

Ingénierie des Modèles (IDM)

TP2 – Domain-Specific Languages (DSLs), Xtext

On souhaite développer un langage pour **VideoGen**, un système qui permettra de générer, à partir de spécifications textuelles, un générateur de vidéos sur le Web.

Le développement des langages s'effectuera en plusieurs étapes, avec l'utilisation de Xtext et Xtend.

Q0 : Effectuez le 5-minutes tutorial de Xtext :

http://www.eclipse.org/Xtext/documentation/101_five_minutes.html

(nul besoin d'installer un nouvel Eclipse, celui du TP1 est suffisant)

1. Les débuts de VideoGen (introduction à Xtext)

Pour commencer, il s'agit de produire une extension du langage de composition de vidéos introduit en cours. Voici un descriptif de VideoGen.

On souhaite spécifier un système pour assembler des séquences vidéo et ainsi générer des variantes de vidéo.

L'ordre des séquences est important (la 1^{ère} séquence vidéo est jouée, puis la 2^{nde}, etc.).

Une variante vidéo correspond à un ensemble de séquences vidéo jouées dans un certain ordre.

Certaines séquences vidéo sont :

- **obligatoires**, i.e., incluses dans toutes les variantes générées ;
- **optionnelles**, i.e., elles peuvent ne pas être incluses dans certaines variantes ;
- **des alternatives**, i.e., pour la séquence considérée, plusieurs vidéos sont candidates mais une seule sera retenue et sera incluse dans la variante finale.

Une spécification s'attache à décrire un ensemble de séquences vidéo.

La spécification débute par le mot clé « VideoGen » et le reste du contenu est entre deux accolades. La description d'une séquence vidéo consiste en :

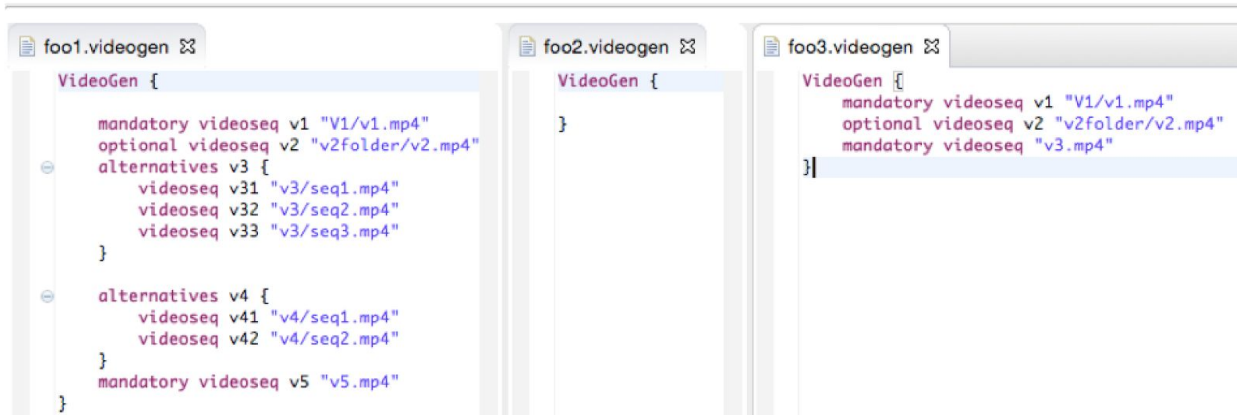
- un identifiant (l'information peut être omise lors de la spécification) ;
- une chaîne de caractères pour localiser la vidéo (en local) ;

Q1: Créez un nouveau projet Xtext et élaborer une grammaire Xtext qui :

- se conforme aux exigences décrites ci-dessus ;
- permet de spécifier les 3 exemples ci-dessous.

Pour valider votre travail :

- utilisez l'éditeur généré par Xtext pour vérifier que les 3 exemples ci-dessous sont effectivement conformes à votre grammaire ;
- créez un ensemble d'exemples supplémentaires (toujours en utilisant l'éditeur généré par Xtext) pour démontrer que vous répondez aux exigences. Pensez aussi aux spécifications qui ne doivent pas être conformes à la grammaire.



Q2 : Programmer des cas de test pour automatiser la vérification de conformité. Les cas de test prendront en entrée des spécifications (typiquement l'ensemble d'exemples spécifiés à la main avec l'éditeur) et vérifieront si ces spécifications sont correctes ou incorrectes. Pour réaliser les tests, vous utiliserez Xtend et l'API générée par Xtext pour (1) parser les exemples et obtenir les modèles (ASTs) correspondants (2) vérifier certaines propriétés des modèles en sortie (par exemple l'absence de redondance dans les identifiants de vidéo).

Vous créerez vos cas de test dans le projet de test automatiquement généré par Xtext. Pour charger / sauvegarder / analyser un modèle, vous pouvez utiliser les ressources présentes ici : <https://github.com/acherm/teaching/tree/master/MDE> qui reprennent les exemples de code vus en cours.

Par exemple:

<https://github.com/acherm/teaching/blob/master/MDE/fr.inria.k3.questionnaire/src/fr/inria/k3/questionnaire/QuestionnaireDemonstrations.xtend> sur l'exemple du questionnaire (à vous d'adapter le code pour qu'il fonctionne avec VideoGen).

2. Quelques extensions pour VideoGen (Xtext)

Le travail précédent nous fournit toute l'infrastructure (API, métamodèle, éditeur, etc.) pour éditer, parser, analyser et transformer une spécification textuelle décrivant l'assemblage (possible) de séquences vidéo. Nous nous appuierons sur cette infrastructure par la suite pour transformer ce modèle en d'autres artefacts (e.g., playlist compréhensible par un lecteur de vidéo).

Nous allons maintenant étendre légèrement le langage et donc la grammaire.

La description d'une séquence vidéo peut maintenant inclure des informations supplémentaires:

- un descriptif de la vidéo ;
- une durée ;
- une probabilité d'apparition.

Ces informations peuvent cependant être omises lors de la spécification, par exemple, il est possible que la durée de la vidéo ne soit pas spécifiée. Si les probabilités ne sont pas spécifiées, on considèrera alors que:

- une séquence vidéo « optionnelle » a une probabilité d'inclusion de 50% ;
- au sein d'une séquence avec des alternatives, il y a équiprobabilité pour les vidéos candidates.

Q3 : Réutiliser les tests de la question Q2 pour implémenter des vérifications supplémentaires; par exemple pour s'assurer que les probabilités des vidéos au sein d'une séquence d'alternatives sont cohérentes.

3. Des métamodèles aux grammaires (Here and Back Again?)

Nous considérons le métamodèle généré à partir de votre grammaire par Xtext (.ecore dans le dossier models/ de votre projet Xtext). Nous appellerons ce métamodèle MM1 et la grammaire obtenue à la fin de la question 3 (**Q3**) GR1.

Nous allons étendre MM1 et l'éditer. Nous obtiendrons ainsi un nouveau métamodèle MM2. Pour ce faire, copiez le métamodèle MM1 dans un nouveau projet. Étendez ensuite ce métamodèle en utilisant l'éditeur Ecore avec les éléments suivants:

- une classe « VideoGenInformation » contenant l'attribut « authorName », l'attribut « creationDate », et l'attribut « version »
- une référence entre « VideoGenInformation » et la classe racine de votre métamodèle (le nom de la référence sera « information » et la cardinalité 0..1).

Créez un nouveau projet Xtext à partir de votre nouveau métamodèle MM2. Xtext génère alors une nouvelle grammaire GR2.

Q4 : Comparez cette nouvelle grammaire (GR2) à la grammaire obtenue à la fin de la question 3 (GR1).

Q5 : Editez la grammaire GR1 obtenue à la fin de la question 3 pour permettre l'ajout d'informations (nom de l'auteur, date de création, numéro de version, etc.) dans la spécification VideoGen. On l'appellera GR3. Discutez de la syntaxe de GR3 par rapport à la syntaxe de GR2.

Q6 : Générez le métamodèle à partir de GR3 (avec Xtext). Comparer le métamodèle obtenu (MM3) avec MM2, MM2 est-il strictement identique à MM3 ? Les instances de MM3 sont-elles également des instances de MM1 ? Discutez.