

Auteurs

Yifang Dong
Ahmed El Bir

Encadrant :
Denis Conan

Contexte du projet

DEBS: Distributed Event-Based System

- Dans les systèmes émergents de l'Internet des objets, le modèle publier/souscrire (en anglais publish/subscriber) fait partie des solutions les plus utilisées.
- Les annonces et les souscriptions se font à travers des filtres.
- Les DEBS offrent une architecture logicielle entièrement décentralisée.



Problématique

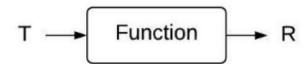
- Un modèle de données semi-structurées
 - les messages échangés sont des documents XML difficiles à analyser
 - les filtres sont écrits en Javascript et exécutés dans un script engine :
- ➔ Dégradation de la performance + difficulté d'écriture et de test des filtres
- Absence (ou peu) de tests unitaires

Démarche

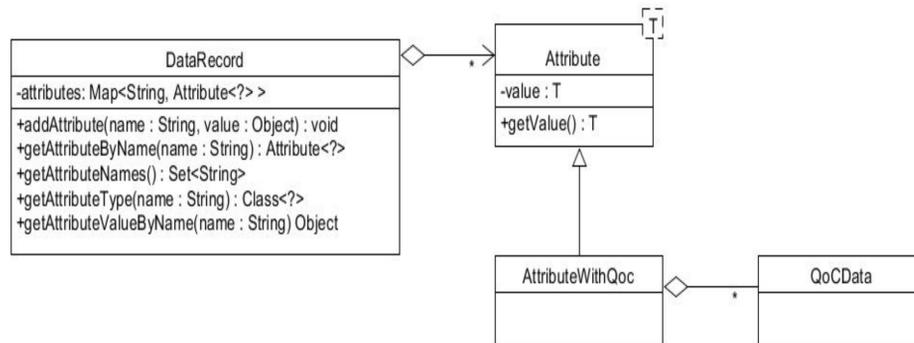
Des filtres avec des lambda expressions

- Création d'une lambda expression à partir d'un String
 - utilisation d'une librairie Open Source : Java 8 - Lambda From String
- Un filtre est une classe avec une interface fonctionnelle comme attribut

```
@FunctionalInterface
public interface Function<T, R> {
    R apply(T t);
}
```



Conception et implémentation du modèle de données structurées

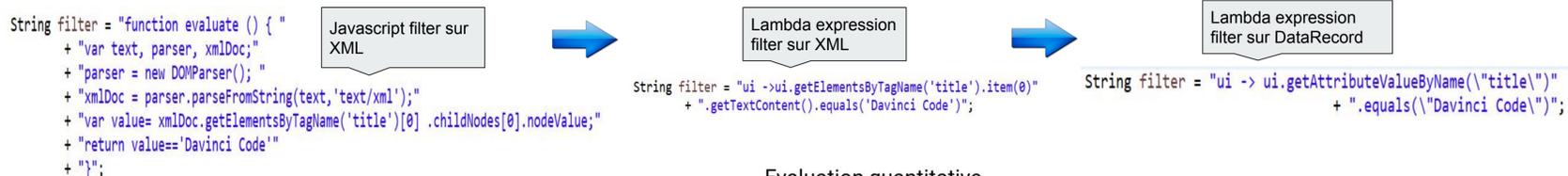


- Ajout de l'option «--mode» pour choisir entre les deux modèles des données structurées et semi-structurées
- Un attribut ModelType dans la classe mère de toutes les publications

Résultat

Évaluation qualitative

- Facilité d'écriture des filtres : 80 % de gain en nombre des lignes de code



Évaluation quantitative

- Gain de 10% en temps d'exécution des filtres avec le modèle des données semi-structurées.
- Gain de 77 % en temps d'exécution des filtres avec le modèle des données structurées.

Evaluation quantitative

