THALES

EGF Tutorial
Reporting of Model-to-Text Transformations

Benoît Langlois – Thales/TGS

EGF - Thales Global Services

The issue

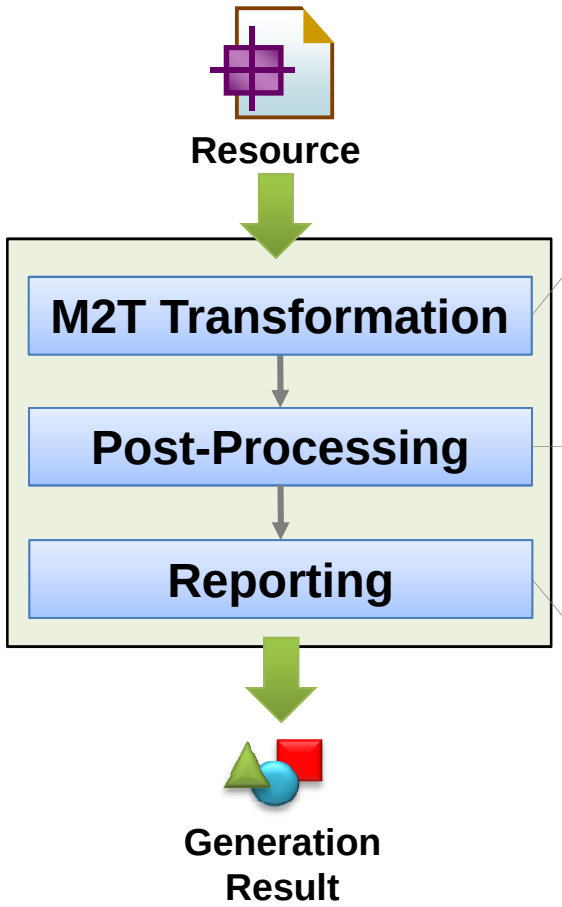
- **The M2T transformation generally combines:**
  - The logic of text generation from a model,
  - Text transformations collateral to M2T transformation,
  - The logic of storage of the generation result.

- **The objective is to dissociate those three concerns**

Need of Post-Processing and Reporting

- **Needs of Post-Processing:**
  - Need #1: Ability to realize text transformations following the M2T, independently of the logic of M2T transformation, such as text reformatting
  - Need #2: Ability to transform text with a context independent of the M2T context
- **Needs of Reporting:**
  - Need #3: Ability to dissociate the logic of M2T storage from the logic of M2T transformation (e.g., storage in one or several files, at one or another location)
- **Common Needs:**
  - Need #4: Ability to consider the post-processor (managing the post-processing) and the reporter (managing the reporting) as parameters of the M2T transformation, in order to reuse the M2T transformation in different ways

Steps of the Text Generation Activity

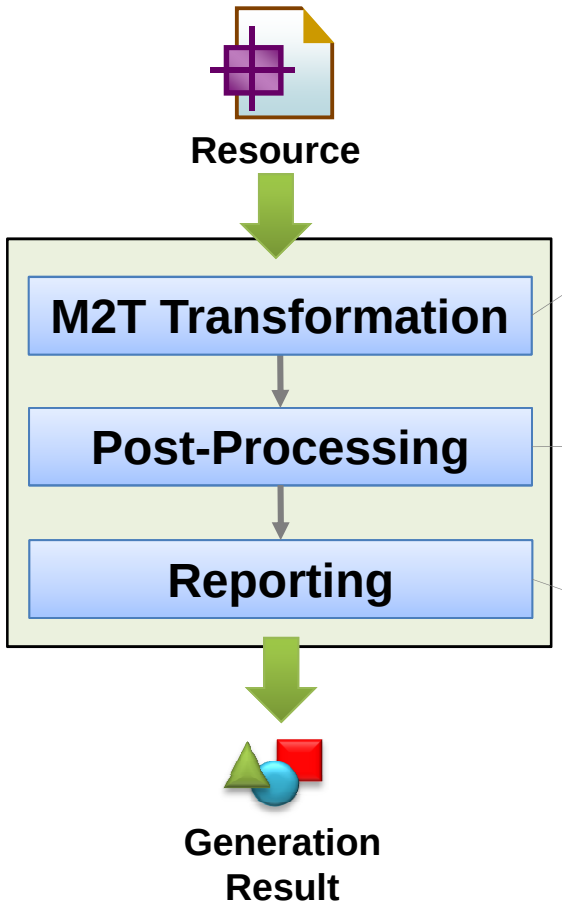


Heart of the generation, this steps consists in transforming a model into text with patterns

This step consists in applying a text transformation independent of the logic of M2T transformation

This step consists in realizing the final reporting operations and storing the generation result

THALES

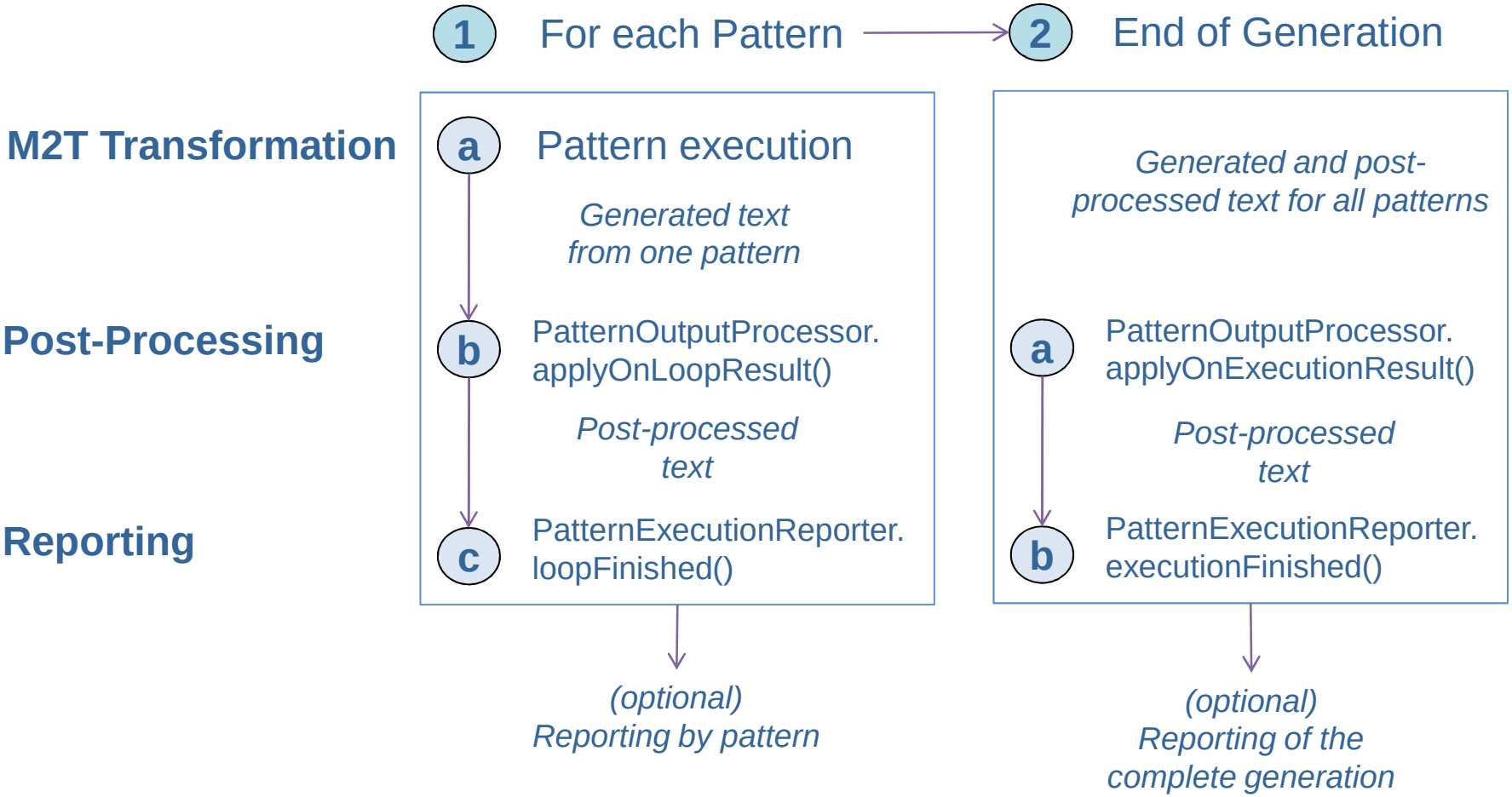


Choice of the M2T language and associated engine, transformation strategy, resource visitor

Choice of the post-processor

Choice of the reporter

Process for Reporting a M2T Transformation



THALES

Organization of the Report

- **The report is incrementally created during the generation**
- **Internal structure of the report**
  - The report structure is built during the pattern execution
  - The report structure is organized as a composite pattern
    - A Container reflects the pattern call orchestration
    - A DataLeaf contains the generated and post-processed text
  - The report tree corresponds to a syntactical tree for reporting

- **Modifying and Building up the final text**
  - Each DataLeaf is accessible by navigating over the report tree; for identification, each Container is associated to its source pattern
  - The final text, transmitted to the reporter, is built by navigating over the Containers of the report tree and concatenating the DataLeaf texts of the Containers

THALES

Reporter Environment

- **The report tree is not the concern of the reporter, which has no visibility on**

- **LoopFinished environment:**
  - There is one loop by tuple of resource elements matching the pattern parameters. Ex: each class of an ecore model instance which matches an EClass pattern parameter.
  - parameterValues: key/value of each pattern parameter value
  - Output: result of pattern execution for one tuple; this output is already post-processed.

- **executionFinished environment:**
  - Output: final result of the execution of all the patterns ; this output is already post-processed.

THALES