

# Thales Global Services

## Développement d'usines logicielles avec EGF

INFORMATION  
HUMAN RESOL  
PURCHASING  
COMMUNICATI  
ENGINEERING  
TRANSFORMATION  
REAL ESTATE & FM

**Thales Global Services**  
the essence of expertise

07/06/2012 – NEPTUNE 2012



6-7 juin 2012, Paris

OPEN

**THALES**

# Plan

1. **Les usines logicielles**
2. **EGF**
3. **Intégration d'un outil d'usines logicielles dans un processus logiciel**

# Plan

1. Les usines logicielles
2. EGF
3. **Intégration d'un outil d'usines logicielles dans un processus logiciel**

# Pourquoi un besoin d'usines logicielles ?

## Constat

- ◆ Complexité croissante du logiciel

## Problématique

- ◆ Déterminer comment intervenir tout au long du processus de développement et de maintenance en vue d'améliorer la maîtrise du triplet coût-délais-qualité
- ◆ Les usines logicielles représentent une solution à cette problématique

# Qu'est-ce qu'une usine logicielle?

## Definition

- ◆ Une usine logicielle est un **environnement** (par ex. éditeurs, frameworks, compilateurs) qui vise à **automatiser** un ensemble de tâches humaines répétitives
- ◆ **Moyen technique** qui contribue à la **rationalisation** de la production logicielle

## Objectifs

- ◆ Augmenter la **performance** et la **productivité** des équipes logicielles
- ◆ Réduire les coûts de production du logiciel par réalisation d'**économies d'échelle** sur les tâches répétitives
- ◆ Gérer le **passage à l'échelle** lorsque la quantité d'information à traiter augmente
- ◆ Améliorer la **fiabilité** et la **qualité** des artefacts produits
- ◆ Capitaliser du **savoir-faire**

# Positionnement

## Une définition assez généraliste

- ◆ Les usines logicielles s'appliquent à différents degrés et phases du cycle de développement logiciel
- ◆ Diversité des moyens de mise en œuvre

## Cadre de cette étude

- ◆ Restriction aux usines logicielles basées sur la technique générative

## Technique générative

- ◆ Elle inclut tous les moyens qui permettent la génération logicielle
- ◆ Exemples:
  - Générateur
  - DSL (Domain-Specific Language)
  - Ligne de produit

# ATTENTES DES USINES LOGICIELLES

- ① Générer en série
- ② Intégrer des langages, des outils hétérogènes, des patterns et des frameworks
- ③ Créer, assembler et mettre sur étagère des composants d'usines logicielles
- ④ Supporter la variabilité de génération
- ⑤ Faciliter la description de génération au moyen de DSLs

# Plan

1. **Les usines logicielles**
2. **EGF**
3. **Intégration d'un outil d'usines logicielles dans un processus logiciel**



# EGF (Eclipse Generation Factories)

## Un composant Eclipse

- ◆ Composant du projet **Modeling** de la plate-forme Eclipse
- ◆ Intégration à la **livraison Eclipse** (Indigo, Juno)
- ◆ Finalité: Un outil pour le **développement d'usines logicielles**

# Architecture d'EGF

Portfolio EGF

EGF

Extensions

Moteur EGF



Fondements



Métamodèles EGF



Comportements de base,  
exécution dynamique



Factory component, Task



Pattern EGF

OPEN

# Architecture d'EGF

Portfolio EGF

EGF

Extensions

Moteur EGF



Extensibilité du moteur



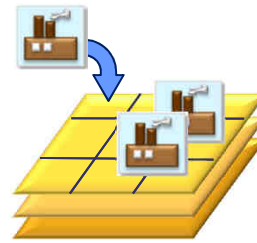
Nouveaux langages et outils



Nouveaux formalismes de génération

# Architecture d'EGF

Portfolio EGF



Usines logicielles sur étagère

Portfolio

**EGF**

Extensions

Moteur EGF

Portfolios fournis avec EGF



**Extension Génération EMF**



**Editeur et générateur pour l'intégration continue**  
(Hudson/Jenkins, Buckminster aujourd'hui)

OPEN

# 4 formalismes pour développer des usines logicielles



## Task / Tâche

- ▶ Unité atomique pour l'exécution de code dans un langage ou un outil



## Factory Component / Composant d'usine

- ▶ Unité composite de génération



## Generation Chain / Chaîne de génération

- ▶ DSL pour faciliter la définition et la maintenance de génération



## Pattern EGF

- ▶ Formalisme déclaratif pour traiter une ressource (ex: modèle)
- ▶ Implémentation dans un langage (ex: Jet pour modèle-vers-texte)

# Correspondance entre attentes et EGF

| ATTENTES                           | EGF   |
|------------------------------------|---|
| Attente 1 –<br>Génération en série | Utilisation de 1) <b>tâches</b> dédiées outils, 2) <b>patterns</b> EGF  |
| Attente 2 –<br>Intégration         | <b>Assemblage</b> de composants d'usines et de tâches de nature hétérogène dans un <b>plan de production</b>  |
| Attente 3 – Usines<br>sur étagères | Partitionnement et déploiement de composants d'usines<br><b>Usine logicielle</b> = composant d'usine avec une finalité dans un processus logiciel<br><b>Portefeuille d'usines logicielles</b> = ensemble d'usines logicielles capitalisant un savoir-faire pour une organisation                              |
| Attente 4 –<br>Variabilité         | Possibilités: 1) <b>paramétrage</b> de composant d'usine, 2) utilisation des <b>patterns</b> avec le mécanisme de <b>substitution</b> pour variabilité de code<br>La notion de modèle de caractéristiques (i.e., feature model) pour la définition de lignes de produits n'est pas encore introduite dans EGF |
| Attente 5 – DSL de<br>génération   | Utilisation de <b>chaînes de génération</b>   |

# Plan

1. **Les usines logicielles**
2. **EGF**
3. **Intégration d'un outil d'usines logicielles dans un processus logiciel**

# Acteurs d'usines logicielles

- **Utilisateur final**
- **Utilisateur d'usines logicielles sur étagère**
- **Réalisateur d'usines logicielles simples**
- **Réalisateur d'usines logicielles complexes**



# Niveaux de maturité d'intégration d'UL

## ➤ Non défini

- ◆ Utilisation d'ULs ponctuellement et sans réelle vision

## ➤ Défini

- ◆ Application d'un plan d'étude et de déploiement d'ULs

## ➤ Maîtrisé

- ◆ Utilisation d'ULs dans une démarche de ligne de produits (niveau générique avec déclinaison spécifique par produit)

## ➤ Optimisé

- ◆ Niveau Maîtrise avec un contrôle sur les optimisations réalisées pour un ensemble de processus logiciel

# Activités spécifiques d'intégration d'ULs

| ACTIVITÉS DE RATIONALISATION DES PROCESSUS DE DÉVELOPPEMENT ET DE MAINTENANCE  | GÉNÉRIQUE | SPÉCIFIQUE |
|--|-----------|------------|
| <i>Évaluer les processus de développement et de maintenance. Dresser la carte des activités avec leur statut d'automatisation.</i>   |           | X          |
| <i>Recueillir les demandes des utilisateurs.</i>   | X         | X          |
| <i>Développer une technologie cœur (frameworks, DSLs, etc., et des usines logicielles).</i>  | X         |            |
| <i>Maîtriser la déclinaison d'une suite de portefeuilles d'usines logicielles. Déterminer comment une suite de portefeuilles peut se décliner pour automatiser un processus existant ou nouveau.</i> | X         |            |
| <i>Implémenter les usines logicielles. Réalisation de portefeuilles d'ULs dans un cadre organisationnel donné.</i>   |           | X          |
| <i>Évaluer l'automatisation d'activité. Évaluation des portefeuilles dans un contexte organisationnel.</i>   |           | X          |

# Conclusion

## Concernant l'enjeu

- ◆ Garder sous contrôle le triplet coût-délais-qualité des processus de développement et de maintenance logiciel alors que la complexité des applications ou systèmes logiciel ne cesse d'augmenter

## Solution présentée

- ◆ Usines logicielles avec la technique générative
- ◆ Outil : EGF pour le développement d'usines logicielles

## Intégration dans un processus logiciel

- ◆ Mener par étape la conduite de la maturité d'un processus outillé au moyen d'usines logicielles

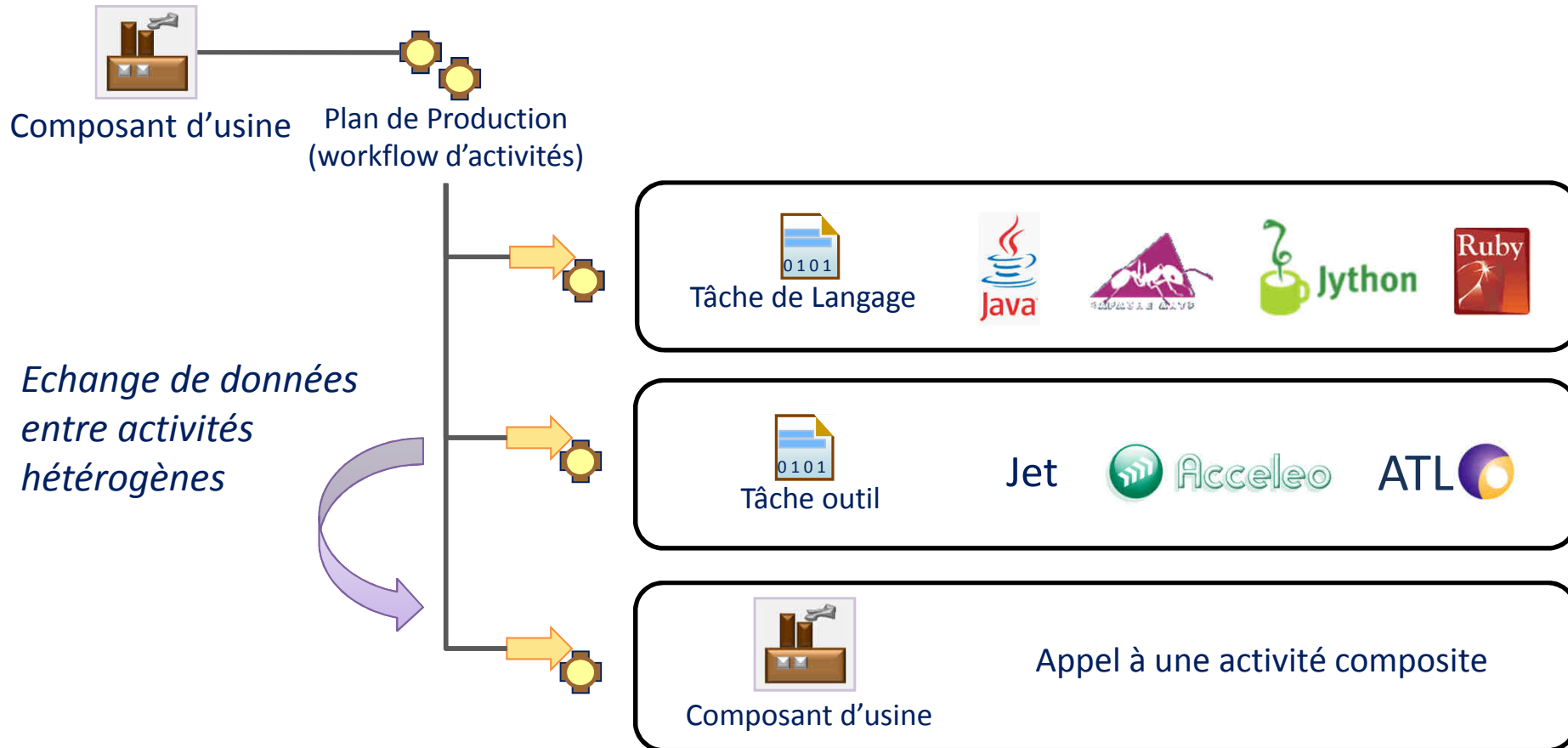
# General info about EGF



Project page: <http://www.eclipse.org/egf>  
Wiki: <http://wiki.eclipse.org/EGF>  
Blog: <http://blanglois.blogspot.com/>  
Twitter: @LangloisBenoit

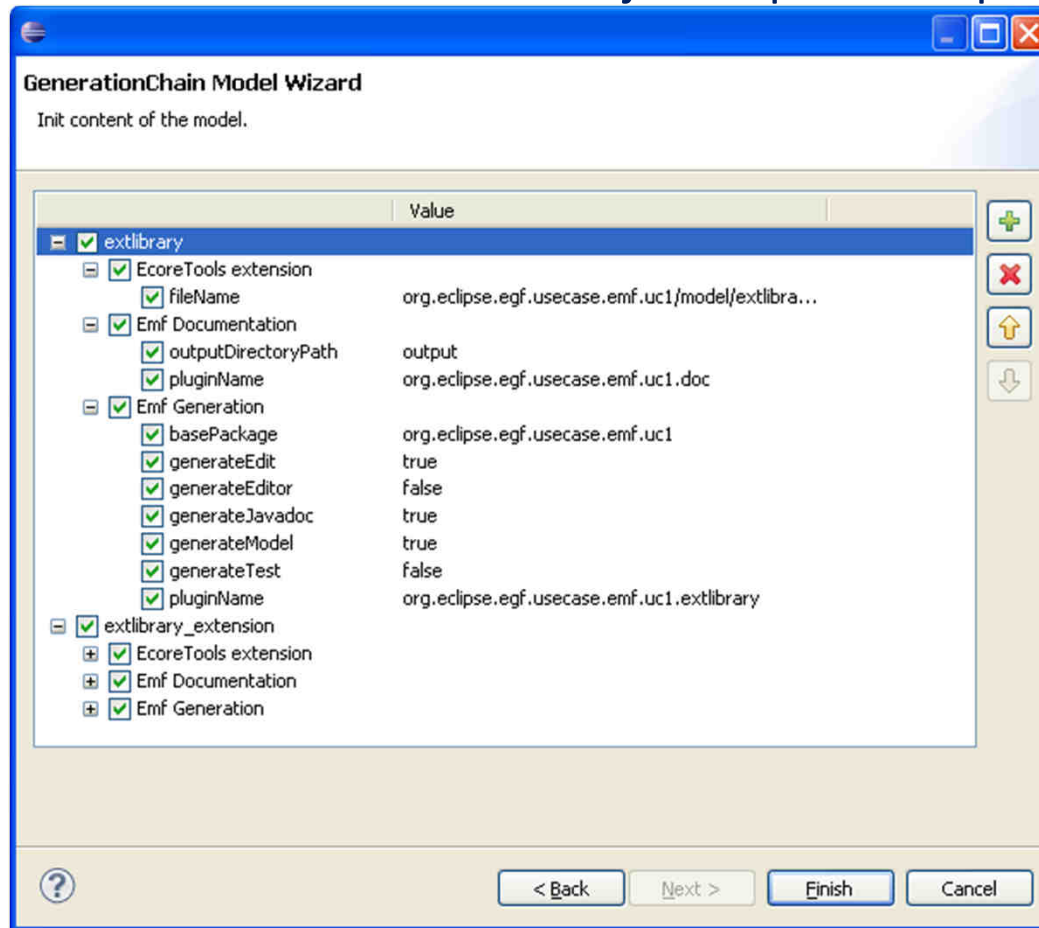
# Annexe

# Composition d'activités EGF



# Chaîne de génération – Application à la génération EMF

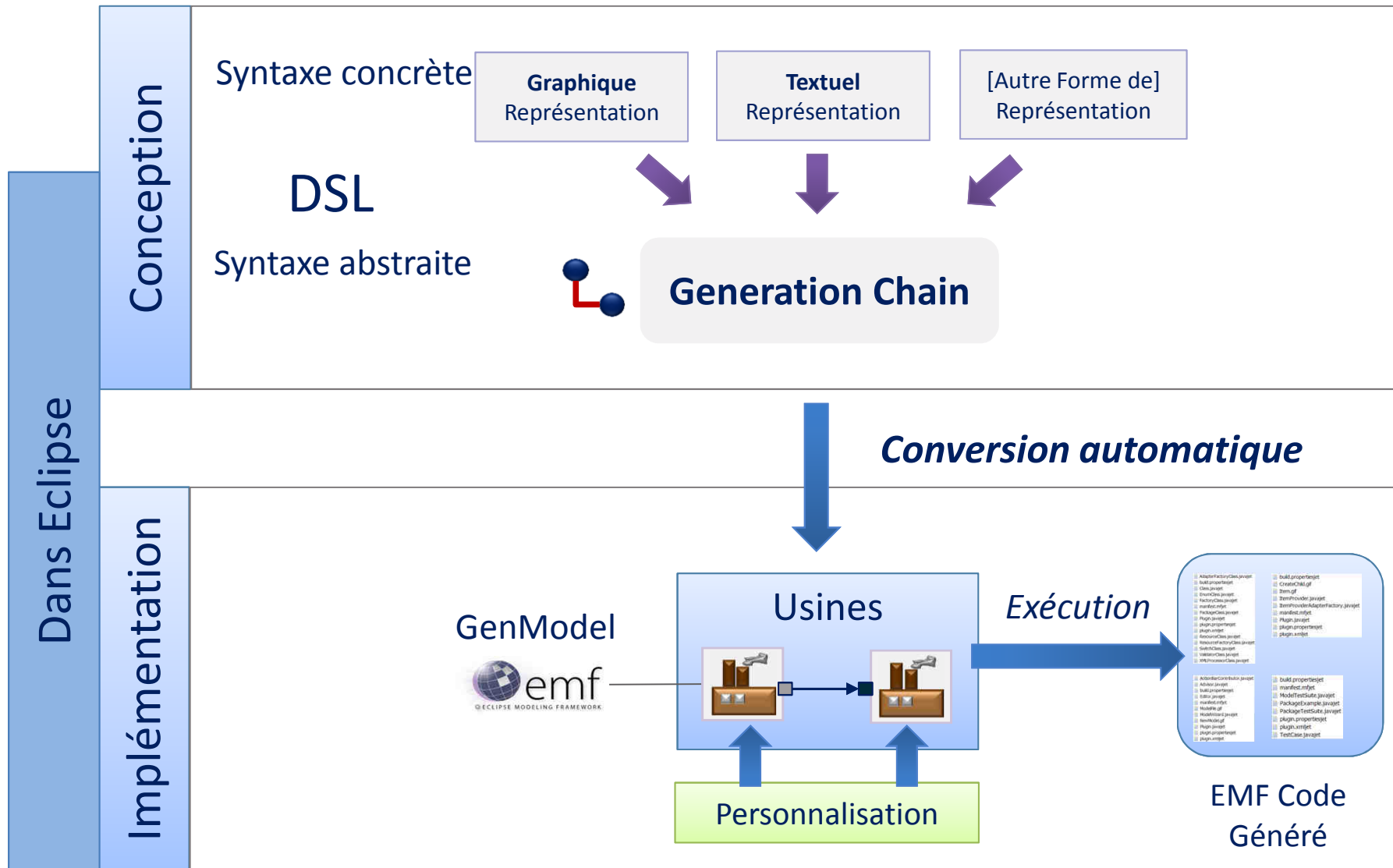
Initialisation des valeurs majeures pour chaque modèle ecore sélectionné



→

Un modèle de chaîne de génération est créé automatiquement par l'assistant

# Chaîne de génération – Génération EMF





# Pattern

