Les outils Eclipse pour améliorer vos développements

JDev Marseille

5 Juillet 2017

Table des matières

Objectifs

Cette présentation contient des informations sur :

- l'IDE Eclipse
- les architectures OSGi, RCP et RAP
- les projets de modeling

Les buts de cette présentation sont :

- vous donner des informations à propos des projets Eclipse
- vous donner des idées accessibles pour améliorer votre usine logicielle
- démarrer votre réflexion sur ces technologies

Presentation

OPCoach

Image 1

- **Formation Eclipse** : RCP, E4, Modeling, Build, en Français, Anglais, Espagnol
- **Expertise Eclipse**
- **Sourcing de profils Eclipse**
- **Web** : [1]
- **Twitter** : **@OPCoach_Eclipse**
- **Certifié ICPF&PSI et inscrit au datadock**

1 - http://www.opcoach.com/en

Eclipse IDE

II

Les différentes versions

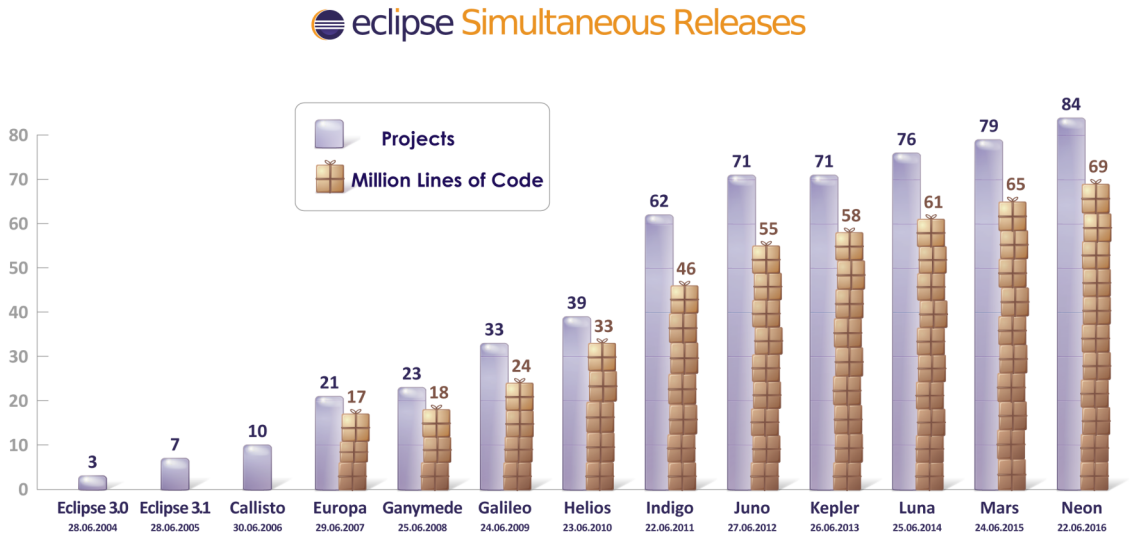


Image 2

Forces

- Environnement riche et outillé (documentation, snippets, ...)
- Portable sur tous les OS
- Open source et gratuit sous licence EPL
- Adopté par tous les fournisseurs de logiciels
- Une communauté active
- Stabilité et régularité des livraisons (1/an)
- Une architecture RCP ouverte basée sur OSGi

Open Services Gateway initiative (OSGi)

Pourquoi OSGi ?

Java semble modulaire

- packages
- classes
- méthodes
- différents jar

Tout est bien séparé lors du développement.

Mais que se passe t il quand on lance un programme java ?

Le cauchemar des jars

- Les jars ne se gèrent pas en version (c'est juste documentaire)
- Les jars ne peuvent pas se lier entre eux
- Il faut connaître toutes les dépendances
- Il faut gérer le **classpath** à la main
- Il faut gérer des : LinkageError, ClassNotFoundException, NoClassDefFoundError
- Deux versions de jar ne peuvent pas cohabiter
- La gestion des jar devient très complexe, c'est l'enfer...

Lancement Java

- La jvm se lance avec **un seul** class loader.
- La modularité s'est dissoute.
- Il faut tout configurer

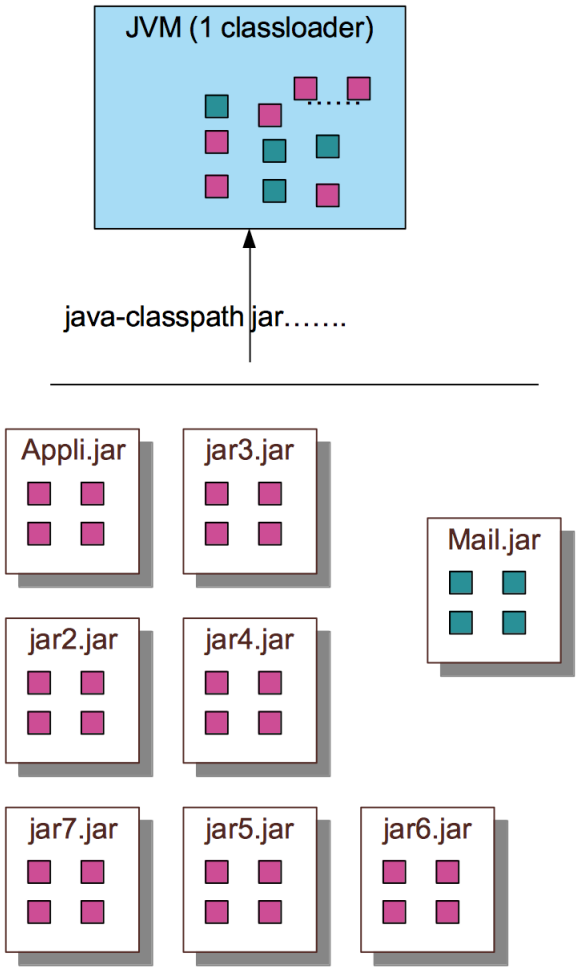
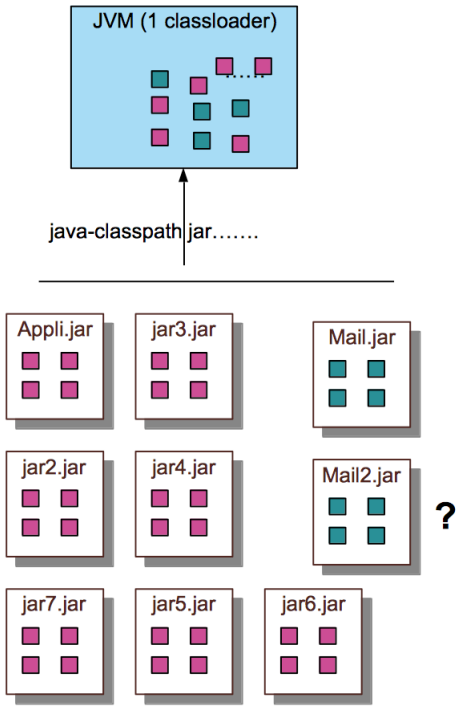


Image 3

Mise à jour de l'application

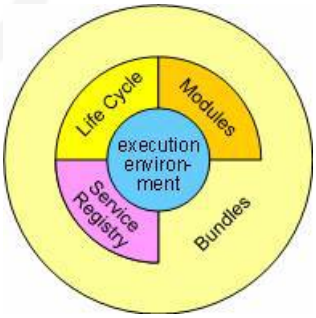
Le mail n'envoie pas les pièces jointes, il faut la V2...
Mais faut il tout mettre à jour ?



OSGi va répondre à ces problèmes

Présentation

- OSGi définit la notion de 'Bundle'
- OSGi gère les versions des bundles
- OSGi gère l'activation des bundles
- OSGi permet de gérer des services (Service Registry)



OSGi Framework

OSGI fournit un modèle de déploiement de modules logiciels java.
OSGi est une spécification gérée par la communauté OSGi Alliance : *http://www.osgi.org*[2]
OSGi a été démarré en 1999 pour gérer du Java embarqué

2 - http://www.osgi.org

Relations avec les autres bundles

- Le bundle peut dépendre d'autres bundles
  - de manière optionnelle
  - selon des versions minimales ou maximales
- Le bundle exporte les packages qu'il veut rendre visible
- **Le classpath se calcule donc automatiquement**

com.opcoach.training.rental

```
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: %pluginName
4 Bundle-SymbolicName: com.opcoach.training.rental;singleton:=true
5 Bundle-Version: 1.0.1
6 Bundle-ClassPath: .
7 Bundle-Vendor: OPCoach
8 Bundle-Localization: plugin
9 Bundle-RequiredExecutionEnvironment: J2SE-1.5
10 Export-Package: com.opcoach.training.rental,
11  com.opcoach.training.rental.impl,
12  com.opcoach.training.rental.util
13 Require-Bundle: org.eclipse.core.runtime,
14  org.eclipse.emf.ecore;visibility:=reexport,
15  org.eclipse.swt;bundle-version="3.5.1";visibility:=reexport
16 Bundle-ActivationPolicy: lazy
```

visibility for the others

others used bundles

Image 4 Relations of a bundle

Comportement dynamique

Le cycle de vie du bundle est géré par le moteur OSGi :

- Il peut être démarré quand cela est nécessaire (lazy loading)
- Il peut être arrêté à tout moment
- Il peut être remplacé par une version plus récente
- Deux versions différentes peuvent cohabiter
- Des nouveaux services peuvent être publiés à chaud

Moteur OSGi

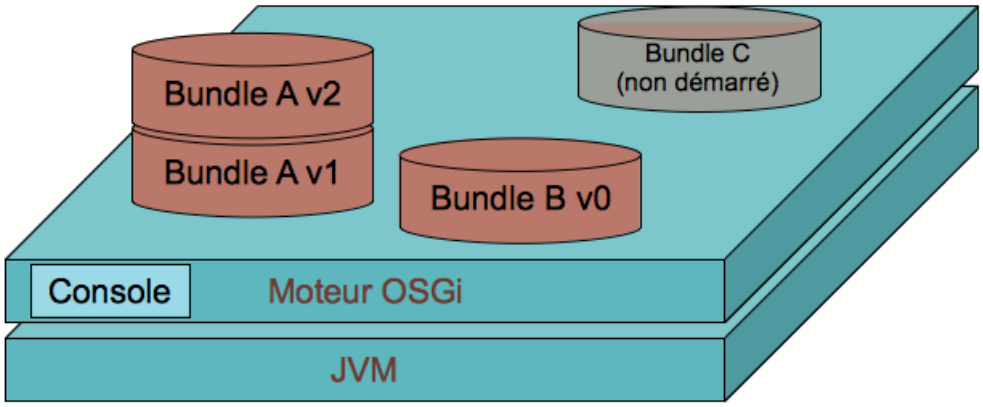


Image 5 OSGi engine

Exemple d'architecture OSGi

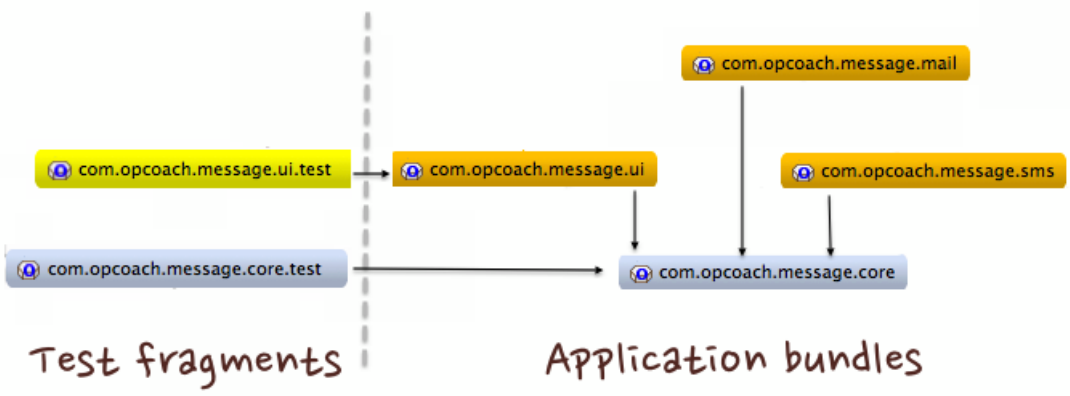


Image 6 OSGi architecture

Eclipse Architecture

IV

Eclipse 3

- L'architecture Eclipse est basée sur la notion de plug-in
- Un plug-in est un composant logiciel élémentaire similaire à un bundle OSGi

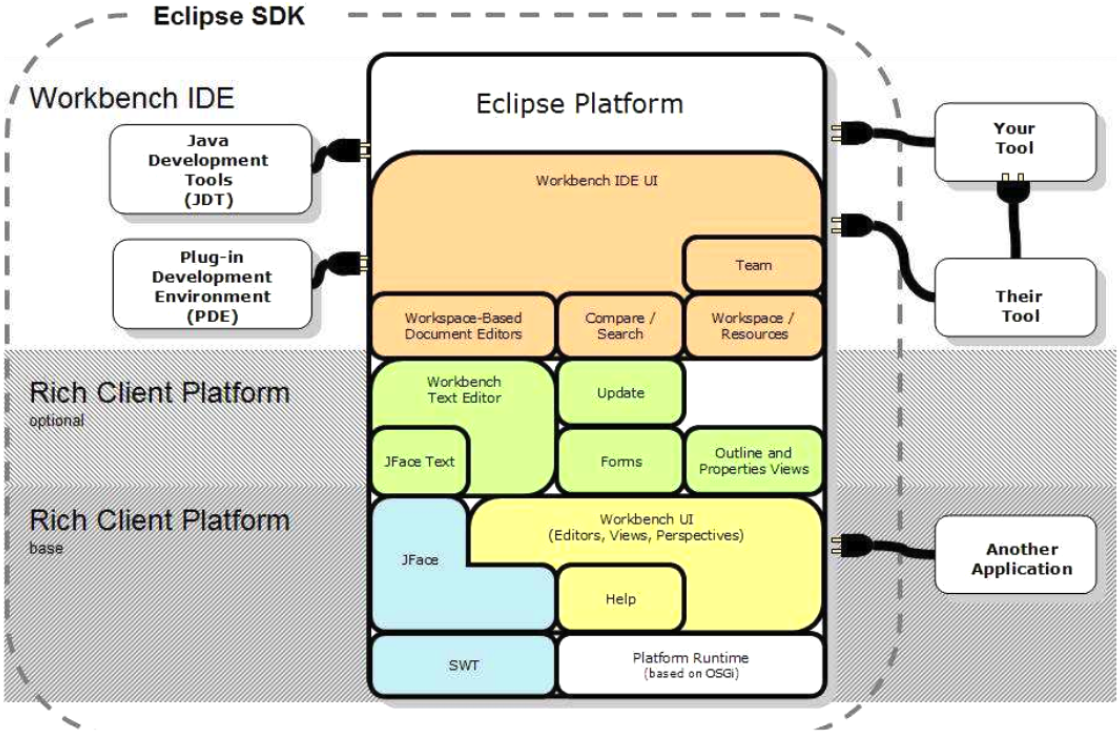


Image 7 Eclipse 3 Architecture

Plug-in vs Bundle

Un **plug-in** définit deux concepts supplémentaires :

- **Le point d'extension** : décrit un concept que le plug-in peut prendre en charge (menu, driver, concept métier, ...)
- **Extension** : définit une instance de ce concept (le driver X, le driver Z, etc..)

Architecture Eclipse 4

- En 2012, la fondation Eclipse a livré un nouveau runtime basé sur l'architecture Eclipse 4
- Cette architecture reste compatible avec la plupart des concepts Eclipse 3
  - Une majorité d'Eclipse reste écrit en version 3
- Eclipse 4 propose :
  - un modèle d'application (défini avec EMF)
  - un renderer d'IHM (SWT ou Java Fx)

- un mécanisme d'injection moderne (avec re-invocation automatique)
- une gestion de CSS
- des 'spies'
- **une couche de compatibilité** pour utiliser Eclipse 3

- Cette nouvelle architecture reste basée sur OSGi et est délivrée sous licence EPL

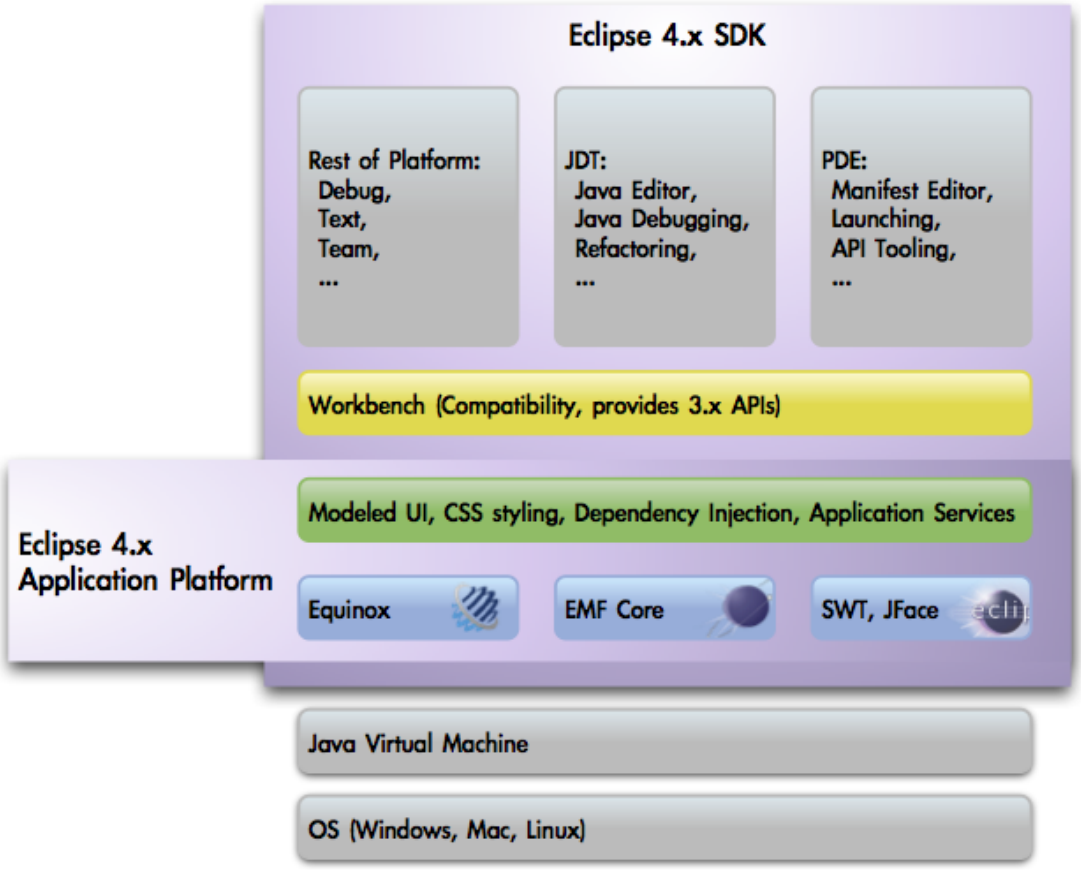


Image 8 e4 Architecture

Architecture RCP/RAP

- **RAP** = Remote Application Platform
- RAP est utilisé pour déporter une application dans un navigateur
- RAP utilise une implémentation SWT spécifique
- RAP utilise JSon pour communiquer entre le client et le serveur

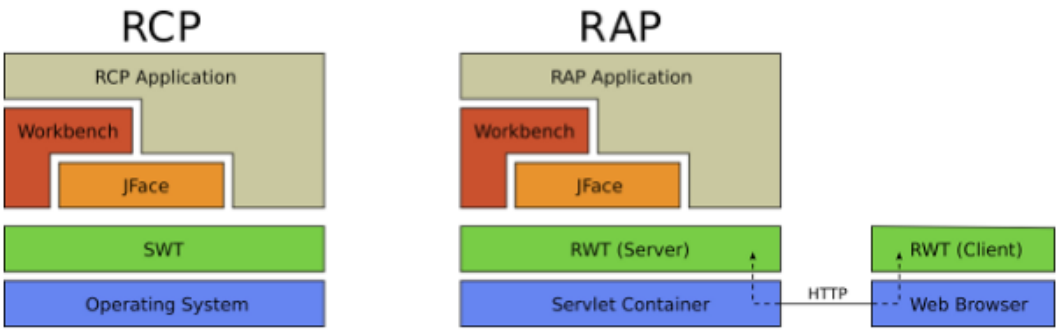


Image 9 RCP and RAP

- RAP permet d'écrire à la fois votre application en mode standalone ou web en single sourcing

Vue d'ensemble des projets de Modeling

V

Projets de Modeling

La puissance des modèles pour votre productivité

Big picture

Les projets de modeling sont basés sur l'architecture RCP

Ils peuvent être utilisés pour améliorer fortement votre productivité logicielle en termes de :

- développement
- d'évolutions
- de maintenance

Les projets de modeling peuvent être utilisés selon deux modes :

- **MDA**: Model Driven Application:
  - votre modèle est utilisé au runtime de l'application
- **MDD**: Model Driven Development:
  - votre modèle est utilisé pour votre outillage

Cartographie

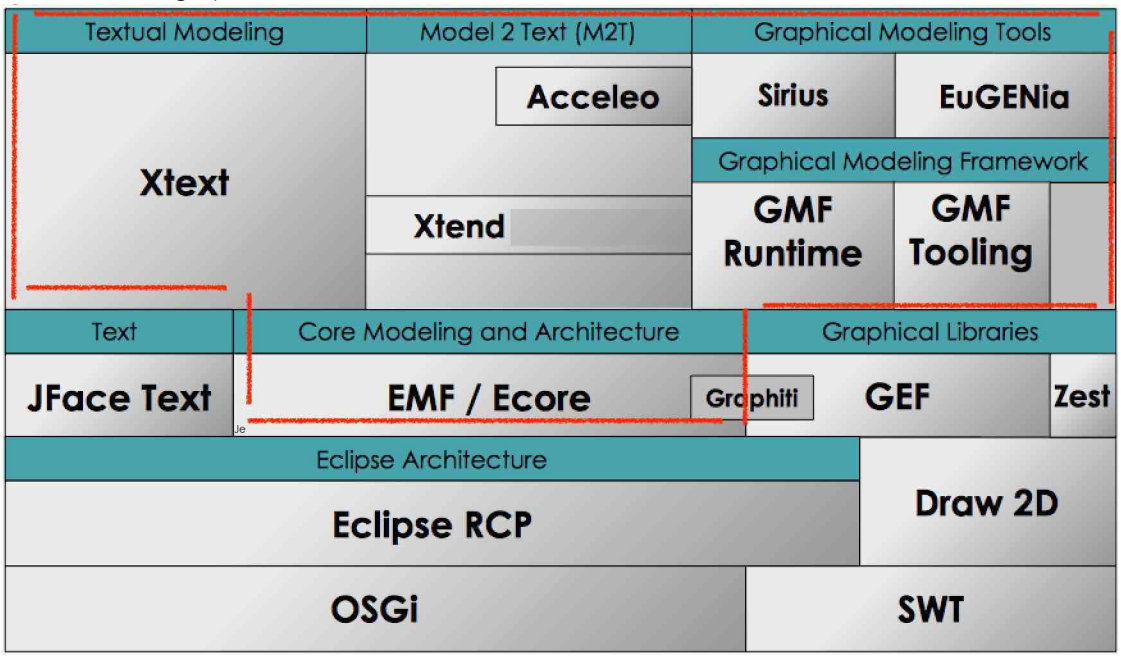


Image 10 Vue d'ensemble des projets Modeling

EMF?

- EMF est le noyau des projets de modelling
- Il définit
  - le language Ecore de modélisation
  - des générateurs par défaut (bean, editor..)

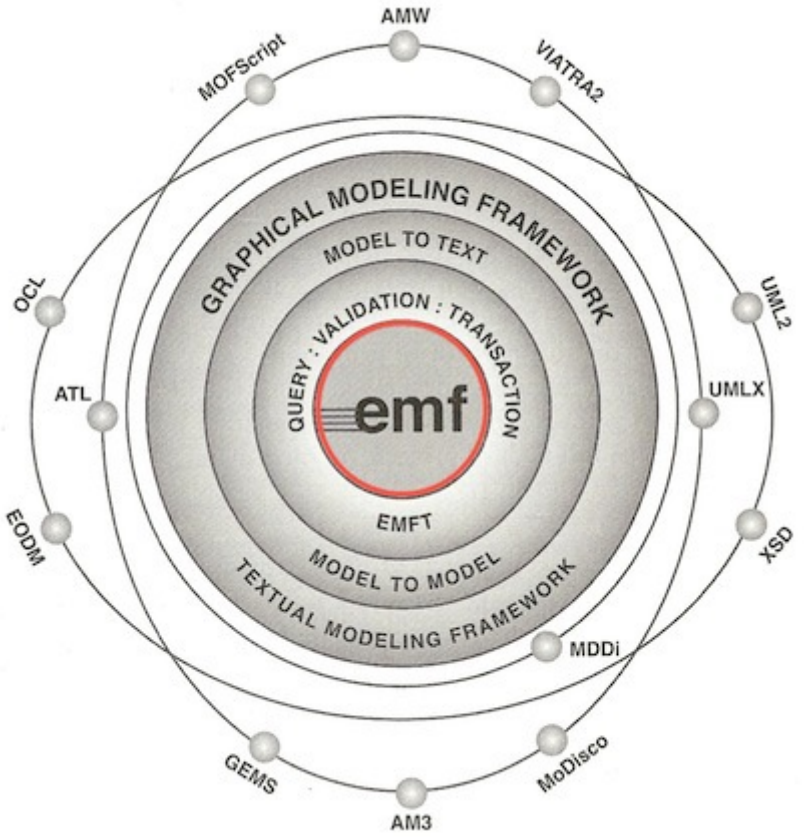


Image 11 EMF in the Modeling project

Etapes d'utilisation

EMF va permettre de définir un **DSL** (Domain Specific Language)

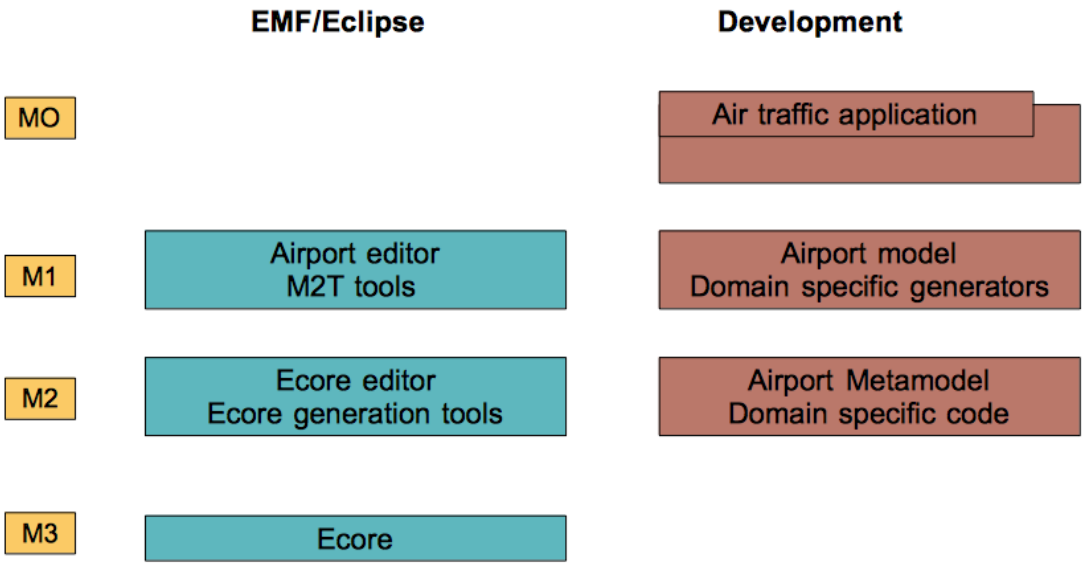


Image 12 EMF in the development process

Editeur arborescent

- L'éditeur arborescent est l'éditeur par défaut
- Les propriétés du modèle Ecore sont saisies dans la vue Properties d'Eclipse

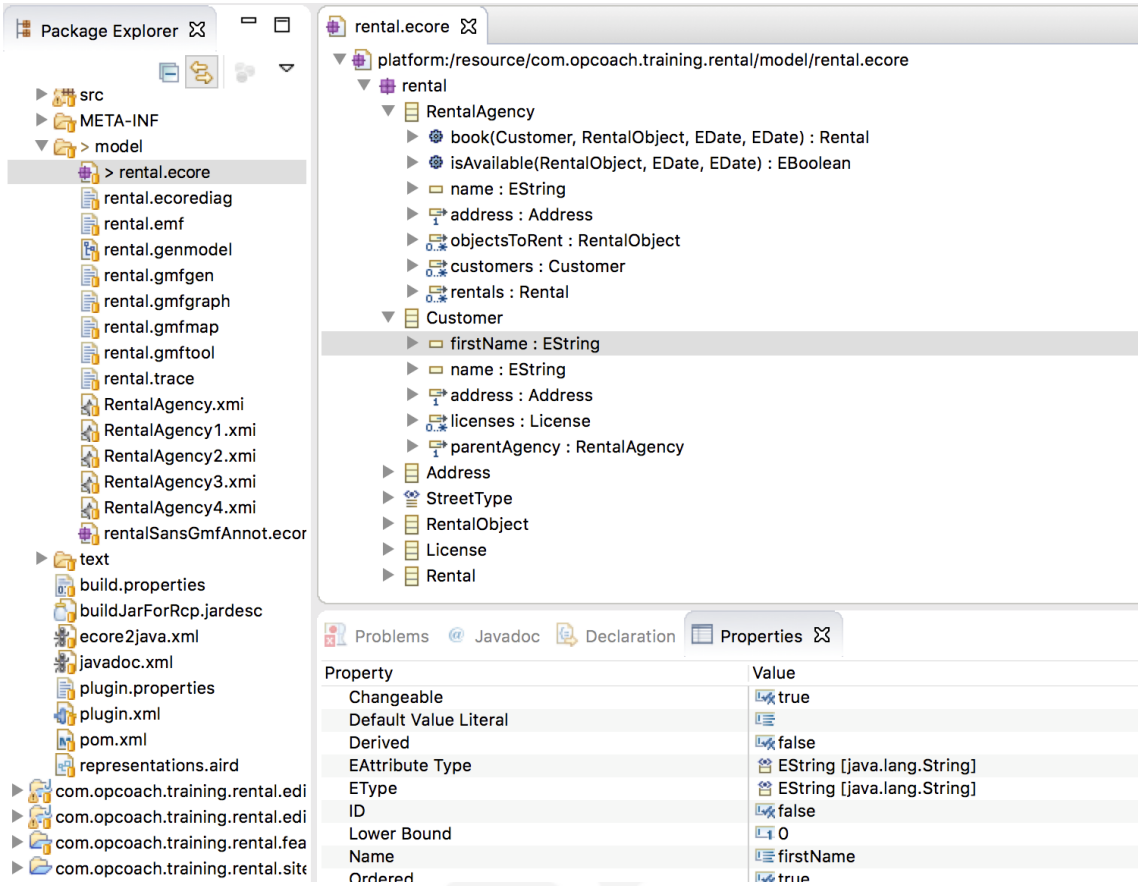


Image 13 Ecore Tree Editor

Editeur graphique

- L'éditeur graphique est aussi disponible dans la livraison Eclipse Modeling :

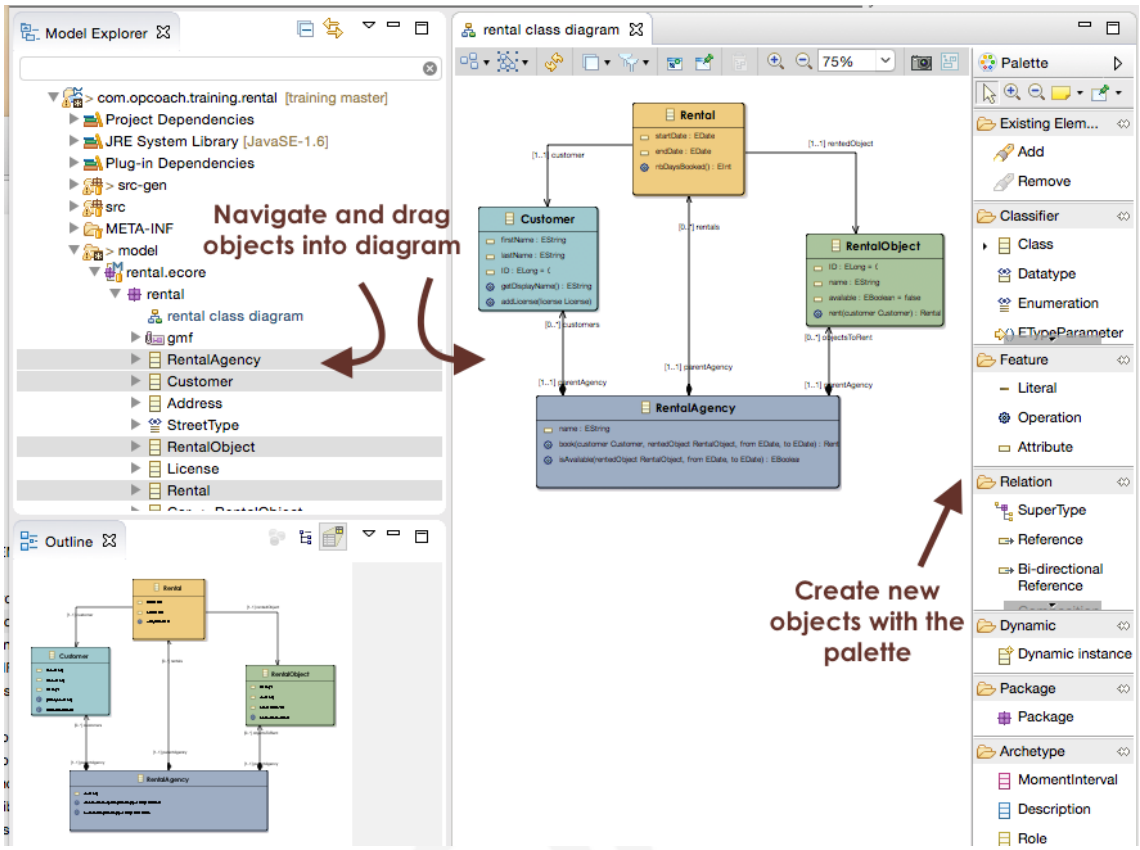
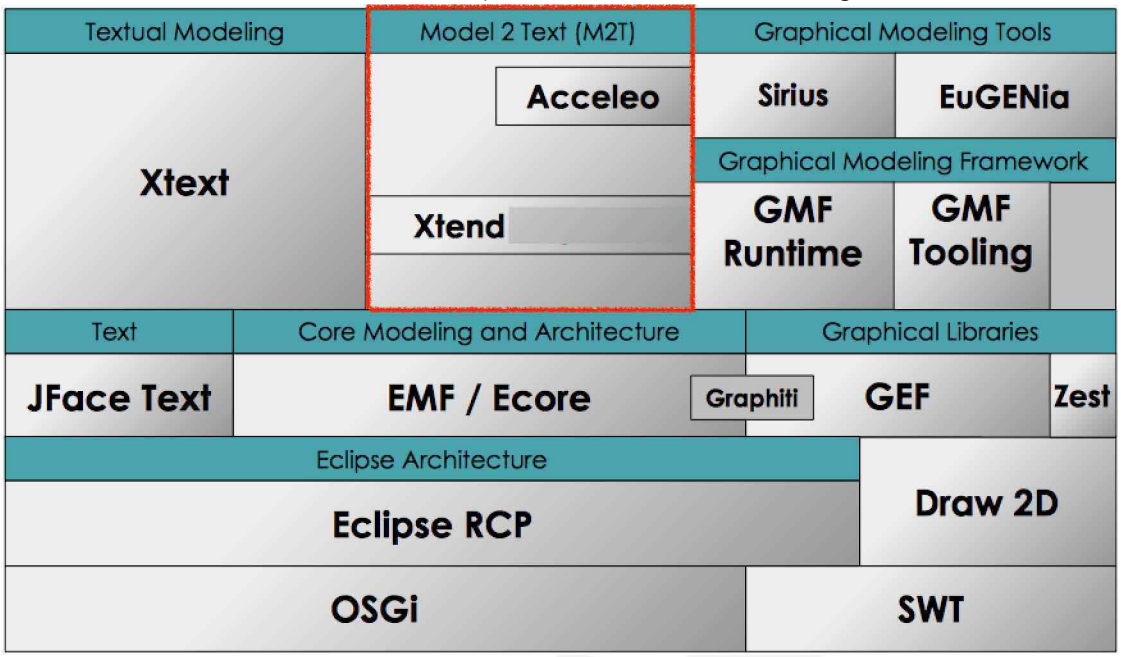


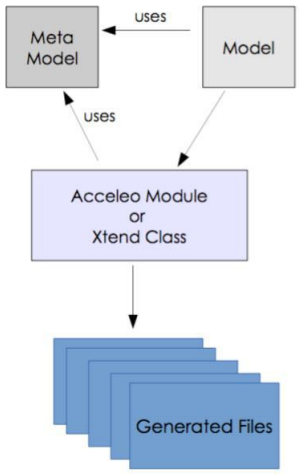
Image 14 Ecore tools

Model to Text (M2T)

Comment utiliser les technologies de génération de code pour augmenter votre productivité ?
Une fois votre meta-modèle définit, on peut créer très facilement des générateurs



Principe M2T



Le langage Acceleo

- Acceleo est un outil de génération M2T développé par Obeo
- Implémente la norme MTL (Model Text Language) de l'OMG
- Permet de générer du texte à partir d'un modèle.
- Respecte la syntaxe OCL pour les accès aux modèles
- http://www.acceleo.org

Intégration Eclipse

Acceleo est parfaitement intégré à Eclipse :

- editeur complet (coloration, completion, detection erreurs, quick outline (Ctrl O), code folding, search references (Ctrl Shift G)...)
- debugger
- perspective
- wizards
- vues
- launch configuration
- commandes d'aide à la génération de templates

Exemple Acceleo

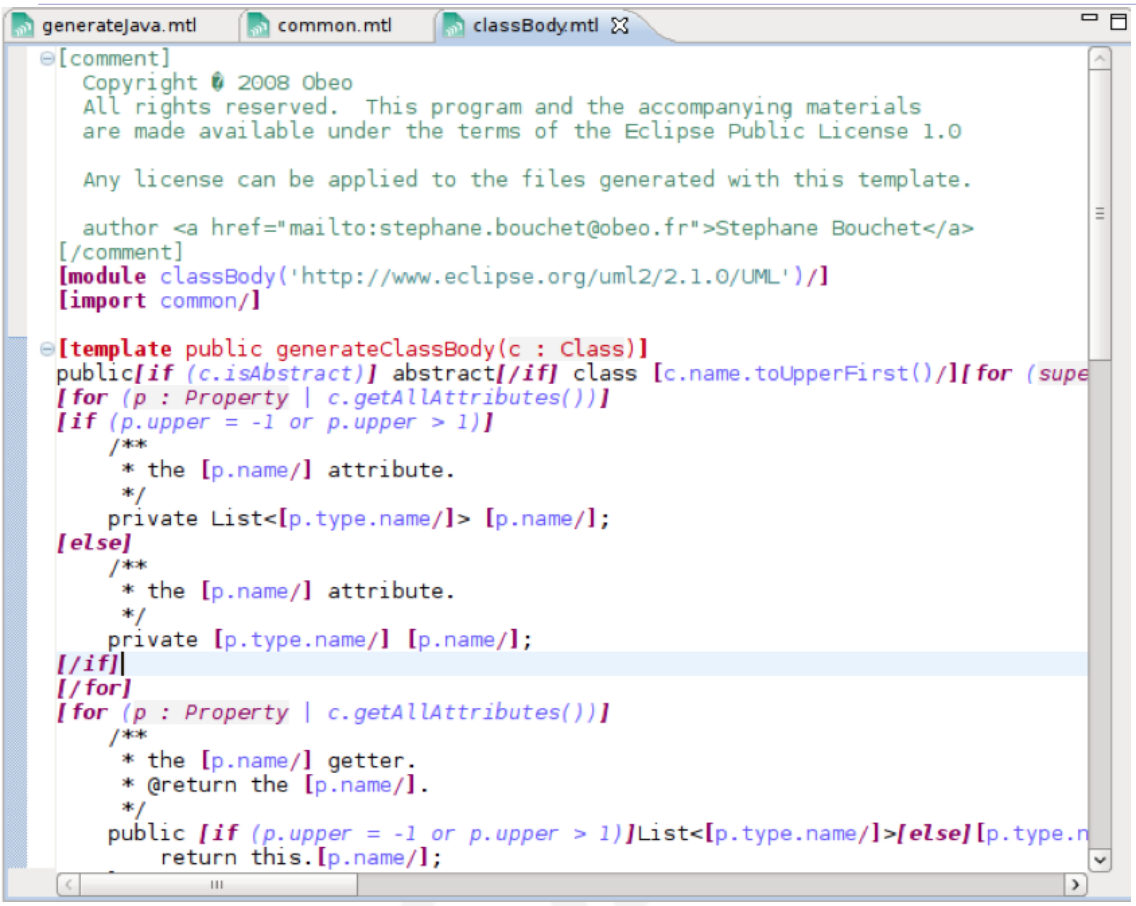


Image 15 Acceleo sample

Le language Xtend

- Xtend est un langage qui simplifie Java et qui se regénère en Java (no byte code)
- Il possède une syntaxe permettant de générer du texte ou du code
- C'est un langage très ludique et productif à apprendre

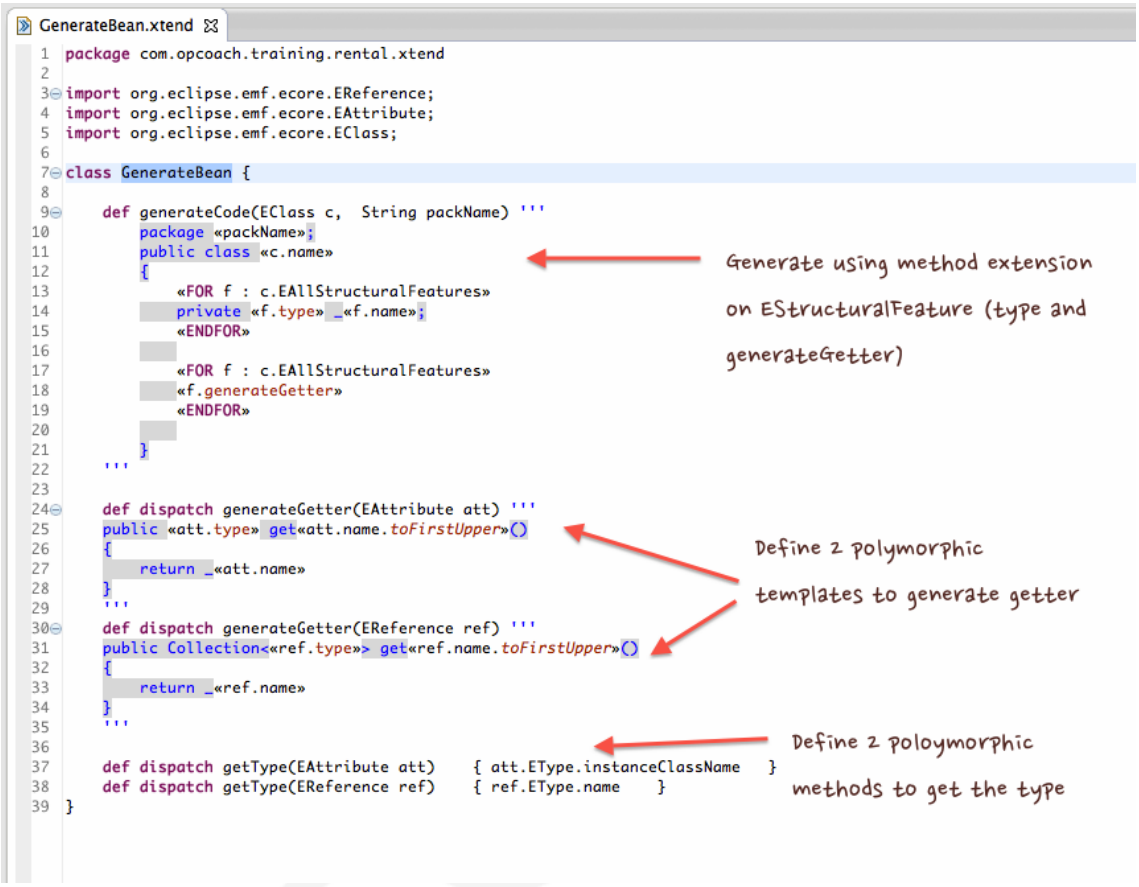


Image 16 Xtend generator sample

Comment l'utiliser ?

Créer un meta-modèle de votre métier :

- description de satellite
- description de system
- description de XXX

Créer une instance de modèle de votre métier :

- le satellite ABC
- le system XYZ

Ecrire le(s) générateur(s) de code associés:

- compatible avec le couple modele/meta-modèle
- pour produire le code de 'plomberie' pénible à écrire :
  - mapping SQL
  - mapping WEB (angular, node.js, php...)
  - mapping vers votre framework métier

Remarque : quand une information est décrite plusieurs fois à plusieurs endroits, pensez à la génération à partir d'un modèle !

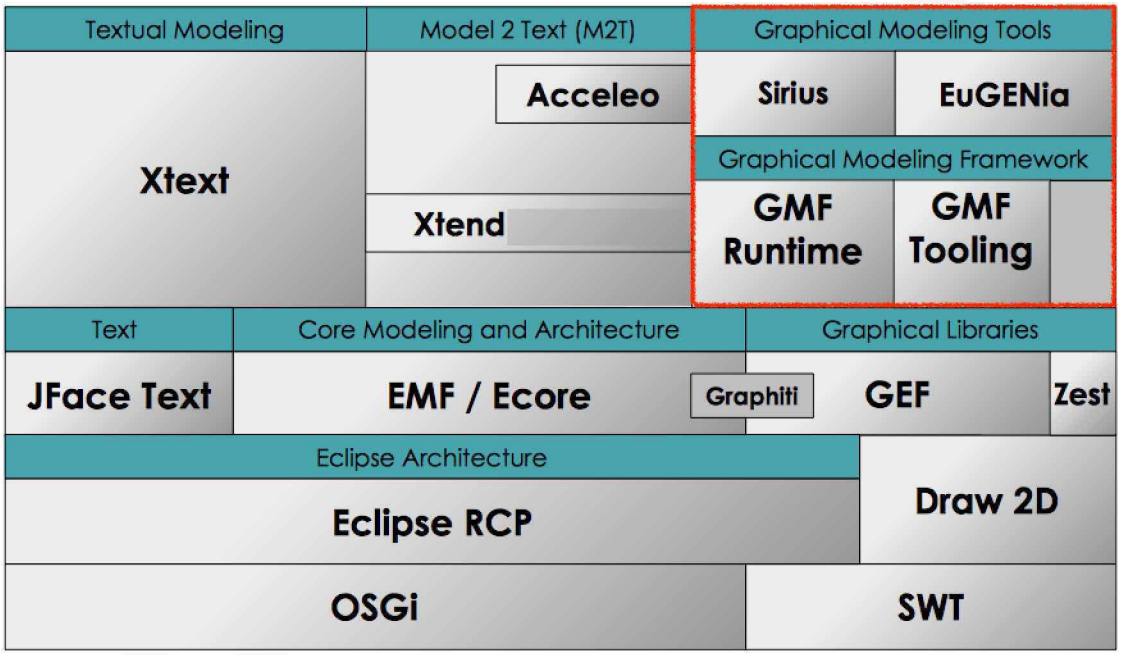
Edition du modèle

On peut aussi définir son propre éditeur en utilisant :

- Sirius pour les éditeurs graphiques
- XText pour les éditeurs textuels

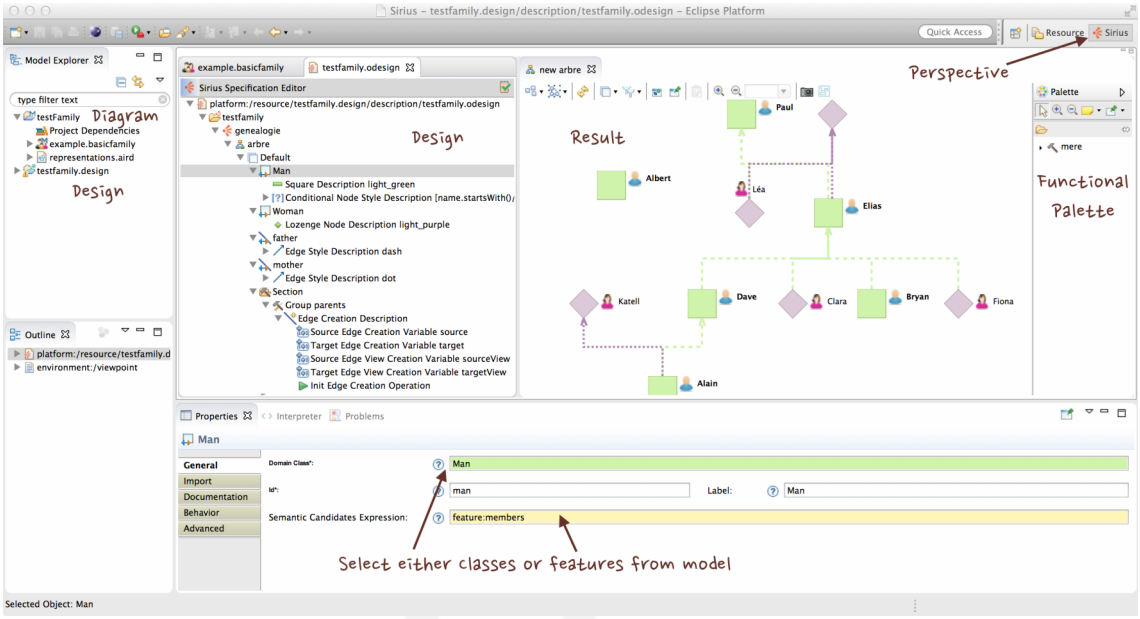
Editeurs graphiques de modèles

- Utilisez les éditeurs graphiques de modèle quand ils sont appropriés
- Intégrez vos éditeurs graphiques en tant que plugin dans vos applications



Sirius

- Sirius est un framework permettant de décrire graphiquement un modèle sous différentes formes
  - **diagram** : noeuds et liens
  - **table** : instances du modèle en ligne et propriétés en colonne
  - **matrice** : lignes et colonnes pour les instances et valeurs dans les cellules
- Il permet de décrire très rapidement son éditeur graphique
- Permet d'associer un élément du modèle à sa représentation graphique (node, lien, cellule...)
- L'éditeur peut être utilisé directement **sans génération de code**

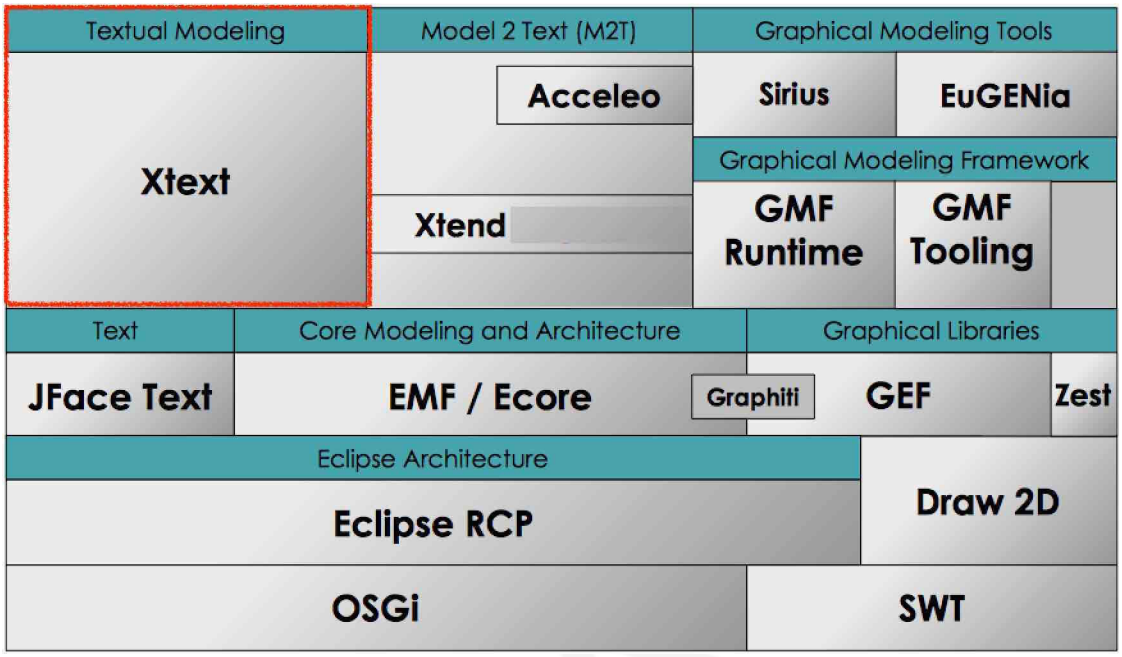


Sirius overview

Remarque : l'éditeur graphique du langage Ecore est décrit avec Sirius

Editeurs textuels de modèles

- Utilisez les éditeurs textuels quand cela est approprié
- Reliez vos générateurs de code (MDD)
- Intégrez ces éditeurs dans vos livraisons client (MDA)



Xtext

- Projet Eclipse permettant de définir un éditeur textuel de modèle
- 'Language IDE framework'
- Génère un éditeur à partir d'une grammaire et d'un modèle Ecore
- Permet d'associer un 'token' textuel à un élement du modèle (classe, champ...)

Xtext website: *http://www.eclipse.org/Xtext/* [3]

Comment l'utiliser ?

- Suivre le tutorial de 5 mn ou de 30 mn
- Xtext sait aussi générer un éditeur compatible avec IntelliJ
- En MDD, on peut facilement relier l'éditeur à un générateur de code
  - décrivez une fois et générez dans différents langages
- Les éditeurs XText sont générés

Caractéristiques des éditeurs XText

Tous les éditeurs obtenus avec XText ont les propriétés suivantes :

- syntax highlighting
- auto completion
- index de recherche optimisés
- validation et quickfixes

3 - http://www.eclipse.org/Xtext/

- gestion des erreurs et warnings
- templates
- outline
- definition d'hyperlink (cross languages)
- relation optionnelle avec un générateur de code (XTend ou Acceleo)

Visualisation graphique de la grammaire

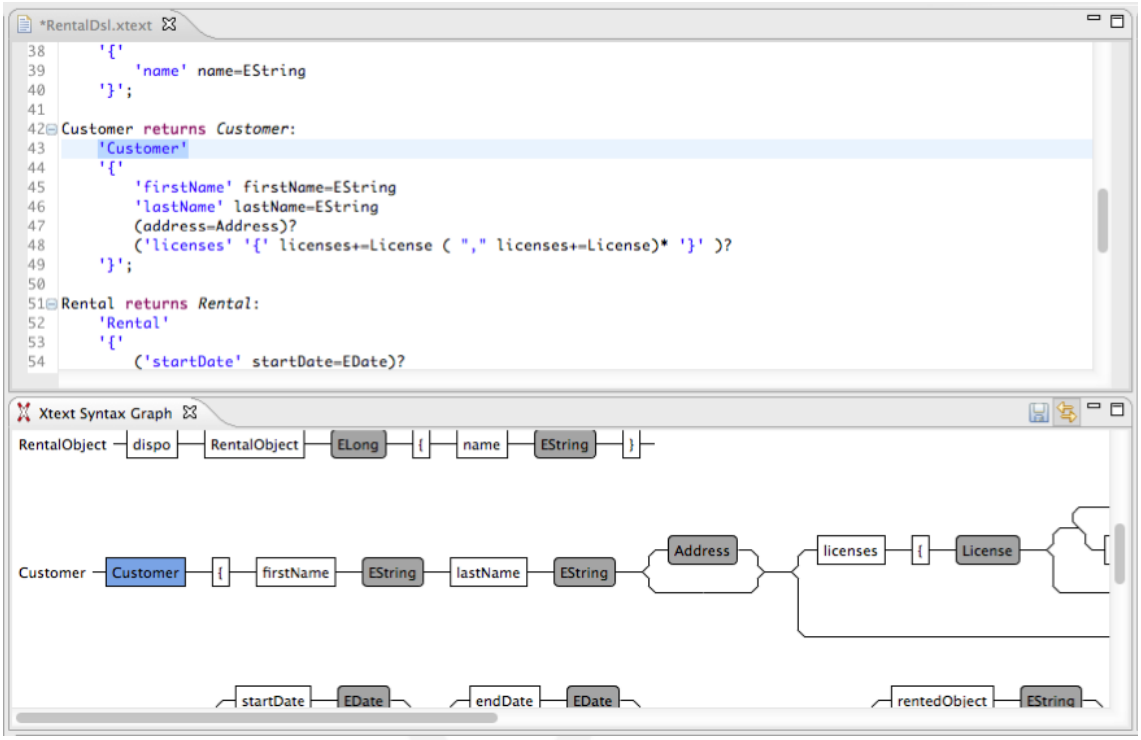


Image 17 XText Syntax Graph

Editeur généré

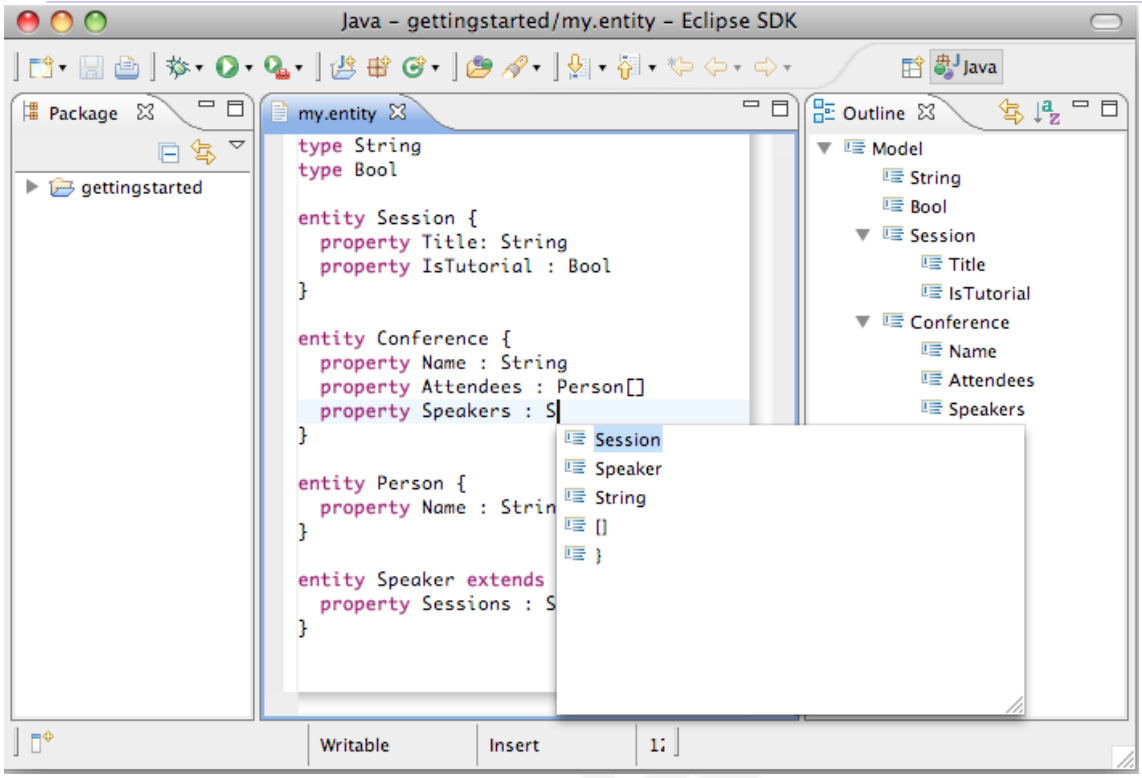


Image 18 Sample editor

Paramétrer votre éditeur

Avec XText il est très simple et ludique de rajouter :

- quickfixes (**@Fix** annotation)
- erreurs et warnings (**@Check** annotation)
- auto completion (textuelle ou sémantique)
- formatage de texte
- générateur de code (déclenché sur la sauvegarde)
- templates (compatibles avec votre syntaxe)

Conclusion

Questions ?