

Examen IDM

12/2015

Tous documents autorisés, 2h

Questions de cours (5 points)

- 1) Une spécification textuelle (e.g., code source d'un programme) peut être vue et traitée comme un « modèle » (au sens de l'ingénierie dirigée par les modèles). Expliquer comment Xtext implémente techniquement cette idée (en 10 lignes maximum et un seul schéma pour illustrer vos propos).
- 2) Qu'est-ce que « ecore.ecore » ?

Questions pour un générateur (15 points)

On souhaite implémenter un générateur capable, à partir d'une spécification textuelle, d'assembler des séquences « audio » pour former une piste « audio » complète.

Plus spécifiquement, notre générateur se focalise sur la génération de parodies de Julien Lepers, lorsqu'il présente l'émission « Questions pour un champion ». L'idée générale est d'assembler des bouts de questions et de réponses formulées par Julien Lepers dans différentes émissions.

Il y a différents types de séquence audio :

- Intro : l'introduction de l'émission
- Question : une question
- Prénom : le prénom d'un candidat
- Réponse : une réponse
- Buzzer : un bruit de buzzer seul
- Oui : Julien Lepers confirme la réponse (« Oh oui oui oui »)
- Non : Julien Lepers réfute la réponse
- Autre : n'importe quelle autre type de piste audio

Pour chaque type de séquences audio, il y a plusieurs alternatives possibles (donc il y a plusieurs fichiers audio qui sont des « Intro », des « Question », ..., des « Non »).

Les fichiers audio sont typiquement organisés dans plusieurs sous-dossiers et structurés par convention de la manière suivante (cf exemple Figure 1): les fichiers audio « Intro » sont localisés dans « intros/ », les fichiers audio « Question » sont localisés dans un dossier « questions/ », etc.

Une variante de parodie est par exemple constituée de l'assemblage des pistes audio suivantes : opening.mp3 (une « Intro »), saucisson.mp3 (une « Question »), buz1.mp3 (un « Buzzer »), micheline.mp3 (un « Prénom »), jla.mp3 (une « Réponse »), ouiouioui.mp3 (un « Oui »).

Figure 1: Un exemple de dossier pour notre générateur

```
macher-wifi:IDM1516-qgen macher1$ ls -R
answers          firstnames       no               questions
buzzers          intros           others          yes

./answers:
a1.mp3  a2.mp3  jla.mp3

./buzzers:
buz1.mp3          longbuzz.mp3      middlebuzz.mp3  simplebuzz.mp3

./firstnames:
fr.mp3           jm.mp3            micheline.mp3

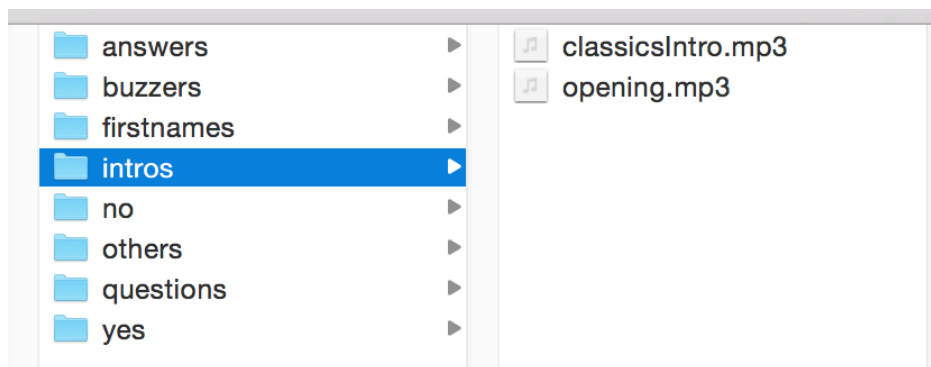
./intros:
classicsIntro.mp3  intro2.mp3          opening.mp3
intro1.mp3         intro3.mp3

./no:
cnon.mp3          cnonbis.mp3

./others:
o1.mp3  o4.mp3

./questions:
ouestdonc.mp3  q1.mp3          q2.mp3          saucisson.mp3

./yes:
ouioui.mp3     ouiouioui.mp3  ouiouioui2.mp3
```



On souhaite spécifier des modèles pour notre générateur. Un modèle sera exploité par notre générateur pour fabriquer une variante par assemblage des fichiers audio.

Un modèle doit d'une part représenter et organiser l'ensemble des pistes audio possibles (une piste audio ayant une « localisation » sous forme de chaîne de caractère et un « type » d'appartenance). D'autre part, le modèle doit aussi servir à spécifier la politique et l'ordre d'assemblage des pistes audio choisi par l'utilisateur final. Par exemple, l'utilisateur final devra pouvoir spécifier :

- qu'une variante consiste en une « Intro », une « Question », et un « Buzzer », une « Réponse »
- qu'une variante consiste en une « Question », une suite de 1 à 3 de (« Réponse », suivi d'un « Non »), puis une « Réponse », un « Oui »
- qu'une variante consiste en une « Question », un « Buzzer », une « Réponse », un « Oui »

Le générateur interprètera ce modèle pour choisir une piste audio par type (e.g., de manière aléatoire) et ensuite fait appel à une procédure (e.g., avec ffmpeg) qui assemble les fichiers audio préalablement choisis.

- 3) Proposer un métamodèle MM1 permettant la création de tels modèles.
- 4) Inventer une syntaxe textuelle permettant d'exprimer un modèle de « questions pour un générateur ». Donner la spécification textuelle dans cette nouvelle syntaxe d'un modèle qui :
- traite en entrée les fichiers audio du modèle de la Figure 1
 - avec comme politique d'assemblage une « Intro » suivie d'une « Question » suivie d'un « Buzzer » suivie d'une « Réponse » suivie d'un « Oui »
- 5) Exprimer la grammaire Xtext de cette syntaxe textuelle
- 6) Dériver le métamodèle Ecore (MM2) correspondant à la grammaire Xtext précédente
- 7) Est-ce que MM2 est strictement identique à MM1? Justifier.
- 8) Ecrire en Xtend une fonction capable de générer aléatoirement une variante d'une piste audio au format « .m3u¹ ». En prenant en entrée le modèle de la question 4, on pourrait générer $5 * 4 * 4 * 3 * 3 = 720$ variantes. Une variante « .m3u » parmi les 720 est :

opening.mp3

ouestdonc.mp3

buz1.mp3

jla.mp3

ouiouioui.mp3

(QUESTION BONUS #1) Plutôt que d'écrire les modèles à la main, on souhaite développer une procédure qui prend un dossier en entrée, parcourt les sous-dossiers selon la convention de nommage (« intros », « questions », etc.) et génère un modèle au format textuel en se limitant aux informations concernant (1) les localisations des fichiers ; (2) leur classification par type. Décrire le principe général de cette transformation.

(QUESTION BONUS #2) Donner un exemple montrant qu'un langage dédié (DSL) peut être utilisé pour développer des variantes d'un logiciel (réponse en 10 lignes maximum. Les 11ème et suivantes ne seront pas lues)

¹ Le format « m3u » consiste en un ensemble de localisations de fichiers sources (mp3, mp4, avi, etc.), chaque localisation étant séparée par un saut de ligne