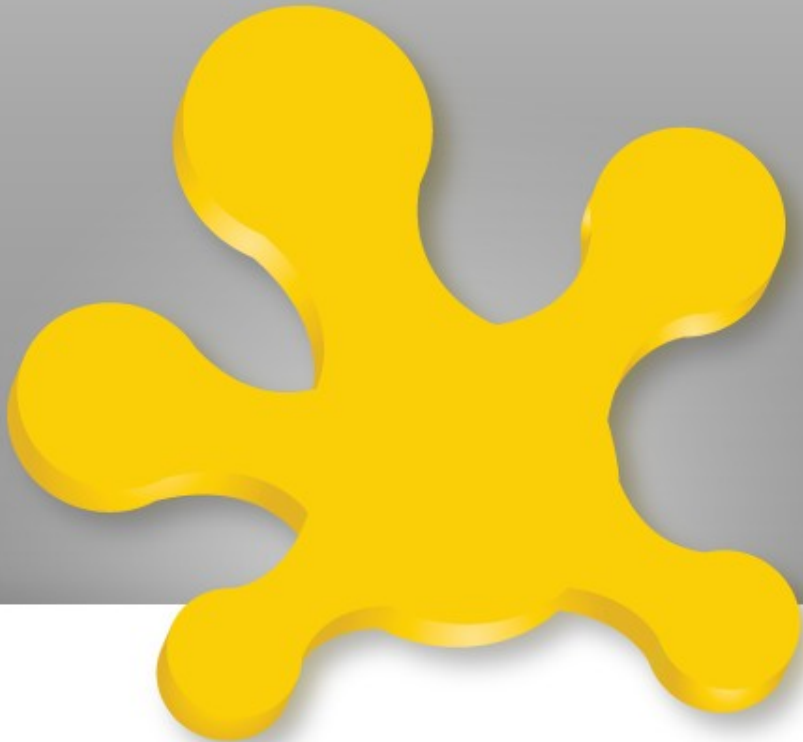


Krzysztof Daniel, Jakub Jurkiewicz

Eclipse Modeling Framework

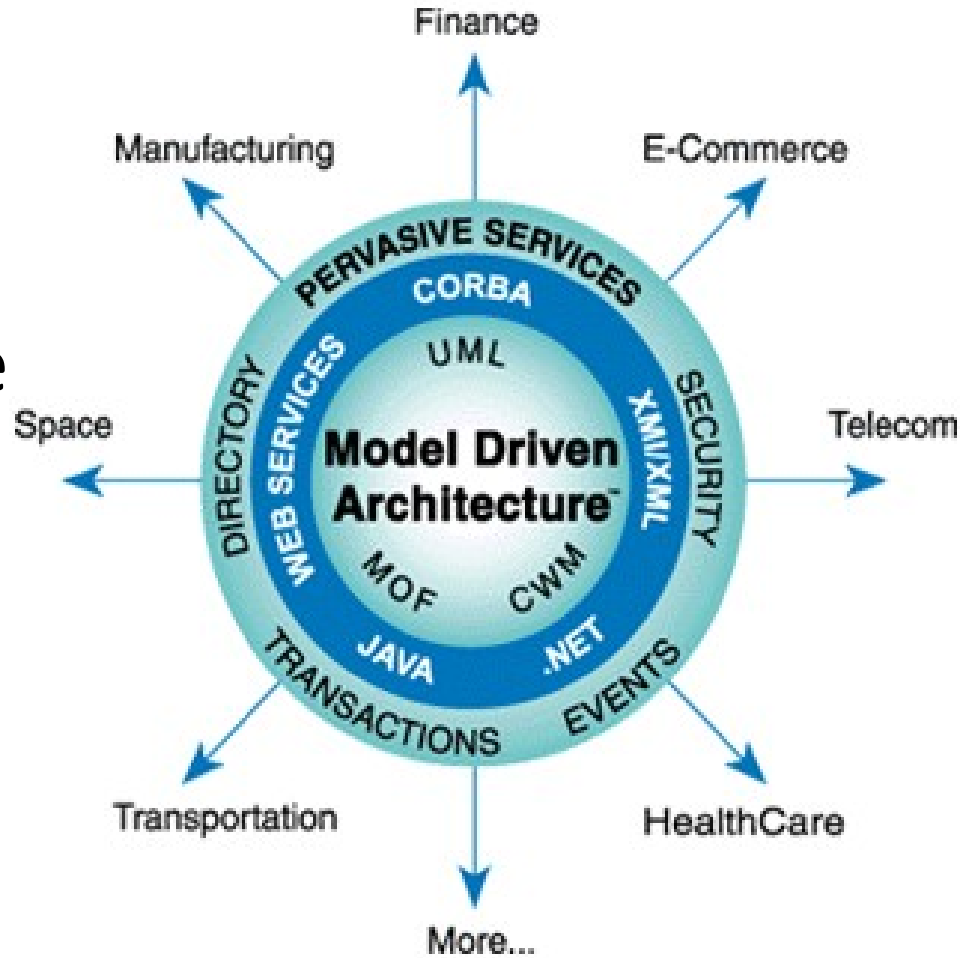


Let's move the Java world!



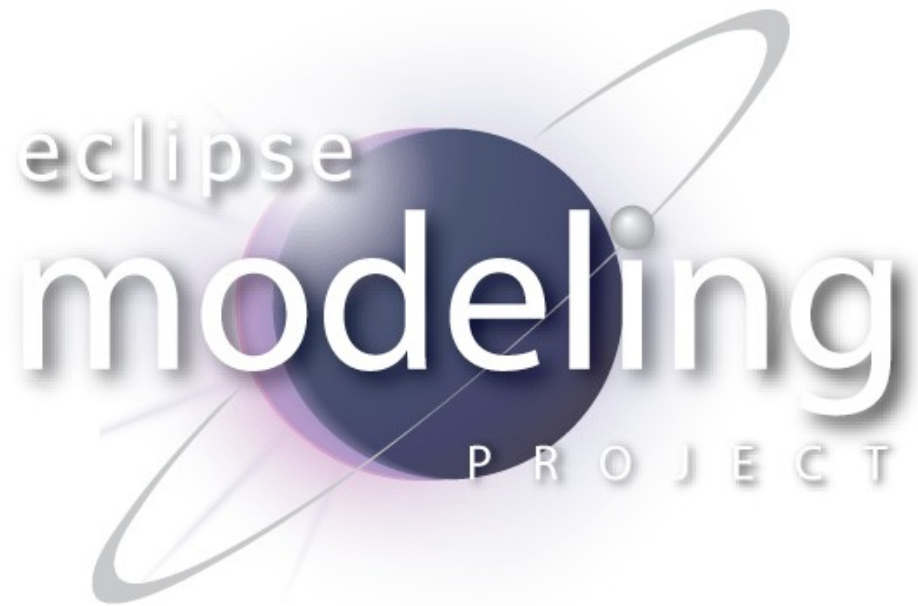
MDA

- Model Driven Architecture
- Approach for software modelling
- Created by OMG in year 2001



Eclipse Modeling Project

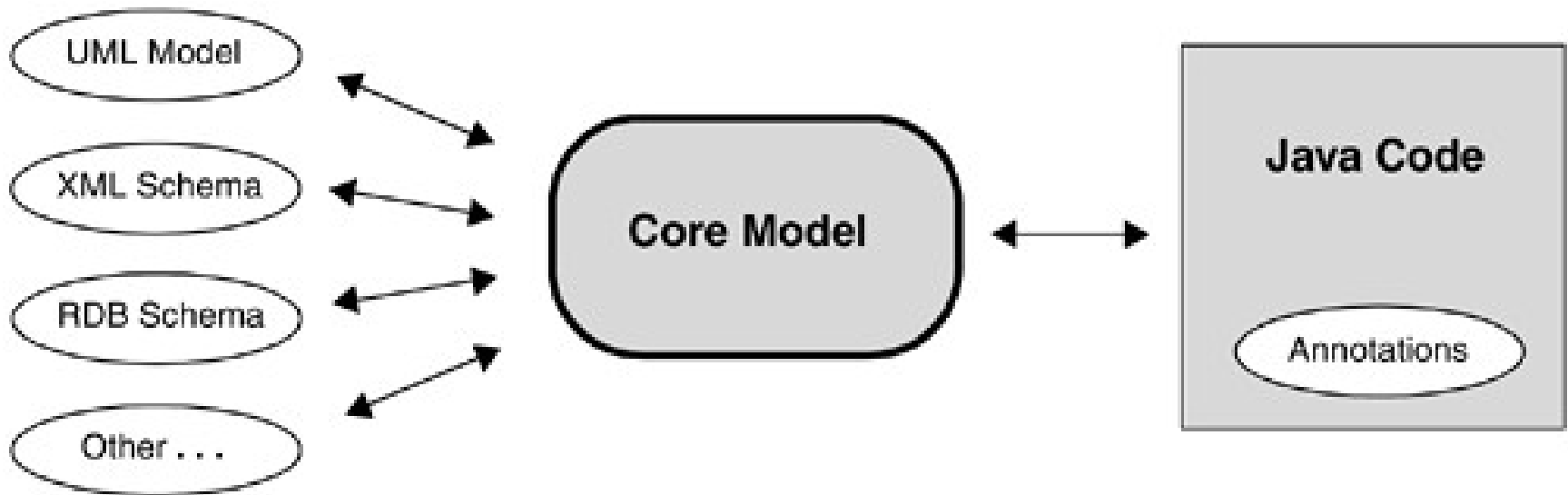
- Set of projects focused on modeling



Projects included in EMP

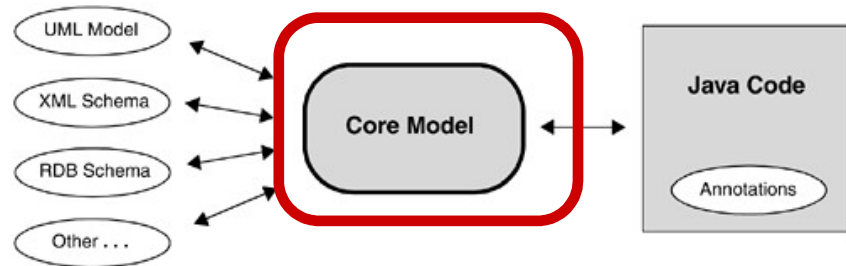
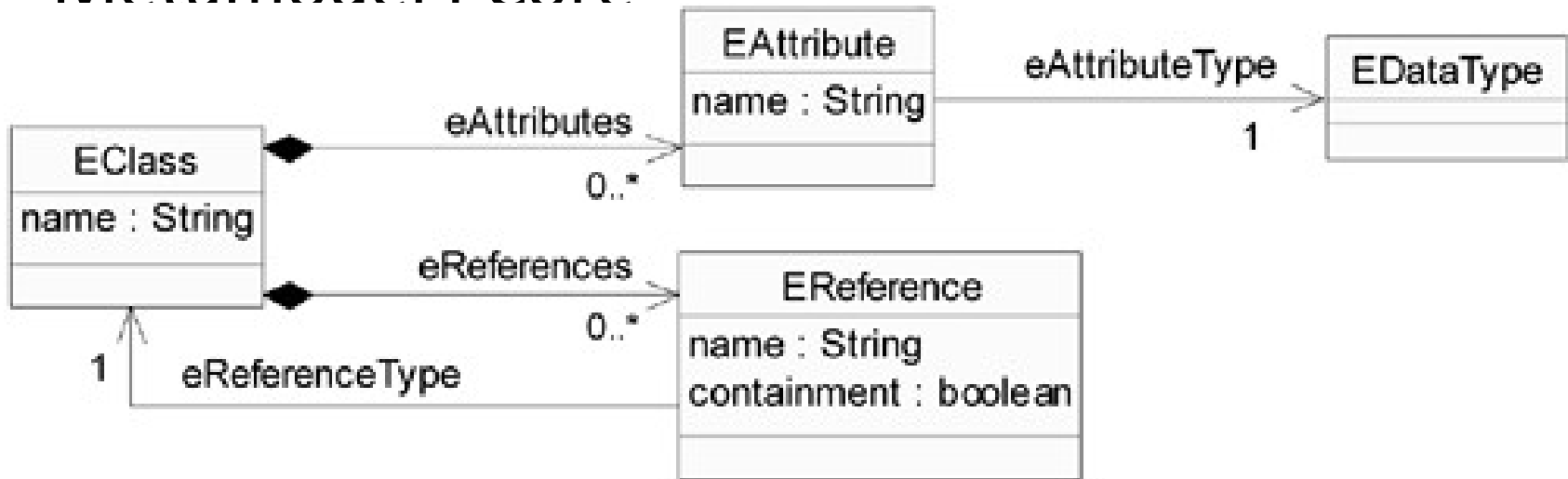
- Teneo – persisting EMF data models in a database based on Hibernate or JPOX/JDO2.0
- Model Query – searching and fetching the data from EMF model based on queries
- Model Transaction – responsible for cohesion when accessing and altering the data from EMF model (multithreading, undo-redo, etc.)
- Validation Framework – allows EMF model validation
- Java Emitter Templates (JET) – custom code generation

How does EMF work?



Basics of EMF

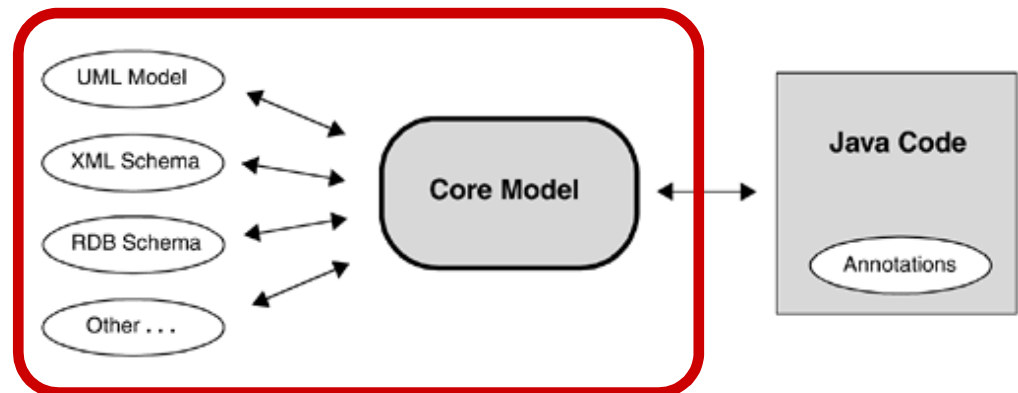
- Metamodel FCore



Let's move the Java world!

Basics of EMF

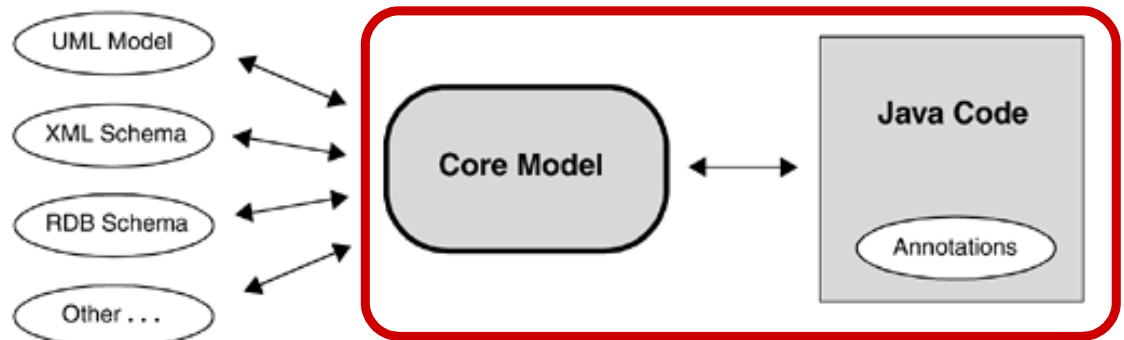
- How can we create EMF model?
- Manual XML file creation
- Import from UML (e.g. Rational Rose)
- Import from XML Schema
- Java code with annotations
- In graphical editors



Basics of EMF

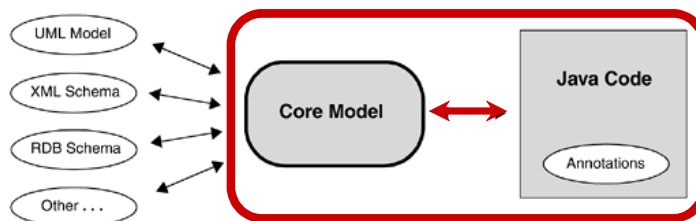
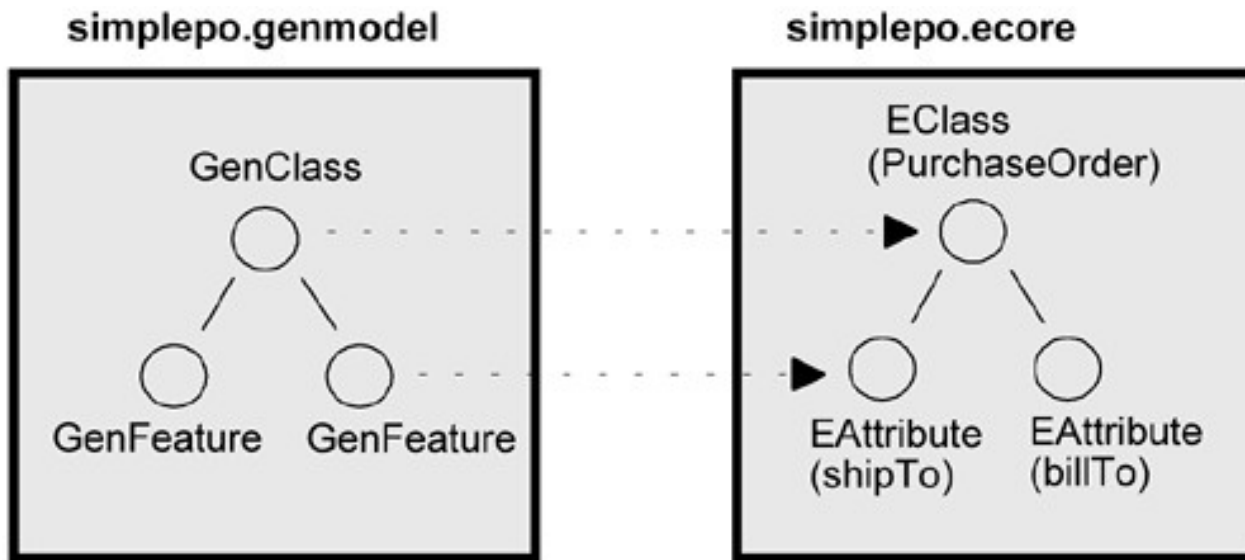
- Code can be generated multiple times
- Generated code has additional annotations:

```
/**  
 * @generated  
 */  
public String getShipTo() { ...
```



Generowanie kodu

- Ecore contains only information about model
- Genmodel – contains information about settings of the code generator



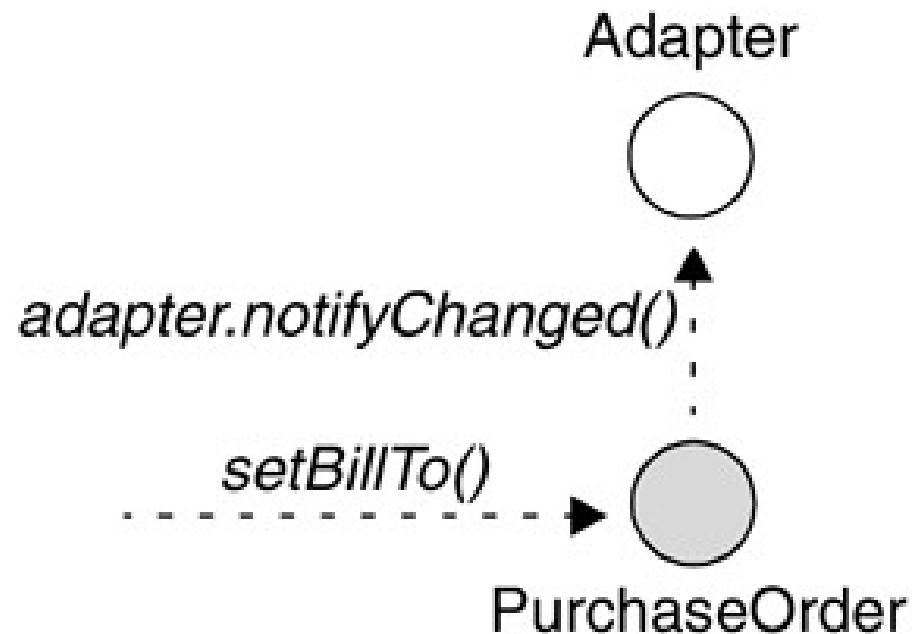
Let's move the Java world!

Listeners

- You can observe EMF objects and listen for changes

```
public interface EObject  
extends Notifier
```

- Method `notifyChanged()` is called everytime an object is changed

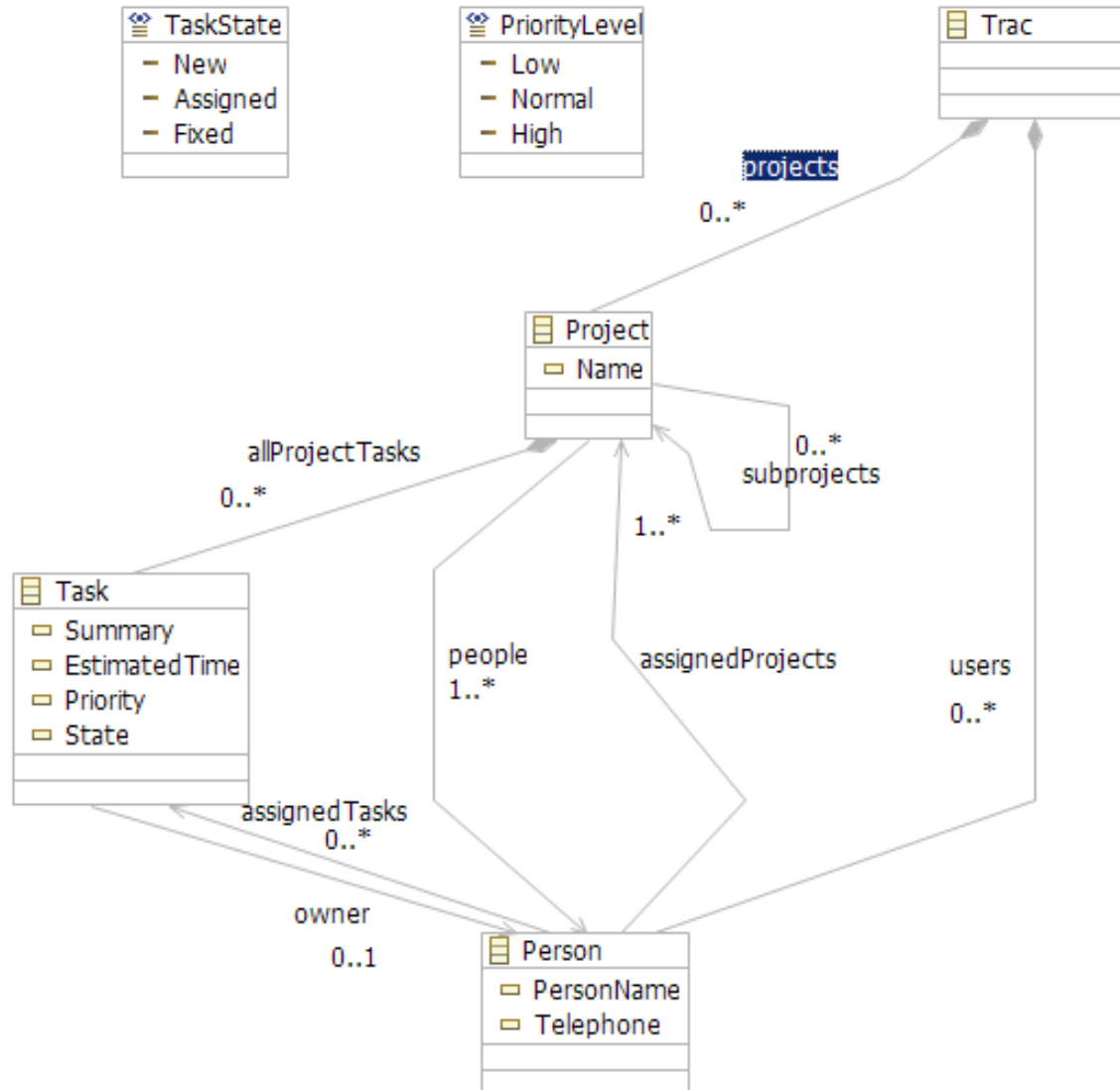


Exercise #1

Task

- Create a given model
- Generate the Java code
- Run the generated editor

Model

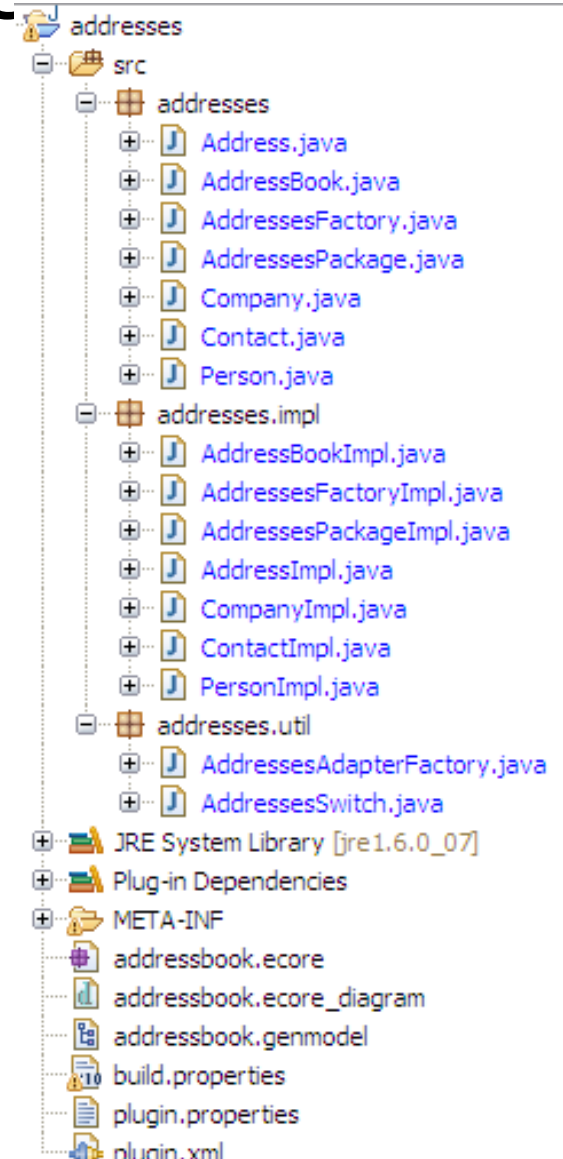


Generated code

- Pure Java model together with additional operations
- Project .edit – helper classes for model edition
- Project .editor – Eclipse editor
- Project .tests – place for unit tests

Generated code

- Interfaces
- Implementations
- Factories for creating model instances
- Package class for access via reflection mechanism



Q & A