THALES

EGF Exercices – Pattern – UC3
Reporting of Model-to-Text Transformations

Benoît Langlois – Thales/TGS

EGF - Thales Global Services - EPM

Objective of this document

Understanding how to realize M2T reporting and post-processing with EGF patterns

Plugin

- org.eclipse.egf.usecase.pattern.uc3.m2treporting

Prerequisite

- Read the EGF Tutorial
- Read the EGF Tutorial / Reporting of Model-to-Text Transformations

THALES

Exercice #1

Problem Statement

- Writing a pattern which displays on the console the list of classes of an ecore model

Learning

- Report on console

Difficulty

- 1/5

Correction

- Pattern_UC3_1_ReportOnConsole.fcore

THALES

Exercice #2

Problem Statement

- Writing a pattern which writes, in one file, the list of classes of an ecore model

Learning

- Using a reporter to write a generation result from patterns in one file

Difficulty

- 2/5

Correction

- Pattern_UC3_2_ReportInOneFile.fcore
- The result is created in a « test » project, folder « Pattern_UC3 »
- Two versions are proposed:
  - One factory component integrating the pattern and the production plan
  - A dissociation of a FC launcher which provides a reporter to a common FC which contains the pattern and the production plan

THALES

Exercice #3

Problem Statement

- Writing a pattern which writes for each class, in separate files, the name of this class
- Difference with the previous exercice: just changing the reporter which splits the result of each pattern class by file (and not in one file)

Learning

- Using a reporter to write a generation result from patterns in several files

Difficulty

- 2/5

Correction

- Pattern_UC3_3_ReportInSeveralFiles.fcore
- Reuse of the common FC, created during the previous exercice, called with a FC providing the specific reporter

Exercice #4

Problem Statement

- Like Exercice #2, writing a pattern which writes in one file the names of ecore classes
- Defining a sub-task of « Domain-Driven Pattern Strategy » which adds two parameters for setting the Project and Folder Names where the file is generated

Learning

- Adding new report parameters to a task which are used by a reporter

Difficulty

- 3/5

Correction

- Pattern_UC3_4_ReportWithAdditionalReportParameters.fcore

THALES

Exercice #5

Problem Statement

- Implementing a factory component :
  - containing two patterns for class and attribute
  - which displays the name of each class and attribute
- Next, writing a class (e.g., MyPatternProcessor.java) for post-processing the ouput of the pattern-based generation

Learning

- Understanding the mechanism of M2T post-processing

Difficulty

- 3/5

Correction

- Pattern_UC3_5_patternOutputProcessor.fcore

THALES