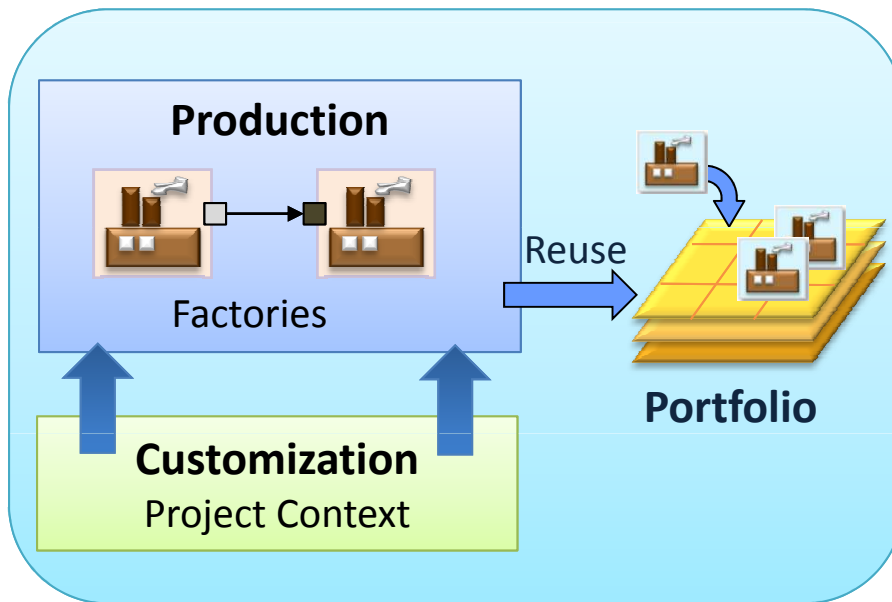




## A Quick Tour of EGF

**Benoît Langlois – Thales/EPM**

- ▶ EGF (Eclipse Generation Factories) is an Eclipse open source component project in incubation under the EMFT project.
- ▶ **Purpose:** providing a **model-based generation framework**
- ▶ **Objectives:**
  - ▶ Supporting complex, large-scale and customizable generations
  - ▶ Promoting generation portfolios in order to capitalize on generation solutions
  - ▶ Providing an extensible generation structure



## Main Features:

- ➔ Software production with **factory components**
- ➔ **Reuse** of off-the-shelf factory components
- ➔ Development by **assembly**
- ➔ **Customization** in a specific context
- ➔ Capitalization with **executable patterns**

# Types of Generation Units

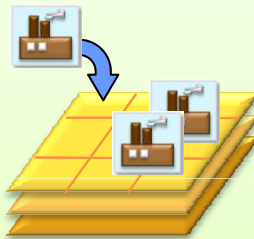


## Factory Component

Composite generation unit with an activity workflow

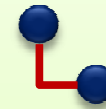
### Task

Leaf generation unit written in a language (e.g., Java, Ruby)



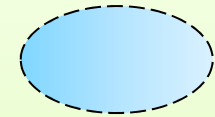
## Portfolio

Capitalization on a specific generation topic



## Generation Chain

High generation view to organize complex generations



## EGF Pattern

Description of systematic behavior

# Example: Enhanced EMF Generation



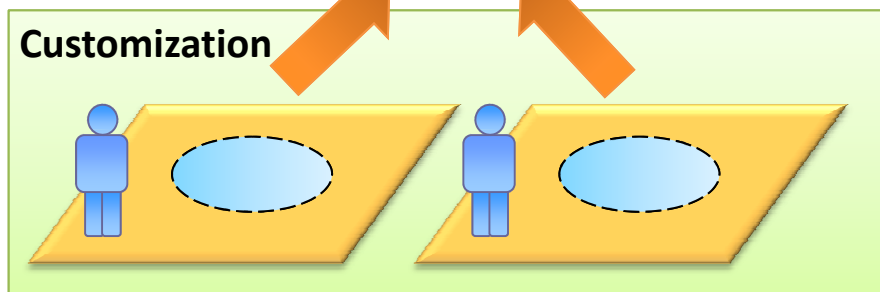
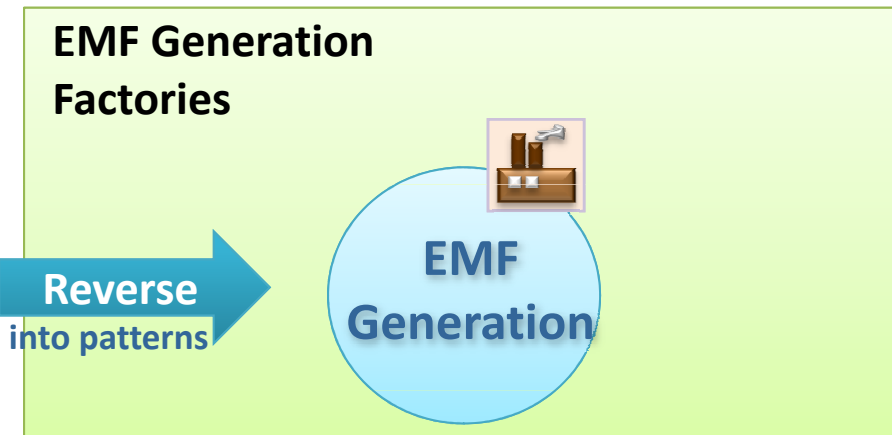
## EMF Generation completely ensured by EGF

- Example of complex generation
- Interest of EMF customization
- Open customization between team members

EGF: Eclipse Generation Factories – Thales Corporate Services/EPM

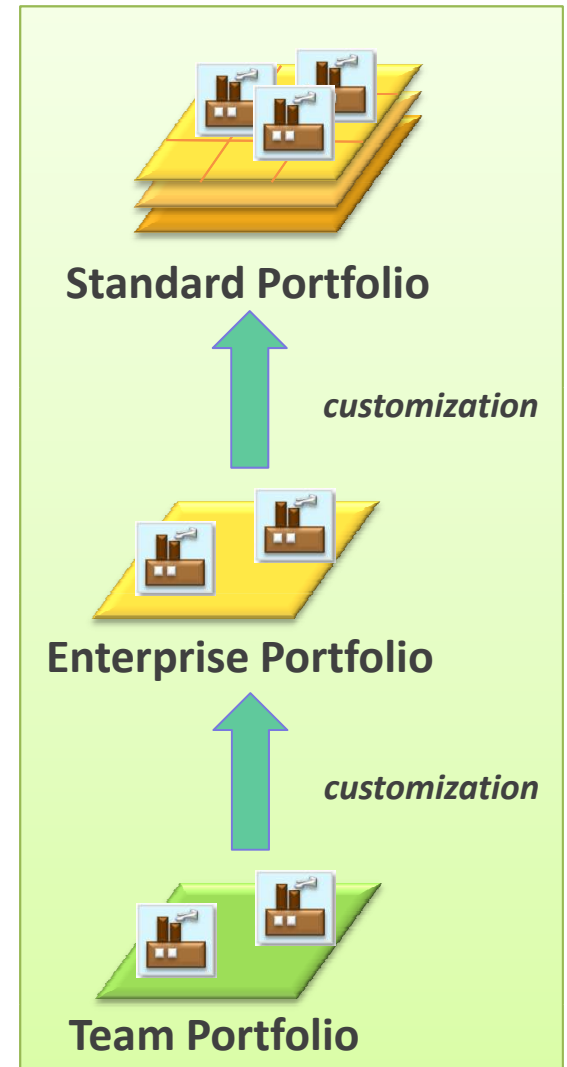


**EMF Generation**



Enriching the standard generation  
Accurate customization of the EMF generation

## Several levels of Customization



# Factory Component Lifecycle

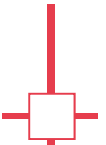


Factory  
Component  
Edition

Factory  
Component  
Execution

The screenshot shows the Eclipse IDE interface. At the top, the 'activityWorkflowUC1.fcore' editor displays a production plan with tasks like 'Change 1 value in Java', 'Change 2 value in Ruby', and 'Change 3 value in Java'. Below, two code editors are open: 'HelloJava.java' and 'HelloRuby.rb'. The Java code implements the 'preExecute' and 'doExecute' methods, while the Ruby code overrides them. At the bottom, the 'EGF Console' shows the output of the execution: 'Hello from Java', 'Hello from Java from Ruby', and 'Hello from Java from Ruby from Java'.

- ➔ Management of the workspace / target platform / runtime environment
- ➔ Edition/Execution of the activity workflow (production plan)



<http://www.eclipse.org/egf>  
<http://wiki.eclipse.org/EGF>