**République du Cameroun**

Republic of Cameroon

*Peace – Work - Fatherland*

**Université de Dschang**

University of Dschang

*Scholae Thesaurus DschangensisIbiCordum*

BP 96, Dschang (Cameroun) –

Tél. /Fax (237) 233 45 13 81

Website:<http://www.univ-dschang.org>.

E-mail : [udsrectorat@univ-dschang.org](mailto:udsrectorat@univ-dschang.org)

**Faculté des Sciences** Faculty of Sciences

**Département de Mathématiques et Informatique**

*Department of Mathematics and Computer Science*

BP 67, Dschang (Cameroun) Tél./Fax (237) …E-mail : [faculte.sciences@univ-dschang.org](mailto:faculte.sciences@univ-dschang.org)



Code/Intitulé de la Matière : **INF 428–> Compilation**

**DSL** (**Domain-Specific Language** )

**Rédigé et présenté par** :

* TCHOUFA NKOUATCHET JOEL
* NKEUGA NGUELIEKAM STEPHANE
* DJAMBOU GEORGETTE MICHELE

**Option** : Réseaux et Services Distribué

**Classe LMD** : Master 1 (INF4)

Sous la supervision de : **Dr CHOUPE**

**Année Académique : 2018-2019**

**Table des matières**

**Introduction**

1. **Xtext ? À quoi ça peut servir ?**
2. **Téléchargement et installation de Xtext.**
3. **Création d’un projet Xtext.**

**VI. Exécution et test de saisi d’un programme**

**Conclusion**

**Introduction**

Dans la programmation informatique ou statistique, les informatiques sont souvent confrontés à des nombreux problèmes. Soit celle lié à la représentation des objets, soit celle lié à intégration de certain outil de compilation. D’où la création du **DSL**, **DSL** est sigle de Domain**-Specific Language** (en anglais), c’est-à-dire Langage Spécifique au Domaine. Il s'agit généralement d'un « petit » langage de programmation dédié à un domaine d'activité spécifique. Il est conçu pour apporter des solutions aux problèmes soulevés dans un domaine particulier et donc est ainsi difficilement réutilisable dans d'autres domaines. C'est pourquoi ce type de langage de programmation se distingue des GPL(**General Purpose Language)** tels que Java, C, C++... dont le but de ces derniers est de proposer des solutions à tout type de problème solvable par ordinateur. La question a la quel nous allons répondre tout a long de notre devoir est de savoir créer notre propre DSL.

1. **Xtext ? À quoi ça peut servir ?**

Xtext est un Framework développé sur Eclipse et permettant d'implémenter votre propre **DSL** textuel ou même de mettre sur pied un langage de programmation générique (un GPL). Il fonctionne sur une JVM (Machine Virtuelle Java) et est constitué de plusieurs API qui vous permettent de décrire les différents aspects du langage que vous voulez créer et propose une implémentation complète de ce langage. Notons que le langage développé sera alors un langage qui s'appuie et qui surcouche le langage Java. C'est-à-dire que les objets offerts par ce Framework sont complètement des objets Java et le développement du compilateur de votre langage sera alors écrit en Java. Xtext peut servir entre autres à proposer des solutions dans les domaines tels que : les systèmes embarqués, l'automobile, les appareils mobiles, l'automatique, les jeux vidéo, etc. Notons aussi que les langages développés via Xtext peuvent selon son tutoriel être définis non seulement avec le langage Java, mais aussi d'autres langages tels que C, C++, C# objective C, Python et Ruby. De plus, une fois que le DSL est mis en œuvre, son utilisation devient indépendante de la JVM.

1. **Téléchargement et installation de Xtext.**

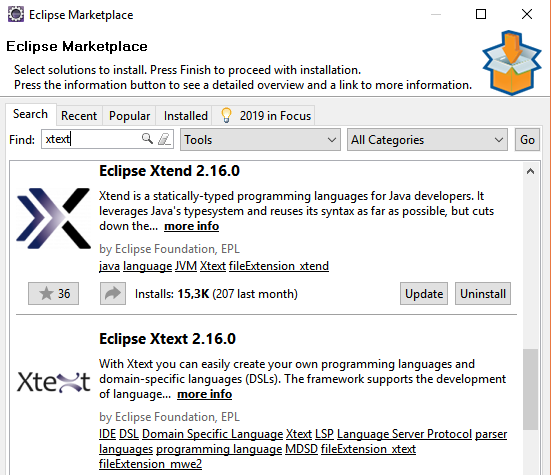
L'installation d'Eclipse Xtext est simple. Il suffit de télécharger le fichier **zip** sur le site itemis des développeurs d’Xtext à l'adresse http suivante : **http://xtext.itemis.com/xtext/language=en/36553/downloads.** Notons qu'il existe plusieurs versions correspondant à différents systèmes d'exploitation. Téléchargez donc la version qui correspond à votre cas. Lorsque le téléchargement est terminé, il suffit de copier le fichier zip sur votre disque dur à l'endroit où vous désirez. Après l'avoir dézippé, dans le répertoire qui se crée, on constate la présence de plusieurs dossiers et fichiers parmi lesquels **eclipse.exe.** On double clic sur le fichier et en suit sur suivant.

1. **Création d’un projet Xtext.**

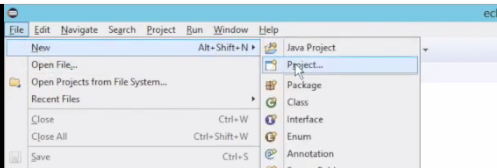
Apres l’ouverture d’éclipse, dans la barre de menu qui souffre il faut cliquer sur **help-- >éclipse** **Marketplace**.



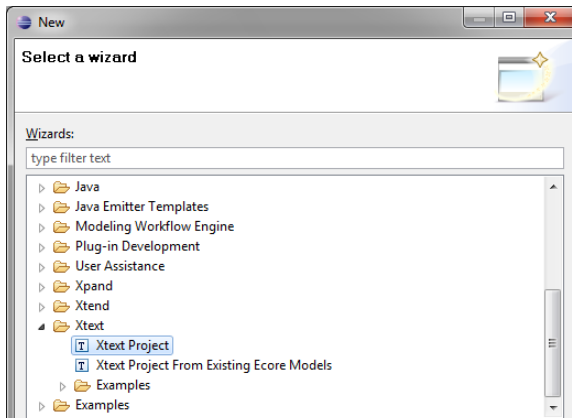
Dans la barre de tache **search** qui s’affiche écrire sélectionner **tools (**au deuxième onglet c’est à dire celui de **All catégories)** puis écrire Xtext et en suite valider sur entré



Une les plugins de Xtext ont été tous téléchargé, maintenant on peut créer notre projet Xtext.

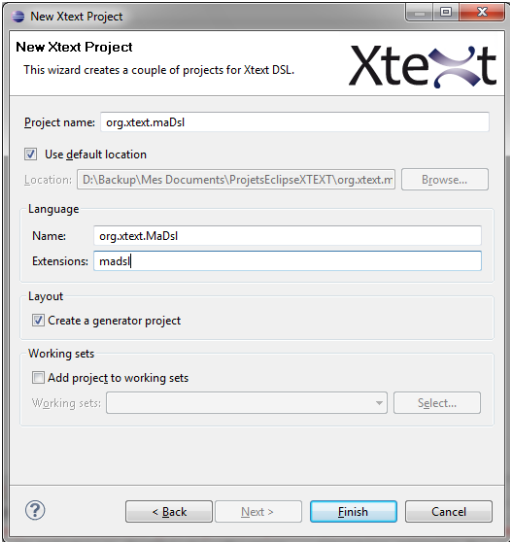


En suit choisie Xtext projet.



Dans la fenêtre qui apparaît, donner un nom à votre projet dans le champ Project name. Exemple : **maDsl**. (**NB** : ce nom doit commencer par une lettre minuscule). Puis, donner un nom à votre langage. Exemple : **MaDsl** (N.B. : ce nom doit commencer par une lettre majuscule). Enfin, nommer l'extension des fichiers où seront enregistrés les programmes de votre langage. Exemple : madsl. Cliquer sur **finish**, pour terminer la création. Il est mieux de créer notre projet dans le package de base org.xtextfourni par Xtext. Ce qui fait que lors de la saisie du project name, nous devons effacer ce qui est écrit par défaut et saisir comme nom de projet : org.xtext.maDSL. Cela est pareil pour le nom de votre langage. De plus, il vaut mieux laisser les cases à cocher telles qu'elles sont. Enfin, il serait quand même intéressant de choisir des noms significatifs pour des projets réels et sérieux et non des choses comme maDsl.

Voici l'aperçu de cette fenêtre :

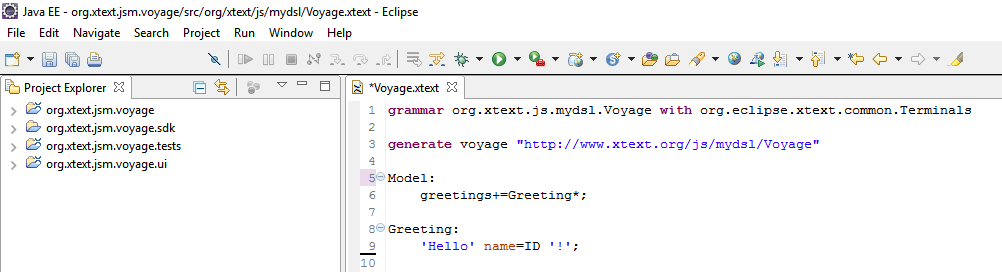


Dans la fenêtre Eclipse qui apparaît, on voit le projet Xtext qui est créé et qui est composé de quatre parties (**org.Xtext.jsm.voyage, org.xtext.jsm.voyage.sdk**, **org.xtext.jsm.voyage.tests** et enfin **org.xtext.jsm.voyage.ui**).

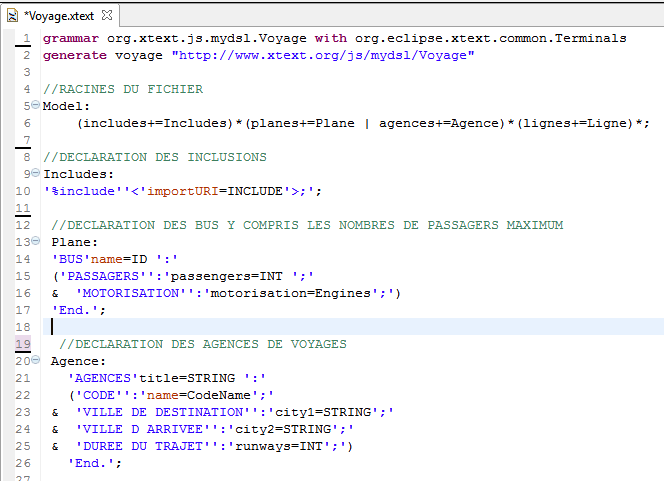
**Remarque :**

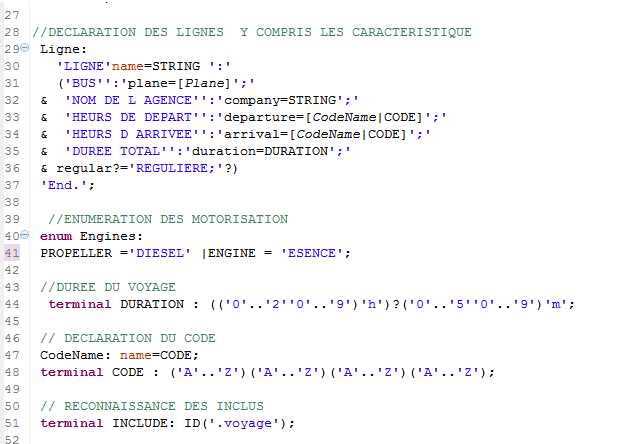
* Il arrive souvent que moins de quatre partie sont créer dans ce cas ne vous s’inquiétez pas car c’est généré de façon automatique par l’ordinateur et a noté que c’est le premier dossier qui va nous intéresser.
* C’est vous qui choisissez comment vous allez appellé votre DSL. Ici mois j’ai choisi l’appeler voyage car suis entrain de représenter les informations sur les agences de voyage

Voici l'aperçu de cette fenêtre :



Nous allons remplacer ce bout de code par le code suivant





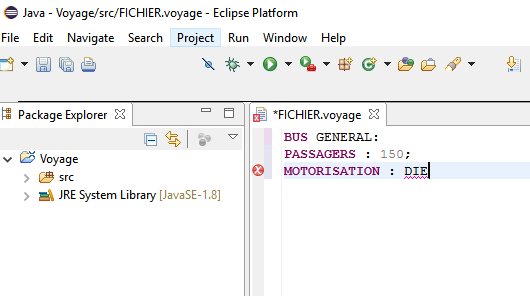
Ce bout de code nous permet de représenter le fonctionnement d’une agence de voyage. Pour savoir comment cela fonction il faut juste créer un fichier dans le dossier **org.jsm.xtext.voyage et** commence à écrire vous allez remarquer une reconnaissance automatique du langage .

**VI. Exécution et test de saisi d’un programme**

Pour exécuter le projet, nous devons cliquer **à droite** sur le premier des trois projets (**org.jsm.xtext.voyage**), dans le menu qui s'ouvre, choisir un As, puis

**1 Eclipse Application**. Une nouvelle fenêtre Eclipse va se lancer et s'ouvrir. Dans cette nouvelle fenêtre Eclipse, nous allons créer un nouveau projet de type général **(File-->Project-->New-->General-->Project).** Renseigner le nom du projet dans la fenêtre qui s'ouvre **(ex : Voyage).** Une fois ce projet créé, cliquer droit sur ce **projet**, puis choisir New dans le menu qui apparait, choisir **File** dans le nouveau menu qui apparait également. Une fenêtre s'ouvre, dans cette fenêtre saisir dans le champ **File name** le nom d'un fichier du projet **FileXtext** avec l'extension. Madsl (exemple : **Voyage**). Cliquer sur OK. Eclipse va se rendre compte qu'il s'agit d'un fichier d'édition des programmes votre DSL et va nous demander s'il importe Xtext dans ce projet.

Voici l'aperçu de cette fenêtre :



**Conclusion**

Au terme de ce travail, nous avons abordé la notion de DSL et nous avons appris comment mettre sur pied un DSL à travers le Framework Xtext. Nous avons également évoqué d'autres technologies aidant à la création des DSL. Par un exemple banal, nous avons appris à écrire une grammaire d'un langage sous Xtext et comment exécuter notre projet jusqu'à la réalisation de tests pour voir si ce projet Xtext réalise ce que nous attendons du langage.