



Presentation XText

23 Juin 2011

Eclipse Party Toulouse

OPC 11 ECP PRE XTX 01 A

Présentation



OPCoach

- SARL créée en juin 2009 <http://www.opcoach.com>
- Membre de la fondation Eclipse (Solution Member)
- Olivier Prouvost olivier.prouvost@opcoach.com¹
 - Co fondateur Anyware Technologies
 - Expert XML, Java / Eclipse
- Objet
 - Formation, Expertise et Audit Eclipse

OLIVIER PROUVOST
CONSULTANT SENIOR ECLIPSE / OPEN SOURCE

olivier.prouvost@opcoach.com
www.opcoach.com

MOBILE : +33 (0)6 28 07 65 64

25 RUE BERNADETTE
31 100 TOULOUSE (FRANCE)



Références

- Grands comptes



...

1 - <mailto:olivier.prouvost@opcoach.com>

Présentation XText



Xtext

- Nouveau projet permettant de générer un éditeur textuel
- 'Language IDE framework'
- Génère l'éditeur à partir d'une grammaire et d'un modèle.ecore
- Fournit des outils d'édition et de génération
- S'interface avec l'édition java
- Permet de générer le modèle Ecore à partir de la grammaire
- Plus de 100 pages de documentation détaillée

Définition de la grammaire

```
grammar org.eclipse.xtext.example.Domainmodel
  with org.eclipse.xtext.common.Terminals

generate domainmodel "http://www.eclipse.org/xtext/example/Domainmodel"

DomainModel :
  (elements+=Type)*;

Type:
  DataType | Entity;

DataType:
  'datatype' name=ID;

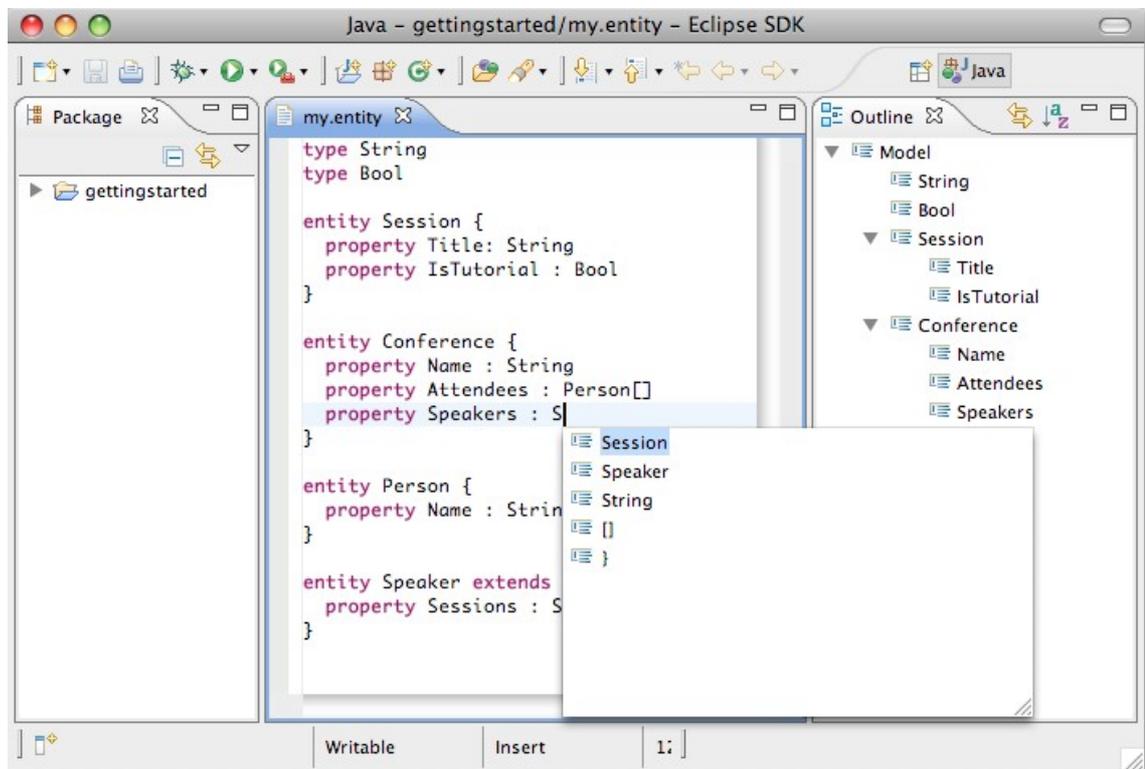
Entity:
  'entity' name=ID ('extends' superType=[Entity])? '{'
    (features+=Feature)*
  '}';

Feature:
  name=ID ':' type=TypeRef;

TypeRef:
  referenced=[Type] (multi?='*')?;
```

Grammaire

Editeur générée



Exemple d'éditeur

Caractéristiques des éditeurs

Tous les éditeurs possèdent les caractéristiques suivantes :

- syntax highlighting
- auto completion
- index pour la recherche
- validation et quickfixes
- gestion des erreurs et warnings
- gestion des templates
- gestion de l'outline
- définition des hyperlink (cross languages)

Génération Xtext

XText utilise :

- Xtend 2(ex Xpand) pour la génération
- MWE2 : Model Workflow Engine

Des templates de génération sont fournis par défaut.

Principe d'utilisation

- Créer un projet de nature xtext
- Définir un modèle ecore ou directement une grammaire

- Exécuter le générateur en exécutant le workflow
- Lancer les plugins générés.

Utilisation Avancée

La customisation de l'éditeur généré se fait à l'aide de fragments de génération
Chaque fragment s'incorpore dans le workflow et fournit des classes spécifiques

Exemples de fragments :

- **EcoreGenerator** : pour générer le.ecore à partir de la grammaire
- **XTextAntlrGenerator** : mis par défaut pour avoir le parser et le lexer
- **Formatter** : pour ajouter un formater de texte
- **LabelProvider** : pour fournir les labels providers dans les selections
- **OutlineNodeAdapterFactory** : pour gérer l'outline
- **SimpleProjectWizard** : pour générer un wizard de new project

De OAW à XText

- Au départ le projet OAW proposait 4 'langages' et éditeurs associés :
 - XTend
 - Xpand
 - Check
 - Workflow (version 1) -> XML
- OAW s'intègre ensuite dans le projet Modeling/M2T et se renomme XPand
- Les outils de workflow sont séparés dans un composant commun (MWE)
- Le workflow version 2 apparait avec un langage dédié et un éditeur
- XText devient un projet indépendant qui utilise XPand pour générer le code
- Dans Indigo, XPand se renomme XTend2 et XText devient totalement operationnel.