

Conclusions et perspectives ☺

Cedric Dumoulin

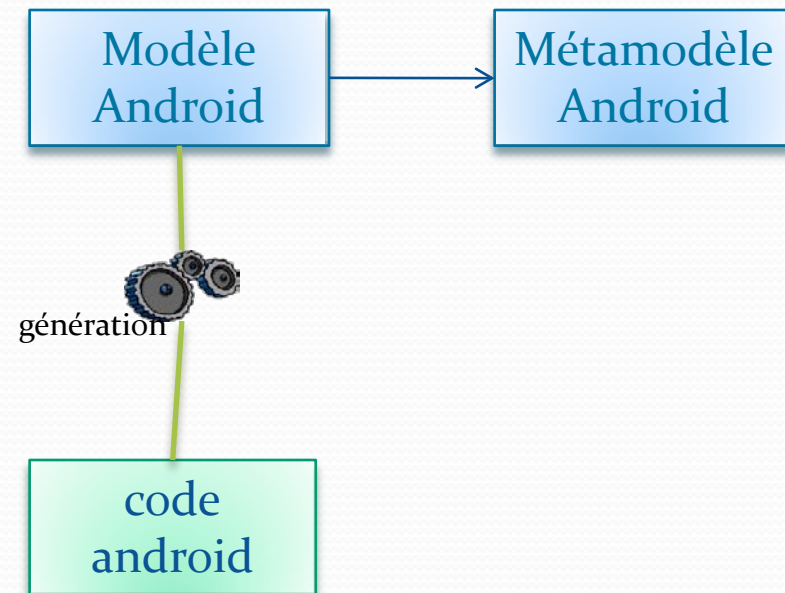
Plan

- Etat actuel
- Une meilleur architecture
 - Xtext, plusieurs MM, transfo
- Contenu minimum
- Un peu plus
- Exemple d'applis générée

Etat actuel

Une approche simple

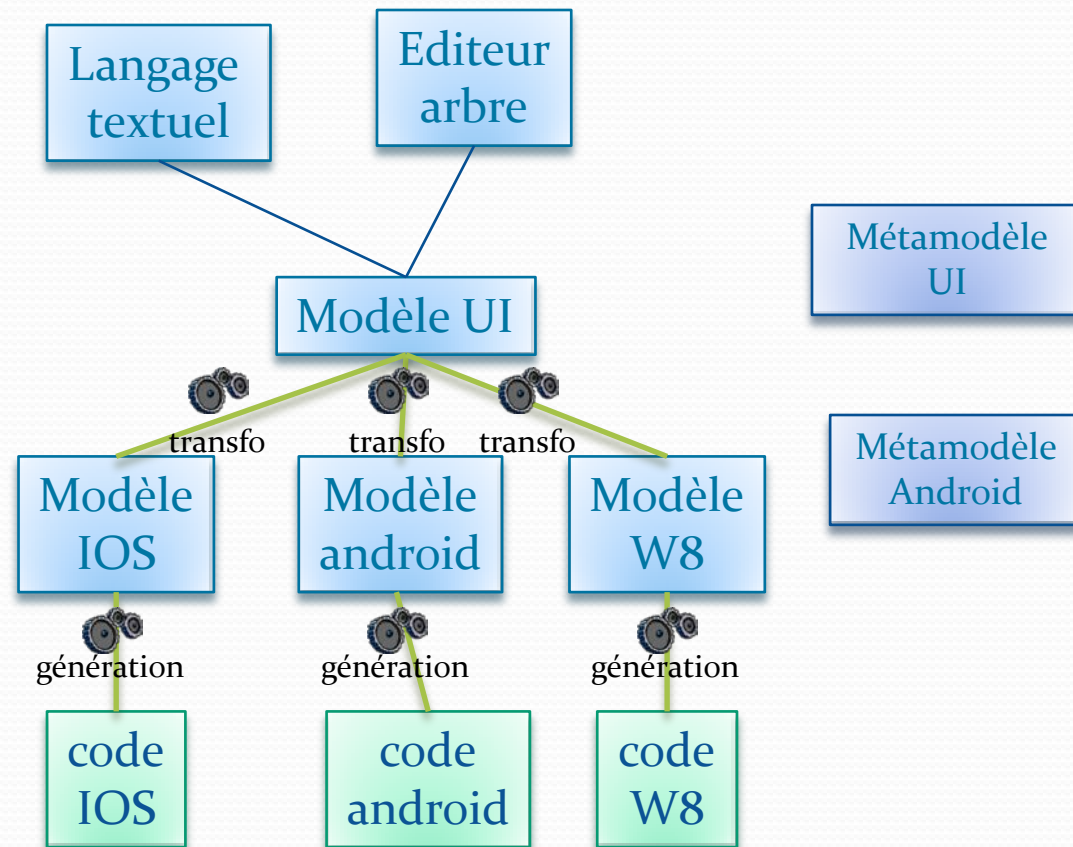
- Un MM android
- Génère directement le code à partir des instances de ce modèle
- Inconvénients :
 - Si langage trop abstrait
 - ➔ Problèmes lors de la génération (ex: générer les ressources, les méthodes des activity, ...)
 - Si langage plus proche d'Android
 - ➔ Pas assez d'abstractions, imitation d'android



Une meilleur architecture

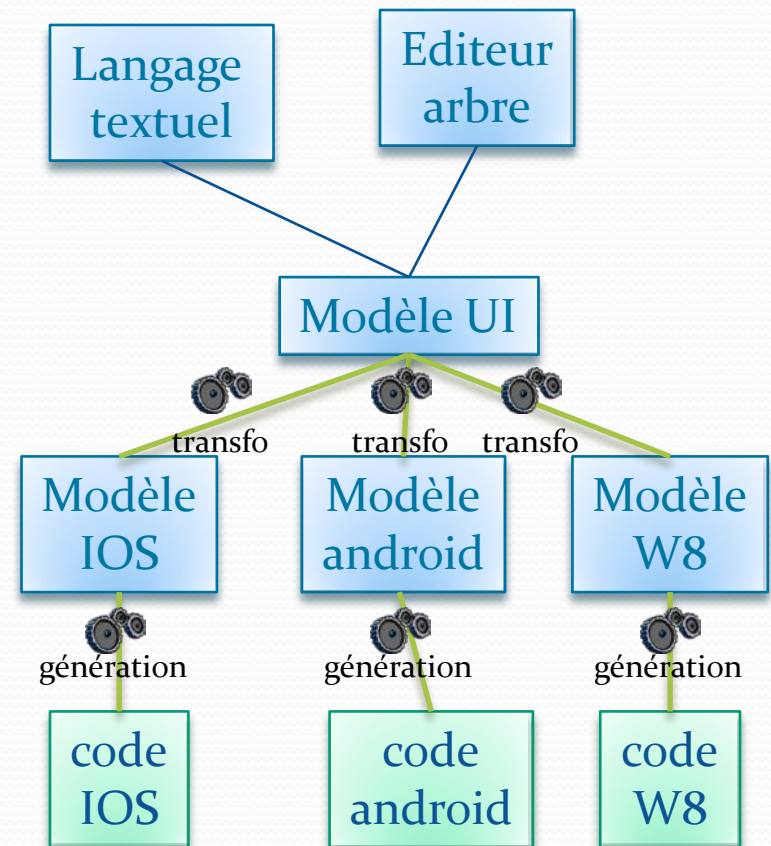
Plus (+) de métamodèles

- Un métamodèle bas niveau, proche de Android
 - Plus facile pour générer le code
- Un métamodèle haut niveau pour concevoir
 - Moins de considérations techniques
- Une transfo entre les deux
 - C'est la transfo qui prend en charge l'introduction des solutions techniques



Syntaxe abstraite et concrete

- Metamodelle == syntaxe abstraite
 - Il peut y avoir des constructions peut naturelle pour un humain
- Syntaxe concrete : utilisé pour modéliser
 - Ex: editeur arborescent
- On peut ameliorer la syntaxe concrete:
 - Editeur textuel
 - Semi automatique avec XText



Contenu

- 
- Les fichiers générés doivent compiler sans erreurs, sans avoir à les retoucher manuellement

Défi

- Le défi
 - Etre capable de modeliser l'application de Palindrome
 - La modelisation doit etre d'un niveau plus élevé que la simple transcription des concepts d'Android
 - Le défi conciste a generer 100% de l'application
 - Il ne doit pas etre necessaire d'intervenir dans le code généré.
 - Pour le corps des methodes, il est possible de modeliser le contenu sous la forme d'un texte dans le modele
 - Cela ne doit pas etre generalisé !

Le grand Défi

- Le grand défi
 - Etre capable de modeliser l'application de Memory.
 - La modelisation doit etre d'un niveau plus élevé que la simple transcription des concepts d'Android
 - Le defi conciste a generer 100% de l'application
 - Il ne doit pas etre necessaire d'intervenir dans le code généré.
 - Pour le corps des methodes, il est possible de modeliser le contenu sous la forme d'un texte dans le modele
 - Cela ne doit pas etre generalisé !