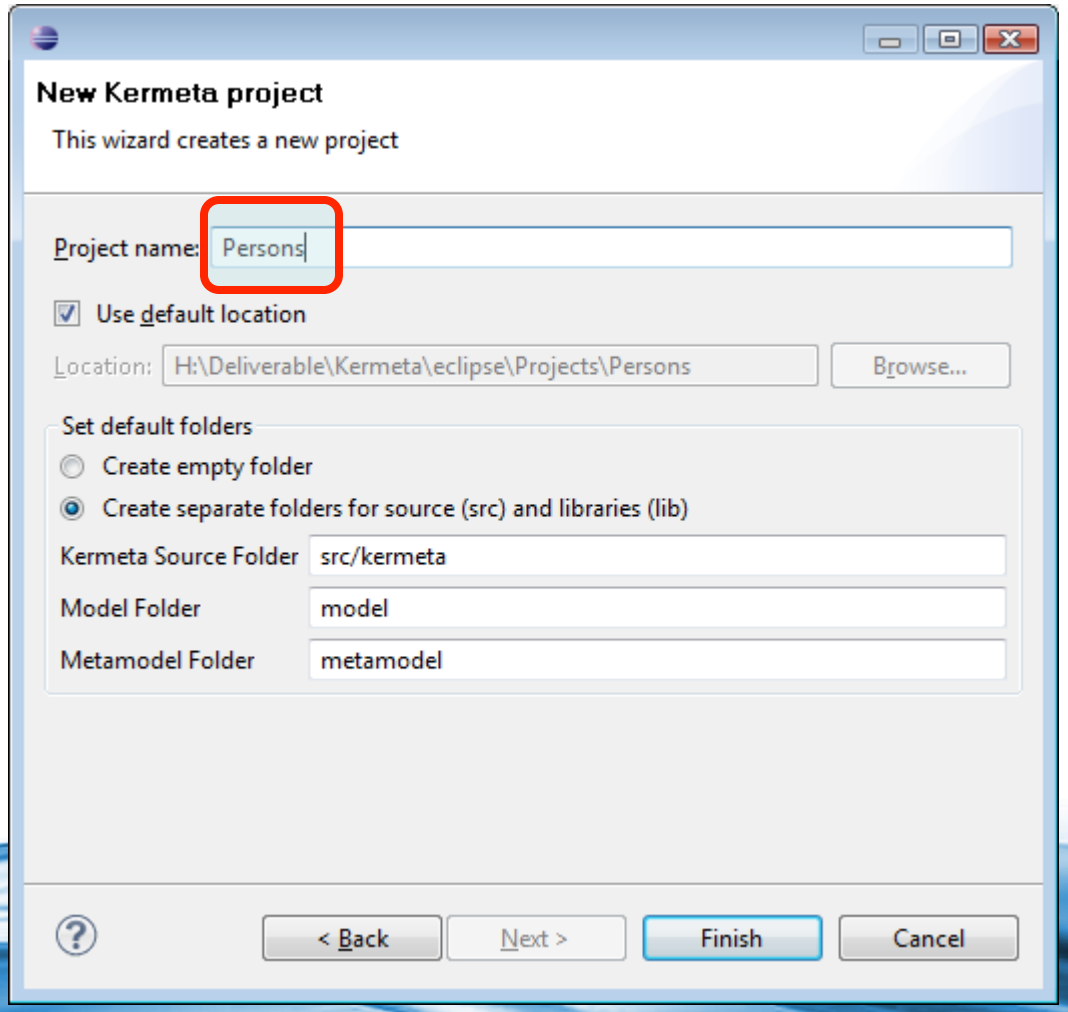
platform:/resource/MyFan

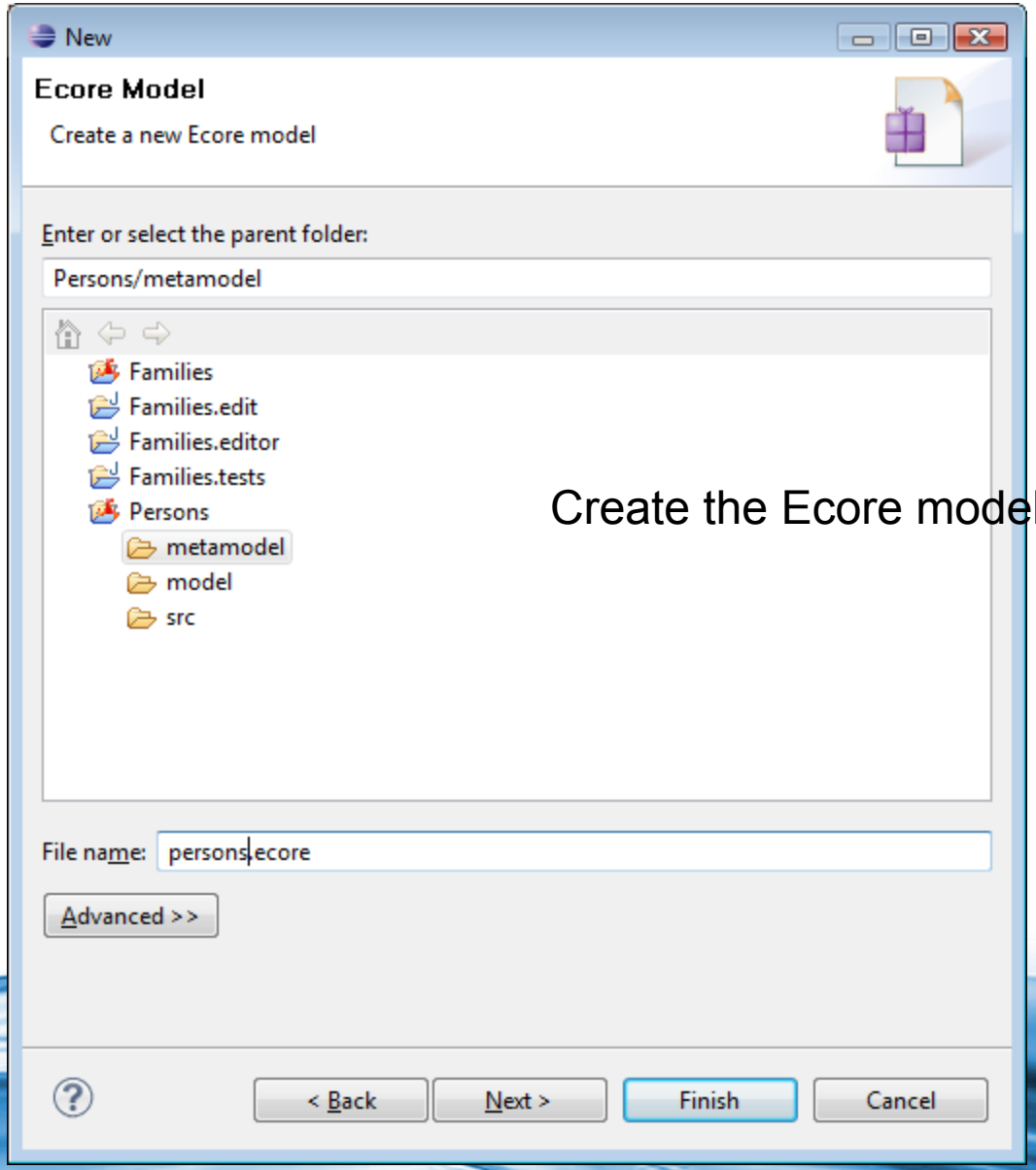
Problems | Javadoc | Declaration | Properties

Property	Value
Last Name	Simpson

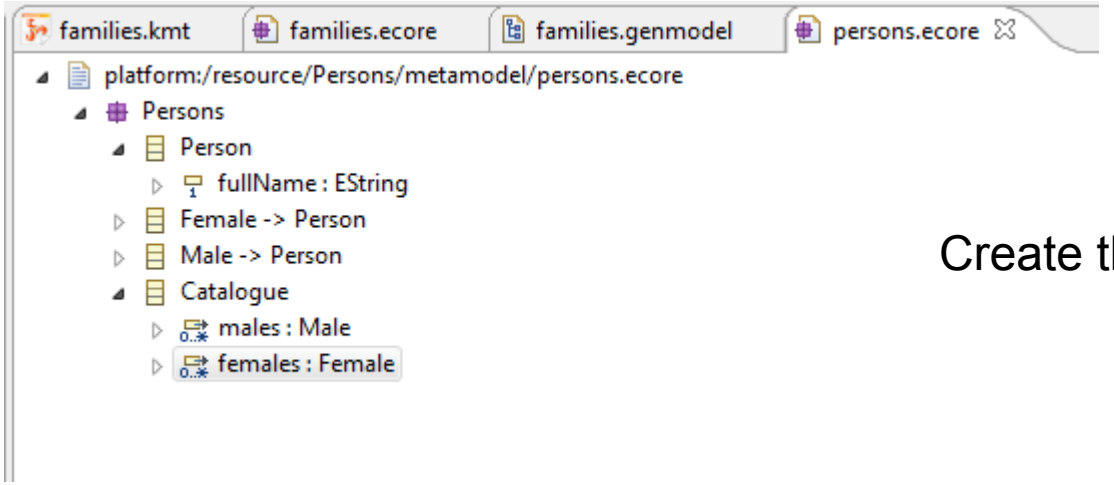

simpson.families
Resource Set
platform:/resource/MyFamily/simpson.families
Catalogue
Family Simpson
Member Homer
Member Marge
Member Bart
Member Lisa

Create the target metamodel

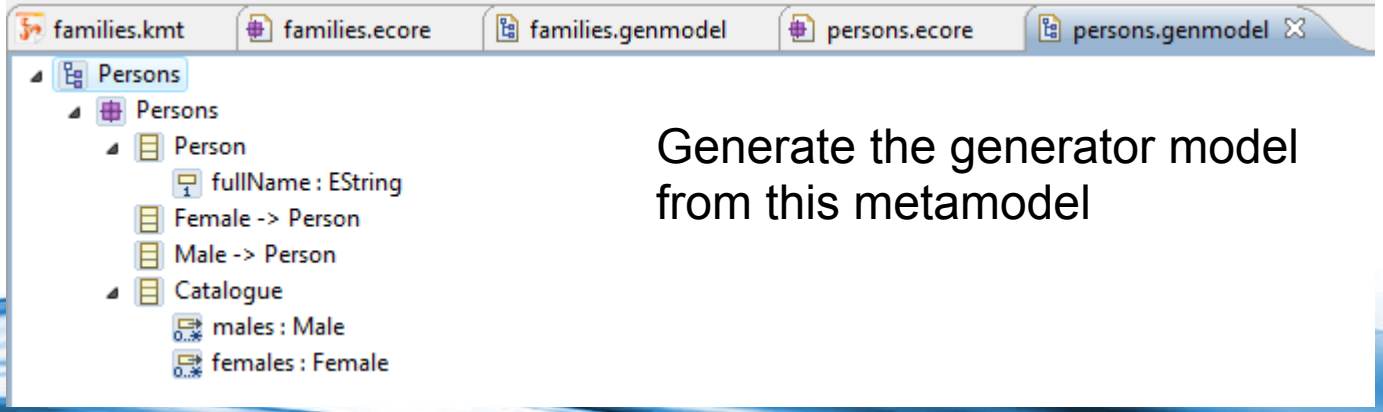




Create the Ecore model persons



Create the metamodel as follows

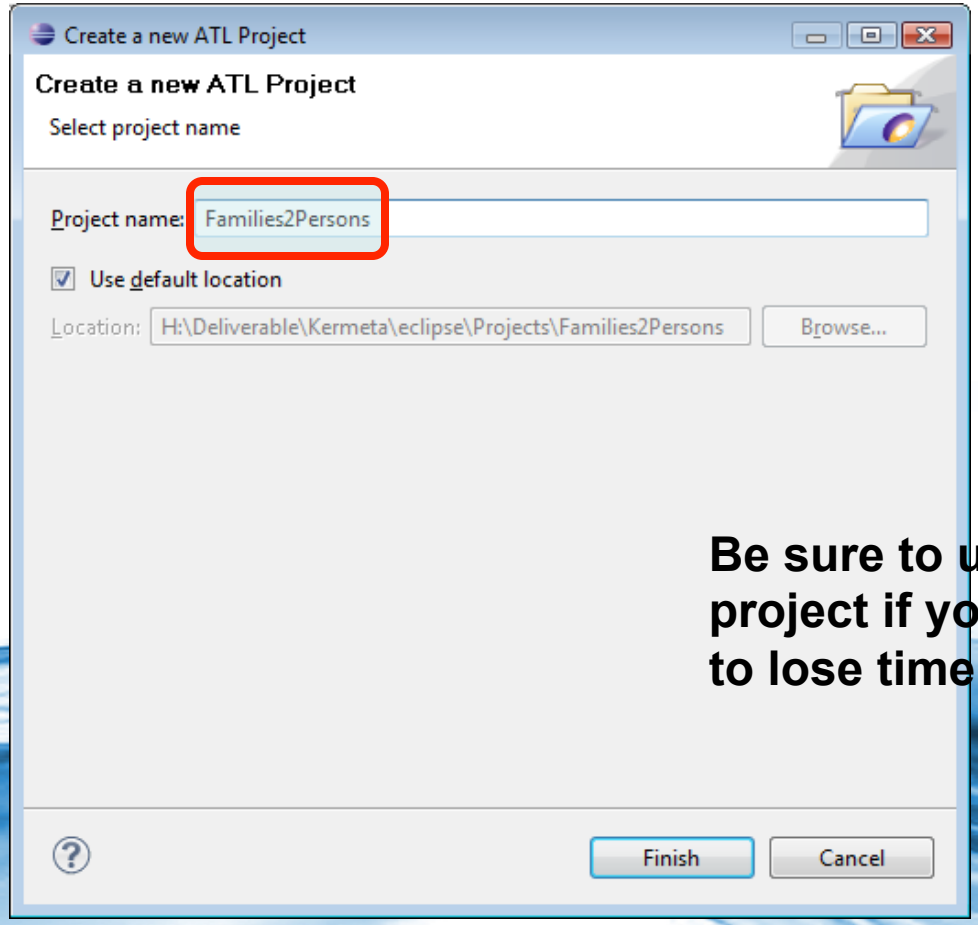


Generate the generator model from this metamodel

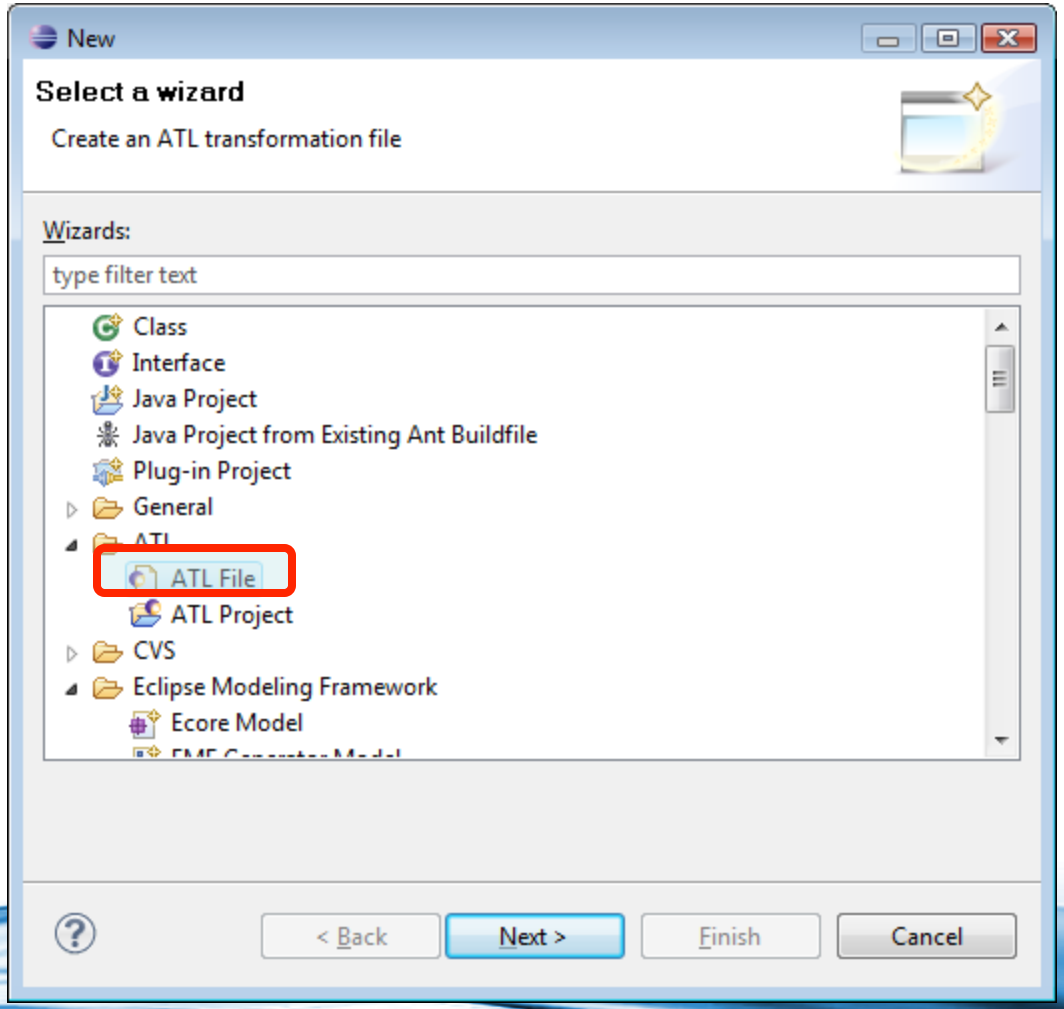
Then generate all

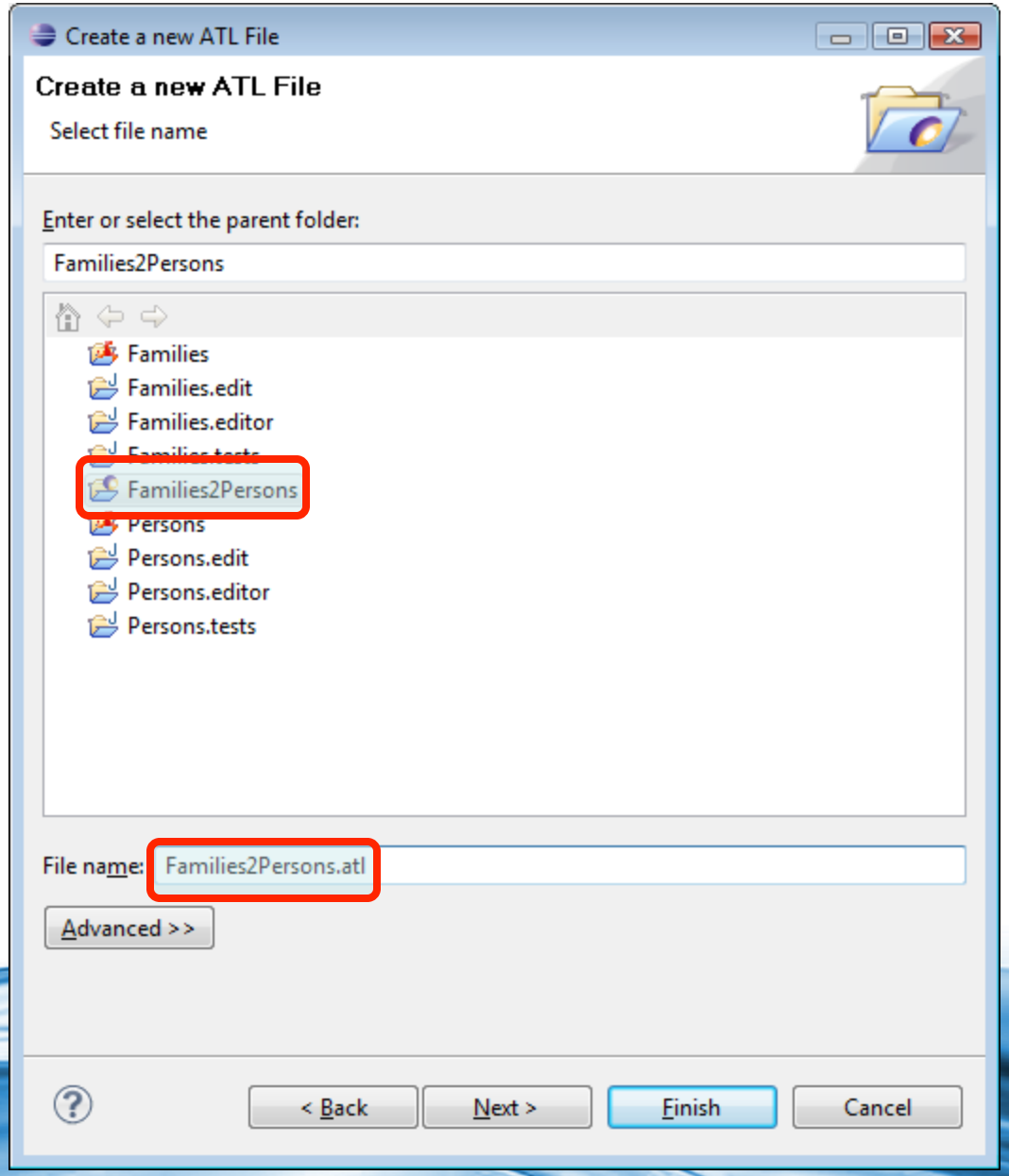
Create the transformation rules

- File>New>Other>ATL>ATL Project



Be sure to use an ATL project if you don't want to lose time finding the asm file


New
Select a wizard
Create an ATL transformation file
Wizards:
type filter text
Class
Interface
Java Project
Java Project from Existing Ant Buildfile
Plug-in Project
General
ATL
ATL File
ATL Project
CVS
Eclipse Modeling Framework
Ecore Model
< Back
Next >
Finish
Cancel


Create a new ATL File
Create a new ATL File
Select file name
Enter or select the parent folder:
Families2Persons
Families
Families.edit
Families.editor
Families2Persons
Persons
Persons.edit
Persons.editor
Persons.tests
File name:
Families2Persons.atl
Advanced >>
< Back
Next >
Finish
Cancel

Create a new ATL File

ATL Header parameters

There must be at least one input model

Module Name: Families2Persons

File Type: module

Input Models: Add... Remove

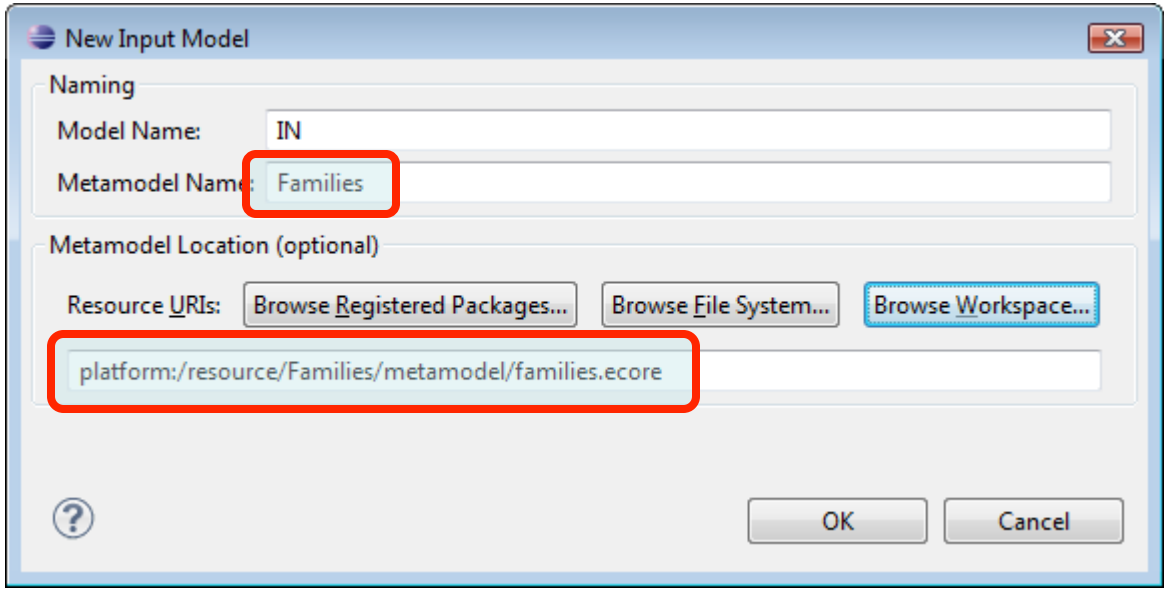
Output Models: Add... Remove

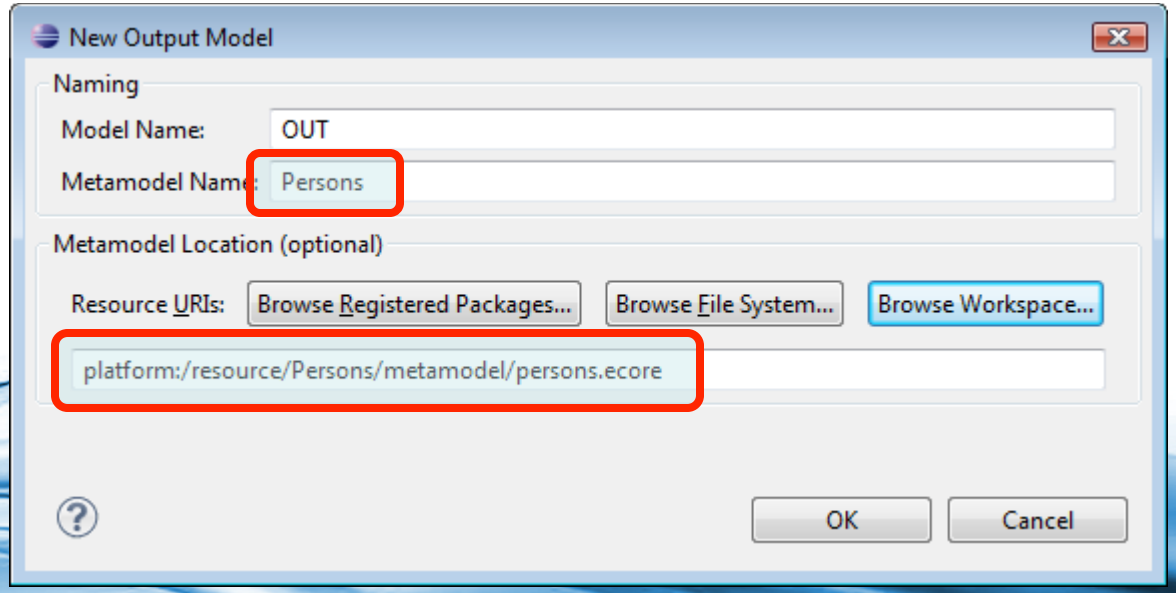
Libraries: Add... Remove

Do you want to create the launch configuration?

☑ Generate configuration

< Back | Next > | Finish | Cancel


New Input Model
Naming
Model Name:
IN
Metamodel Name:
Families
Metamodel Location (optional)
Resource URIs:
Browse Registered Packages...
Browse File System...
Browse Workspace...
platform:/resource/Families/metamodel/families.ecore
OK
Cancel


New Output Model
Naming
Model Name:
OUT
Metamodel Name:
Persons
Metamodel Location (optional)
Resource URIs:
Browse Registered Packages...
Browse File System...
Browse Workspace...
platform:/resource/Persons/metamodel/persons.ecore
OK
Cancel

Create a new ATL File

ATL Header parameters

Create ATL Header parameters

Module Name: Families2Persons

File Type: module

Input Models: IN : Families (platform:/resource/Families/metamodel/fa

Add...

Remove

Output Models: OUT : Persons (platform:/resource/Persons/metamodel/

Add...

Remove

Libraries:

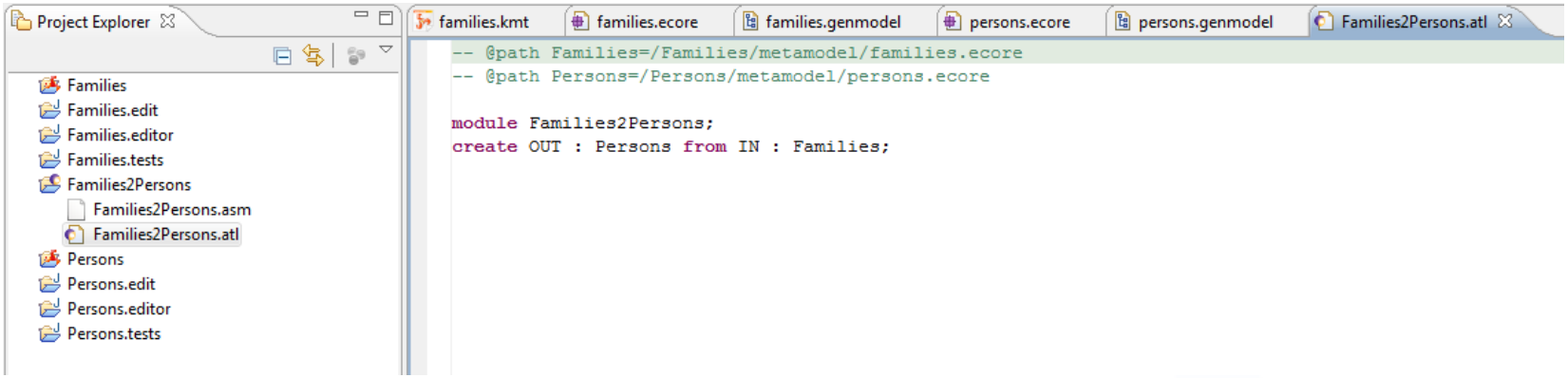
Add...

Remove

Do you want to create the launch configuration?

☑ Generate configuration

< Back | Next > | Finish | Cancel


Project Explorer
Families
Families.edit
Families.editor
Families.tests
Families2Persons
Families2Persons.asm
Families2Persons.atl
Persons
Persons.edit
Persons.editor
Persons.tests
families.kmt
families.ecore
families.genmodel
persons.ecore
persons.genmodel
Families2Persons.atl
-- @path Families=/Families/metamodel/families.ecore
-- @path Persons=/Persons/metamodel/persons.ecore
module Families2Persons;
create OUT : Persons from IN : Families;

Families2Persons.atl

```
-- @path Families=/Families2Persons/Families.ecore
-- @path Persons=/Families2Persons/Persons.ecore

module Families2Persons;
create OUT : Persons from IN : Families;

helper context Families!Member def: familyName : String =
	if not self.familyFather.oclIsUndefined() then
		self.familyFather.lastName
	else
		if not self.familyMother.oclIsUndefined() then
			self.familyMother.lastName
		else
			if not self.familySon.oclIsUndefined() then
				self.familySon.lastName
			else
				self.familyDaughter.lastName
			endif
		endif
	endif;
```

Families2Persons.atl

```
helper context Families!Member def: isFemale() : Boolean =
	if not self.familyMother.oclIsUndefined() then
		true
	else
		if not self.familyDaughter.oclIsUndefined() then
			true
		else
			false
		endif
	endif;

rule Member2Male {
	from
		s : Families!Member (not s.isFemale())
	to
		t : Persons!Male (
			fullName <- s.firstName + ' ' + s.familyName
		)
}
```

Families2Persons.atl

```
rule Member2Female {
	from
		s : Families!Member (s.isFemale())
	to
		t : Persons!Female (
			fullName <- s.firstName + ' ' + s.familyName
		)
}
```

Last step!

Project Explorer

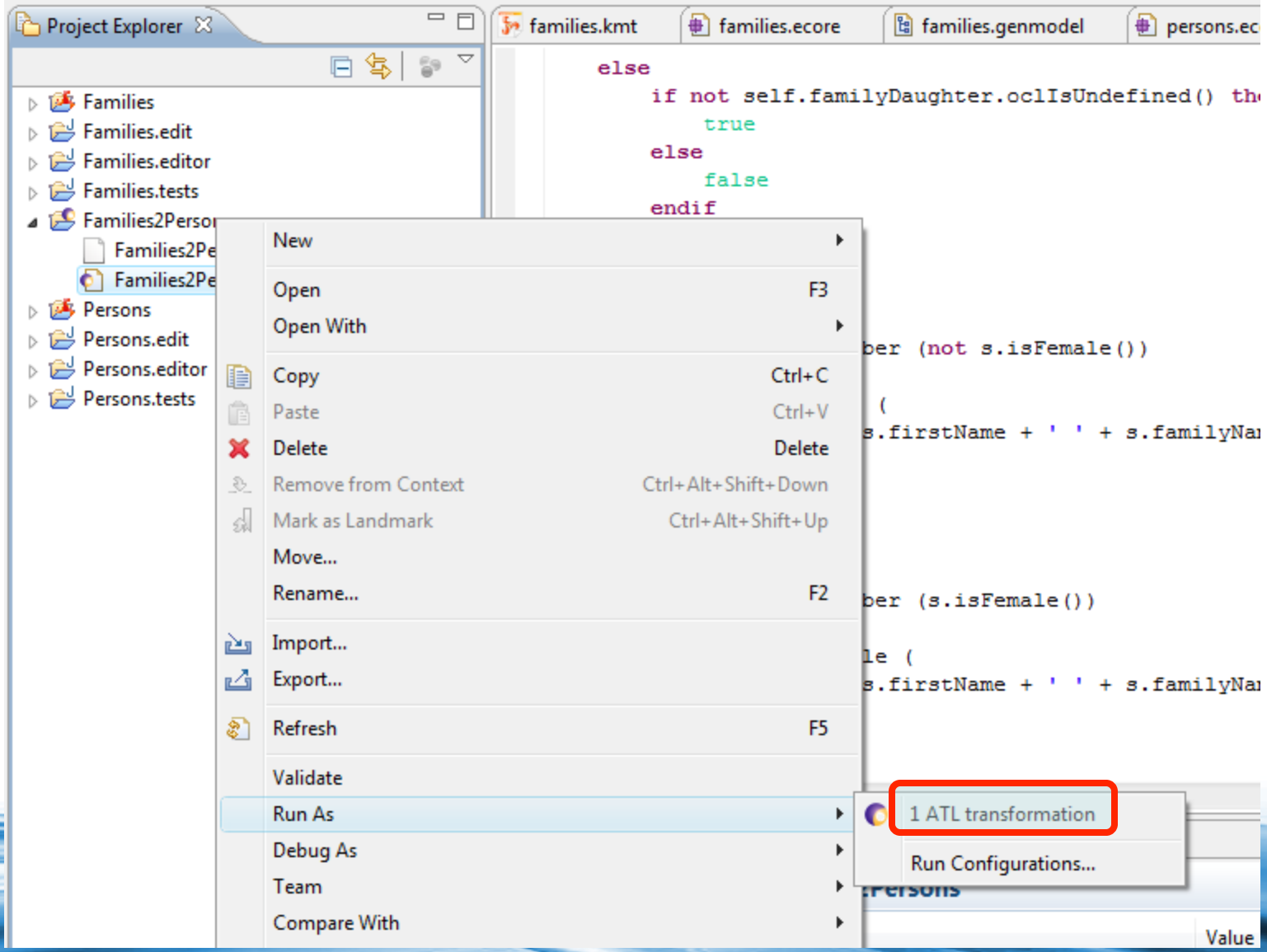
- Families
- Families.edit
- Families.editor
- Families.tests
- Families2Persons
  - Families2Persons.asm
  - Families2Persons.atl
- Persons
- Persons.edit
- Persons.editor
- Persons.tests

families.kmt | families.ecore | families.genmodel | persons.ecore | persons.genmodel | Families2Persons.atl

```
    else
        if not self.familyDaughter.oclIsUndefined() then
            true
        else
            false
        endif
    endif;

rule Member2Male {
    from
        s : Families!Member (not s.isFemale())
    to
        t : Persons!Male (
            fullName <- s.firstName + ' ' + s.familyName
        )
}

rule Member2Female {
    from
        s : Families!Member (s.isFemale())
    to
        t : Persons!Female (
            fullName <- s.firstName + ' ' + s.familyName
        )
}
```

Families2Persons.asm is the assembler version of the ATL file for use on the ATL virtual machine

Right-click on the ATL file



Edit Configuration

Edit configuration and launch.

Please, give a path for IN

Name: Families2Persons

ATL Configuration | Advanced | Common

ATL Module

/Families2Persons/Families2Persons.atl — Workspace...

Metamodels

Families: /Families/metamodel/families.ecore

Is metametamodel — Workspace... File system... EMF Registry...

Persons: /Persons/metamodel/persons.ecore

Is metametamodel — Workspace... File system... EMF Registry...

Source Models

IN:

conforms to Families — Workspace... File system...

Target Models

OUT:

conforms to Persons — Workspace... File system...

Apply Revert

Run Close

Edit Configuration

Edit configuration and launch.

Name: Families2Persons

ATL Configuration | Advanced | Common

ATL Module

/Families2Persons/Families2Persons.atl — Workspace...

Metamodels

Families: /Families/metamodel/families.ecore

Is metametamodel — Workspace... File system... EMF Registry...

Persons: /Persons/metamodel/persons.ecore

Is metametamodel — Workspace... File system... EMF Registry...

Source Models

IN: /Families/model/simpsons.families

conforms to Families — Workspace... File system...

Target Models

OUT: /Persons/model/simpsons.persons

conforms to Persons — Workspace... File system...

Apply | Revert

Run | Close

If you are lucky, you should see this

Project Explorer

- Families
  - src
  - JRE System Library [JavaSE-1.6]
  - Plug-in Dependencies
  - META-INF
  - metamodel
  - model
    - simpsons.families
  - build.properties
  - plugin.properties
  - plugin.xml
- Families.edit
- Families.editor
- Families.tests
- Families2Persons
  - Families2Persons.asm
  - Families2Persons.atl
- Persons
  - src
  - JRE System Library [JavaSE-1.6]
  - Plug-in Dependencies
  - META-INF
  - metamodel
  - model
    - simpsons.persons
  - build.properties

families.ecore | families.genmodel | persons.genmodel | Families2Persons.atl | simpsons.families | simpsons.persons

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:Persons="http://persons.ecore">
  <Persons:Male fullName="Homer Simpson"/>
  <Persons:Male fullName="Bart Simpson"/>
  <Persons:Female fullName="Marge Simpson"/>
  <Persons:Female fullName="Lisa Simpson"/>
</xmi:XMI>
```

Problems | Console | Error Log | Properties

Another bug: errors are not in Error log tab but in Console tab...

Kermeta

Advantages

- MDE tool with metamodeling, modeling and transformation

Drawbacks

- Not easy to understand the Kermeta notation and OCL
- Developing from metamodels to code is long and painful task

Labs on Kermeta (MDE)

- Install Kermeta
- Realize all the steps for a domain (Books-> Publications, your domain)
  - Source metamodel
  - Source model
  - Target metamodel
  - Transformation rules

Domain-specific Languages (DSL)

Domain-specific Languages (DSL)

A domain-specific language (DSL) is a programming or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain.

Domain-specific Languages (cont'd)

- MDA and MDE provide editors to create metamodels and models, ideal for users not fluent with computers
- Developers prefer some programming approaches
- Ease reuse by copy and paste

Domain-specific Languages (cont'd)

- You already know some DSL...

Domain-specific Languages (cont'd)

- Microsoft Excel macro commands
- Visual Basic for Applications
- *nix commands
- Matlab
- SQL

Domain-specific Languages (cont'd)

Several alternatives to realize DSLs:

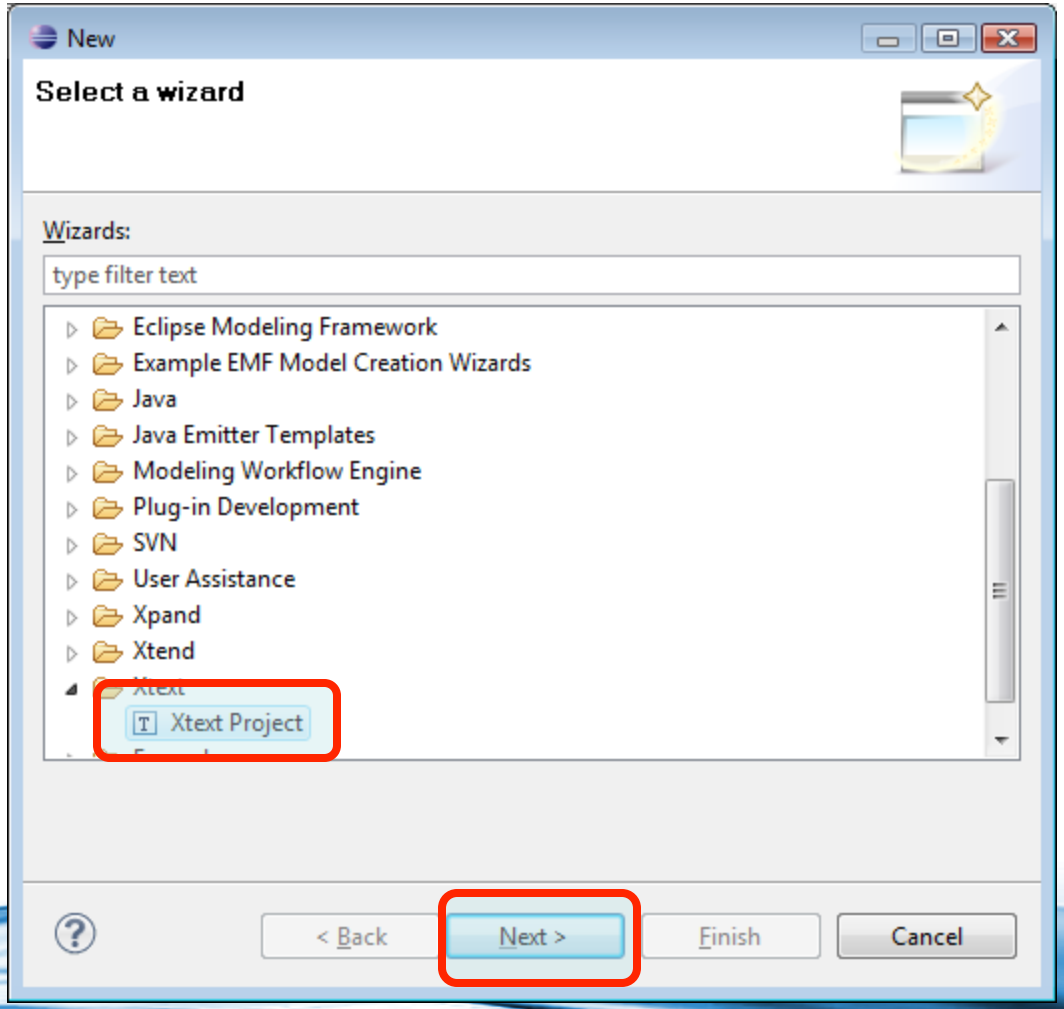
- *The old (geek?) school*: use a lexer (lex, flex) and a parser (yacc, bison) to create a DSL, use Emacs to use the DSL
- *The fashion way*: use a dynamic language (Smalltalk, Ruby)
- *The model-driven way*: use Xtext

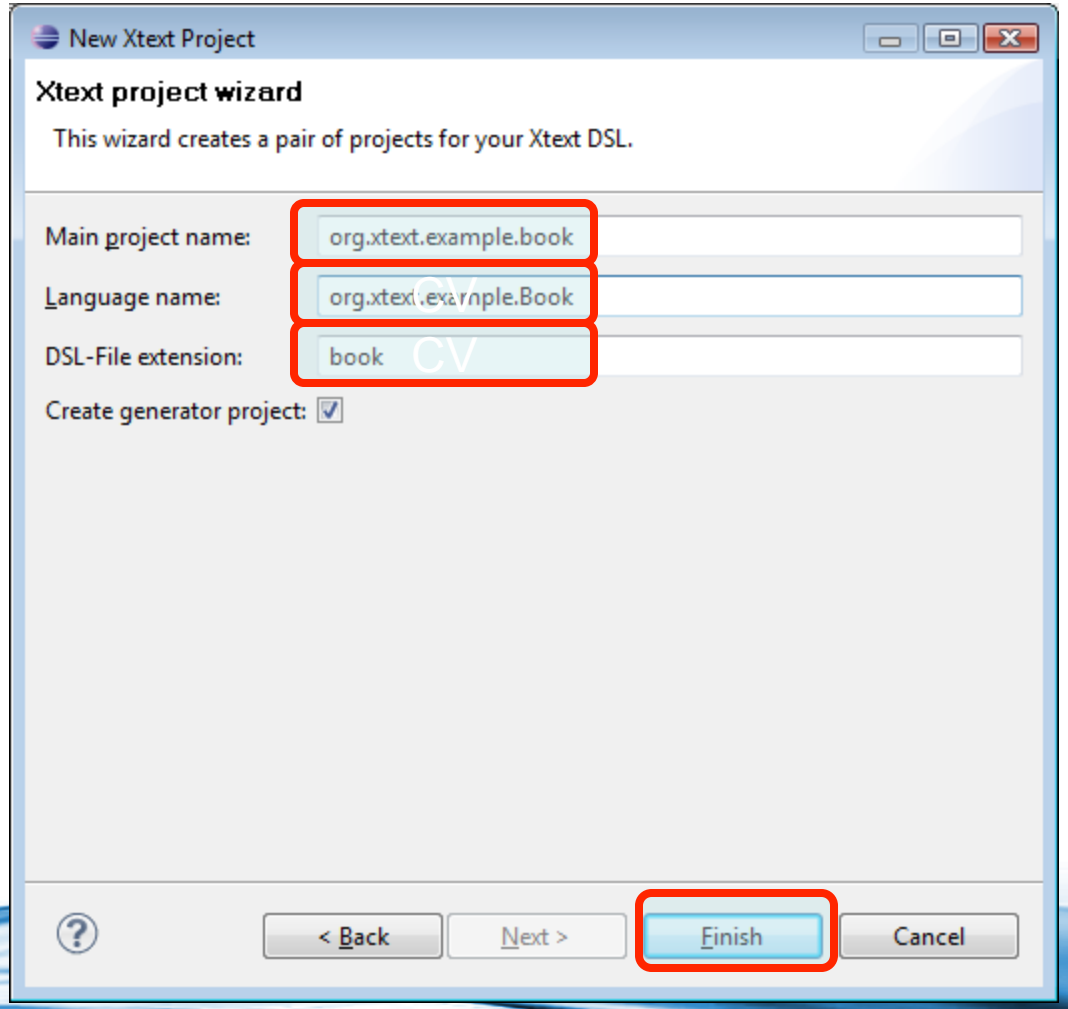
DSL in practice

Xtext

- Framework for the development of domain-specific languages
- Fully benefits of Eclipse
  - Syntax coloring
  - Code completion
  - Code templates
  - Eclipse perspectives and views
- Model to Text tool

Create a project
File>New>Xtext>Xtext Project




New Xtext Project
Xtext project wizard
This wizard creates a pair of projects for your Xtext DSL.
Main project name:
org.xtext.example.book
Language name:
org.xtext.example.Book
DSL-File extension:
book
Create generator project:
< Back
Next >
Finish
Cancel

Java - org.xtext.example.book/src/org/xtext/example/Book.xtext - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Package Exp | Hierarchy

- org.xtext.example.book
  - src
  - src-gen
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties
- org.xtext.example.book.generator
  - src
  - src-gen
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties
- org.xtext.example.book.ui
  - src
  - src-gen
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties

Book.xtext

```
grammar org.xtext.example.Book with org.eclipse.xtext.common.Terminals

generate book "http://www.xtext.org/example/Book"

Model:
    (imports+=Import)*
    (elements+=Type)*;

Import :
    'import' importURI=STRING;

Type:
    SimpleType | Entity;

SimpleType:
    'type' name=ID;

Entity :
    'entity' name=ID ('extends' extends=[Entity])? '{'
        properties+=Property*
    '}';

Property:
    'property' name=ID ':' type=[Type] (many?='[]')?;
```

Your grammar

Code generator

Editor, Code completion

Problems | @ Javadoc | Declaration

0 errors, 6 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (6 items)				

The grammar for your DSL, looks like a model...

Book.xtext

```
grammar org.xtext.example.Book with org.eclipse.xtext.common.Terminals

generate book "http://www.xtext.org/example/Book"

Model :
    (imports+=Import)*
    (elements+=Type)*;

Import :
    'import' importURI=STRING;

Type:
    SimpleType | Entity;

SimpleType:
    'type' name=ID;

Entity :
    'entity' name=ID ('extends' extends=[Entity])? '{'
        properties+=Property*
    '}';

Property:
    'property' name=ID ':' type=[Type] (many?='[]')?;
```

Book DSL

- Libraries
  - Contains Books
- Books
  - Title
  - One or more Authors
  - One or more Chapters
- Authors
  - First Name, Last Name, Date of Birth
- Chapters
  - Number
  - Number of Pages
  - Text

Here is our grammar in Xtext
Terminals between quote characters
Nonterminals with a ';'

Book.xtext

```
grammar org.xtext.example.Book with org.eclipse.xtext.common.Terminals

generate book "http://www.xtext.org/example/Book"

Model :
    (elements+=Book)*;

Book :
    'book' title=ID '{'
    authors+=Author+
    chapters+=Chapter+
    '}';

Author :
    'author' '{'
    'firstName' first=ID
    'lastName' last=ID
    '}';

Chapter :
    'chapter' number=ID '{'
    'pages' pages=ID
    'text' text=ID
    '}';
```

Java - org.xtext.example.book/src/org/xtext/example/Book.xtext - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Package Exp | Hierarchy

- org.xtext.example.book
  - src
    - org.xtext.example
      - Book.xtext
      - GenerateBook.mwe
      - GenerateBook.properties
  - src-gen
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties
- org.xtext.example.book.generator
  - src
  - src-gen
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties
- org.xtext.example.book.ui
  - src
  - src-gen
  - JRE System Library [jre6]
  - Plug-in Dependencies
  - META-INF
  - build.properties

Book.xtext

```
grammar org.xtext.example.Book with org.eclipse.xtext.common.Terminals

generate book "http://www.xtext.org/example/Book"

Model :
    (elements+=Book)*;

Book :
    'book' title=ID '{'
    authors+=Author+
    chapters+=Chapter+
    '}';

Author :
    'author' '{'
    'firstName' first=ID
    'lastName' last=ID
    '}';

Chapter :
    'chapter' number=ID '{'
    'pages' pages=ID
    'text' text=ID
    '}';
```

Run this file to create the language generator

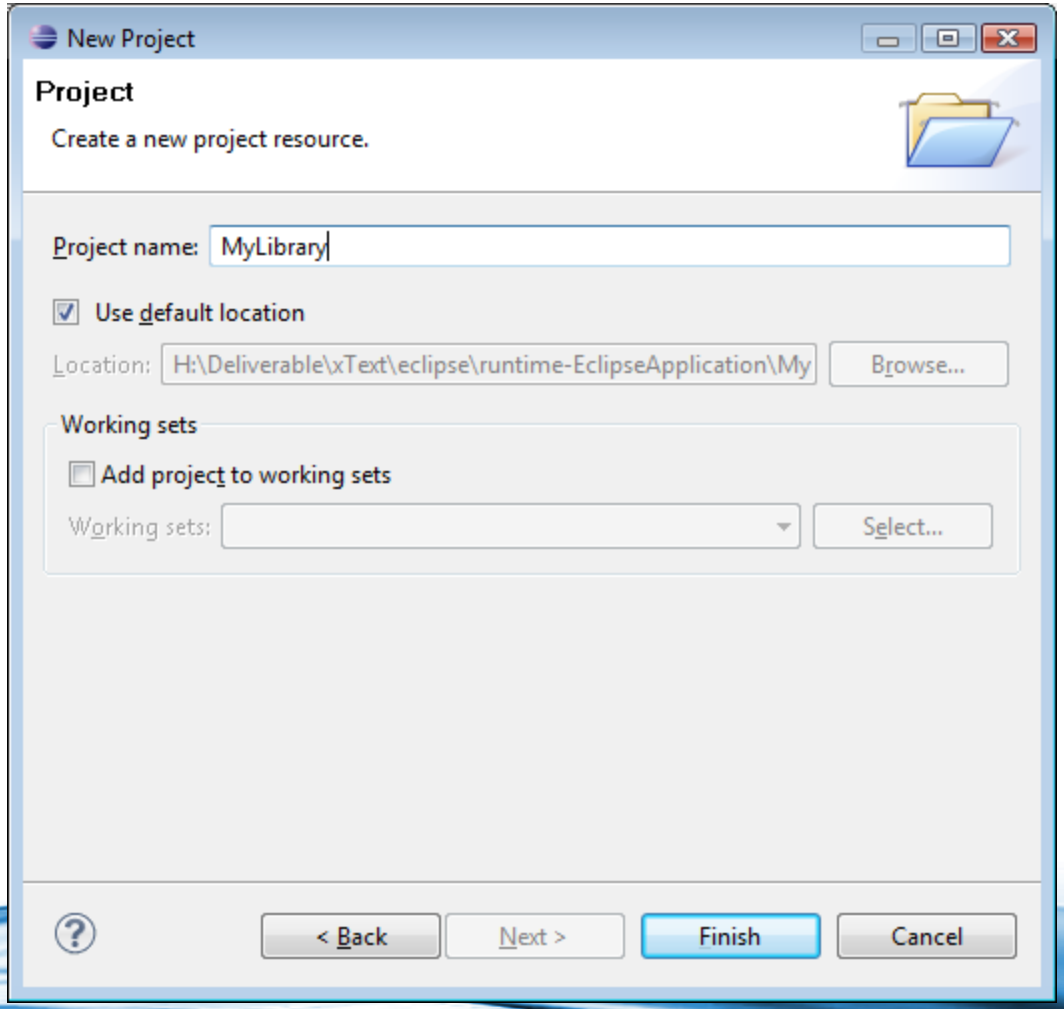
Problems | Javadoc | Declaration

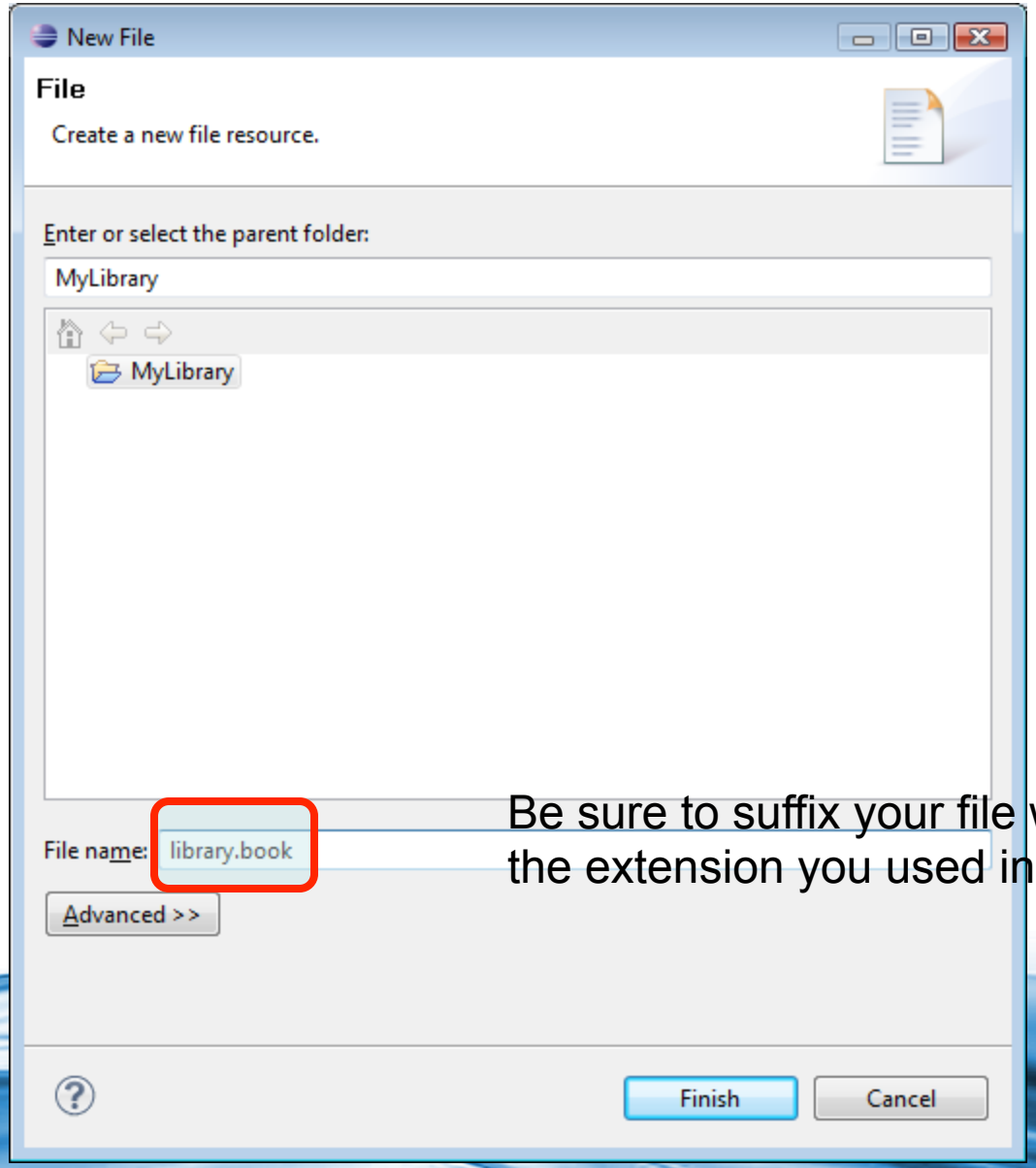
0 errors, 6 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (6 items)				

Then selects the project, and execute it
as an Eclipse application

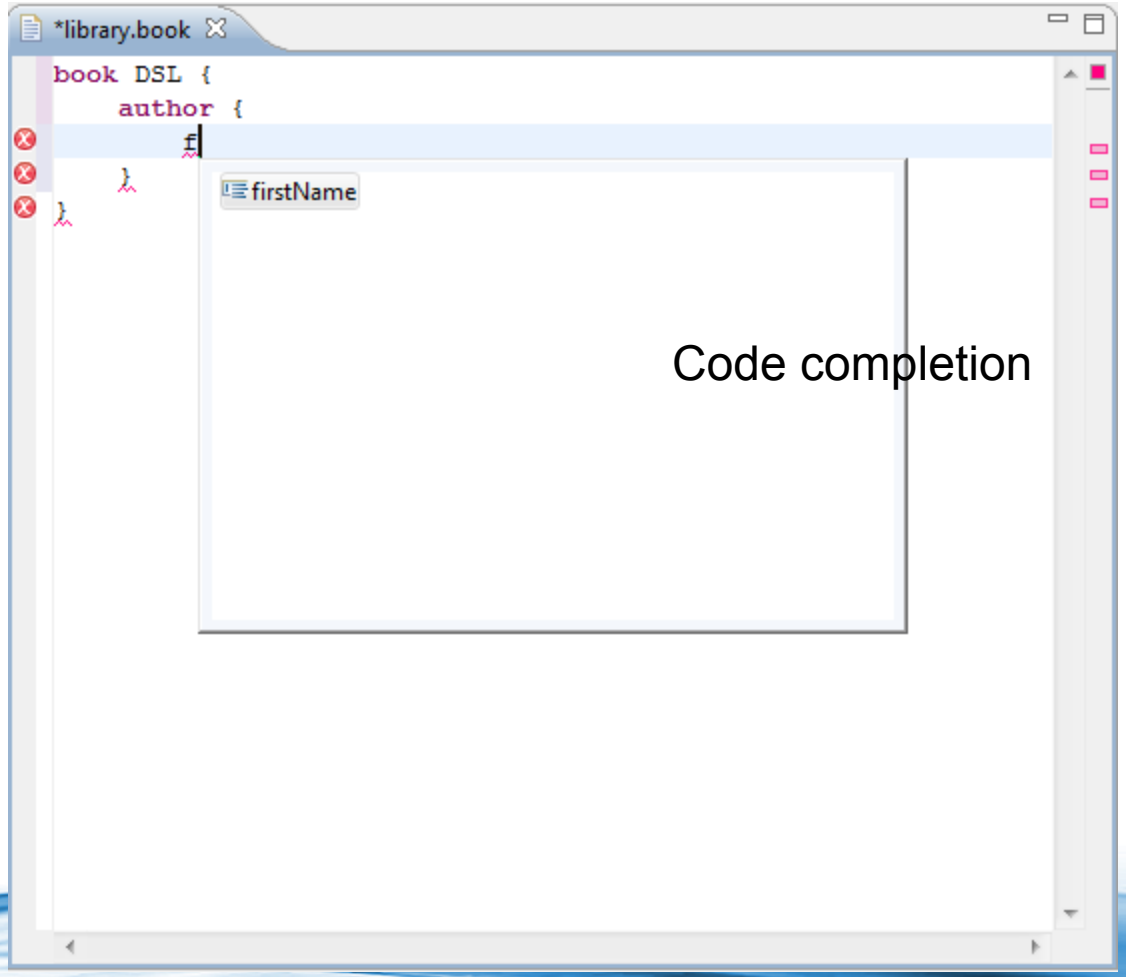
In this second Eclipse, create a new project File>New>General>Project





Be sure to suffix your file with the extension you used in the project

Syntax coloring



Code completion

The current grammar was limited, it is difficult to enter data, Update the grammar as follows

Book.xtext

```
grammar org.xtext.example.Book with org.eclipse.xtext.common.Terminals

generate book "http://www.xtext.org/example/Book"

Model :
    (elements+=Book)*;

Book :
    'book' isbn=ID title=STRING '{'
    authors+=Author+
    chapters+=Chapter+
    '}';

Author :
    'author' '{'
    'firstName' first=STRING
    'lastName' last=STRING
    '}';

Chapter :
    'chapter' number=INT '{'
    'pages' pages=INT
    'text' text=STRING
    '}';
```

One final step

- Xtext approach:
  - Code generating from the model defined before.
- Still incomplete in Xtext

Labs in Xtext (DSL)

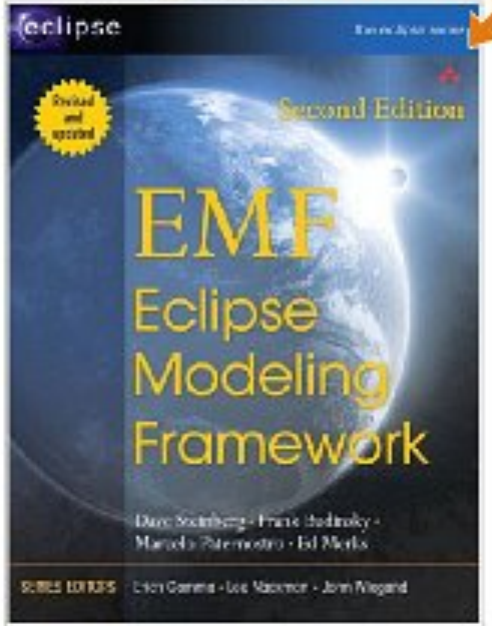
- Install Xtext
- Develop a DSL for publications (like Bibtex but simplify the metamodel)

Conclusion

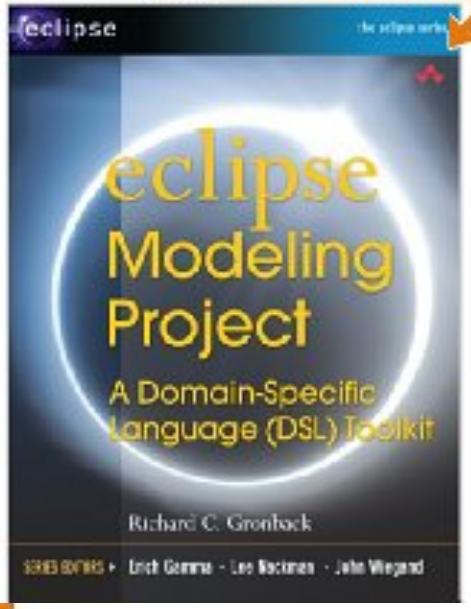
- Model-driven Architecture is a reality in industry
- Developers have to be trained to work on Model-driven Engineering
- Several tools available:
  - MDA
  - MDE
  - M2M
  - M2T

Thanks a lot for your attention

Cliquez pour **Feuilleter!**



Cliquez pour **Feuilleter!**



Cliquez pour **Feuille**