



 **EGF Exercices – EMF – UC2**  
**EGF 0.4.0**

**Benoît Langlois – Thales/EPM**

## Understanding how to customize the EMF generation with EGF

### Correction of the exercices

- ▶ Download the org.eclipse.egf.usecase.emf.uc2.zip file from the [EGF wiki](#)
- ▶ Use the libray model and project in the org.eclipse.emf.examples.library.zip file

### Prerequisite

- ▶ Installation of Eclipse 3.6 and egf 0.4.0
- ▶ Read the EGF Tutorial
- ▶ Read the « reuse and customization » EGF tutorial
- ▶ Read the « EMF Generation Patterns » EGF tutorial
- ▶ Understand the « EMF-UC1 » use case



## Problem Statement

- ▶ Writing a first factory component which applies a basic EMF generation with patterns from a genmodel file
- ▶ Indications:
  - ▶ Cf. UC EMF UC1 / Exercice #2

## Learning

- ▶ Simple Emf generation without customization

## Difficulty

- ▶ 1/5

## Correction

- ▶ EMF\_UC2\_1\_Library.fcore

## Problem Statement

- ▶ Redefine the « Writer.[set | get]Name » methods, « Library.getBooks » and « Library.getStock» methods
- ▶ Indications:
  - ▶ Create four patterns: both 1) for Writer and Library, and 2) for generating the get and set methods. The get/set patterns inherit from the « Class.getGenFeature.TODO.override » and « Class.setGenFeature.TODO.override » patterns.
  - ▶ In the preCondition, write the precondition to apply the pattern
  - ▶ In a doGenerate method, write the get/set generation code
  - ▶ Invoke « EMF [Model | Edit | Editor] Pattern » factory component. For the Model, add a pattern substitution. This is a list of three substitution [super-pattern EMF pattern] / [pattern created for overriding]
  - ▶ In the plugin.xml dependencies, don't forget to add a dependency toward « org.eclipse.egf.emf.pattern » which contains the « Class.getGenFeature.TODO.override » pattern

## Learning

- ▶ Override of get/set feature

## Difficulty

- ▶ 3/5

## Correction

- ▶ EMF\_UC2\_2\_Library\_ClassGetSetGenFeature.fcore

## Problem Statement

- ▶ In the Person Class from the Library model, a « getFullName » Operation is added. Define a pattern to implement the method body.
- ▶ Indications:
  - ▶ Create a pattern which inherits from the « Class.implementedGenOperation.TODO.override ».
  - ▶ In the implementation part, redefines the contents of the « doGenerate » method with the expected code.
  - ▶ In the preCondition, write the precondition to apply the pattern: the current genOperation equals « getFullName »
  - ▶ Invoke « EMF [Model | Edit | Editor] Pattern » factory component. For the Model API generation, add a pattern substitution. To the substitution list, add the new created pattern.
  - ▶ In the plugin.xml dependencies, don't forget to add a dependency toward « org.eclipse.egf.emf.pattern » which contains the « Class.implementedGenOperation.TODO.override » pattern

## Learning

- ▶ Override of operation

## Difficulty

- ▶ 3/5

## Correction

- ▶ EMF\_UC2\_3\_Library\_ClassImplementedOperation.fcore

## Problem Statement

- ▶ Emf Model level: Realize an implementation of derived attributes and references
- ▶ Emf edit level: Redefine the getText and getImage
- ▶ Apply it the « extlibrary\_extension.ecore » model stored in the « org.eclipse.egf.usecase.emf.uc1 » plugin
- ▶ General indications:
  - ▶ Like EMF with insert and override, a pattern inherits from Class.insert and Class.getGenFeature.TODO.override
  - ▶ To redefine a getText method, a pattern inherits from the ItemProvider.getText.override pattern
  - ▶ For the getImage, the overlay method defined in a super-class can be redefined in the current Class; to insert this overlay redefinition (i.e., with a new method in the class), a pattern inherits from the ItemProvider.insert pattern

## Learning

- ▶ Overwrite of derived features and item providers

## Difficulty

- ▶ 3/5

## Correction

- ▶ EMF\_UC2\_4\_LibraryExtension\_ClassInsert\_ItemProvider.fcore

## Problem Statement

- ▶ Refines the Edit and Editor « plugin.properties » file contents.
- ▶ General indications:
  - ▶ Create two patterns, for Edit and Editor. The Edit[or] pattern inherits from the « PluginProperties » pattern in the EMF Edit[or] Pattern factory component.
  - ▶ Next, there are two possibilities to change the plugin.properties file. See below.
  - ▶ Invoke « EMF [Model | Edit | Editor] Pattern » factory component. For the Model API generation, add a pattern substitution. To the substitution list, add the new created patterns.
  - ▶ In the plugin.xml dependencies, don't forget to add a dependency toward « org.eclipse.egf.emf.pattern » which contains the « Class.implementedGenOperation.TODO.override » pattern
- ▶ Indications – Solution 1 – Append in-contents:
  - ▶ This solution adds / redefines properties at the end of the file. This solution is just given for illustration.
  - ▶ In the implementation part, create a method which declares the new properties code and, in the orchestration, add it at the end.
- ▶ Indications – Solution 2 – Redefinition:
  - ▶ In the implementation part, redefine the contents of the « doGenerate » method with the expected code. Generally, we start from a copy/paste from the original code and next adaptations are made. Pro: accurate modifications. Con: maintenance.

## Learning

- ▶ Redefinition of specific parts in an EMF generation

## Difficulty

- ▶ 3/5

## Correction

- ▶ EMF\_UC2\_5\_DomainIndependent\_PluginProperties.fcore