Université de Cergy-Pontoise

RAPPORT

Intégration et Entrepots de Données
Master 1 Informatique et Ingénierie des Systèmes Complexes (IISC)
Parcours Système Intelligents et Distribués(SID)

sur le sujet

Le but de ce projet est de concevoir et réaliser un médiateur simple, exploitant différentes sources de données dans le domaine du cinéma

rédigé par

Mohamed II Bayo,Benoît FAGOT

Avril 2019

Table des matières

Table des figures

1 Description extraction des informations API JSoup :

a) On constate un pattern régulier dans le DOM des pages web du site : les données sont toujours rentrées dans la 2ème table de la page, il suffit donc de sélectionner les <tr> puis les <td> de la table mis à part les deux dernières qui sont des données récapitulatives.

b) Pour obtenir les données des années 2000 à 2015, on effectue une boucle sur l'adresse de la page, et on récidive pour chaque genre demandé.

c) On extrait les données de la table qui nous intéressent à savoir le distributeur et le titre du film.

d) Pour chaque genre, on créé et écrit dans un fichier CSV contenant une colonne Genre éponyme, et on remplit les deux autres colonnes par les données extraites précedemment.

e) Chaque fichier csv aura pour entete(genre, titre, distributeur)

Exemple pratique :

— on fait une combinaison de l'année et du genre pour former l'URL correspondant à l'année et le genre du film à extraire
exemple :http ://www.the-numbers.com/market/" + i + "/genre/"+genres[j]).get() ici i = année, et genre[j] un élément de la liste des genres.Par exemple pour l'année 2000 et genre Adventure , on aura :
http ://www.the-numbers.com/market/2000/genre/Adventure

— ensuite,
  — On selectionnne la balise table -> les tr Exemple : Elements tables = doc.select("table").get(1).select("tr") ;
  — On regarde ensuite pour chaque <tr> les <td> :
  for (Element headline : tables) {
  Elements tds = headline.select("td") ;
  }
  — On extrait les <td> qui nous intéressent (tds.get(1) && tds.get(3))

2 La description des transformation sur Talend :

2.1 Fusion des fichiers extrait de la page HTML avec l'API JSoup :

Après avoir extrait le fichier csv pour chaque genre avec JSoup, on les a fusionné afin de former un seul fichier csv sur talend avec le composant "tUnite".

— Le composant tUnite fusionne des données de diverses sources.

— tlogRow pour afficher le résultat de la fusion des 6 fichiers csv dans la console Run.

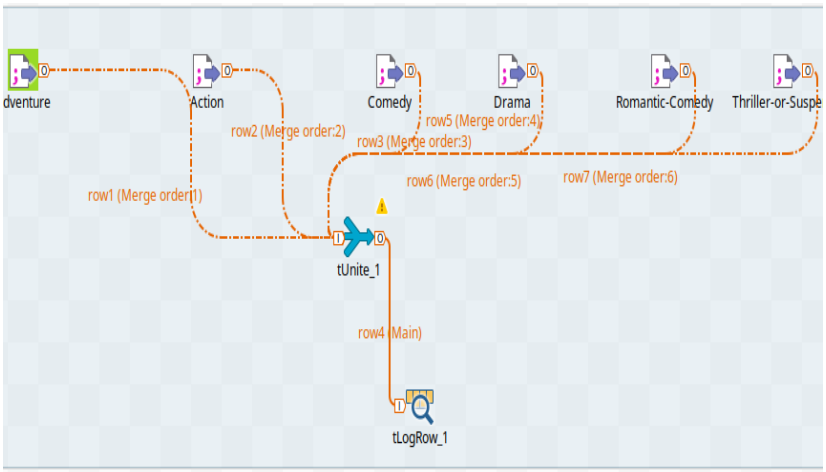

FIGURE 1 – Fichier genre extrait avec JSoup

2.2 Configuration des mappages avec tMap :

a) L'éditeur de tmap nous affiche trois tables movieBudget, genreFichierFusionJsoup et movieOut correspondant respectivement au schéma du fichier de movieBudget, au schéma du fichier de fusion des six fichiers genre extrait du jsoup et au schéma de sortie pour les informations de film.

b) On a sélectionné la colonne movie dans la table movieBudget et déposez dans la colonne titre de la table genreFichierFusionJsoup pour créer une jointure entre les deux ensembles de données en entrée en fonction des titres. Puis sur le buton paramètre tMap de la table genreFichierFusionJsoup, on a choisi la jointure externe gauche

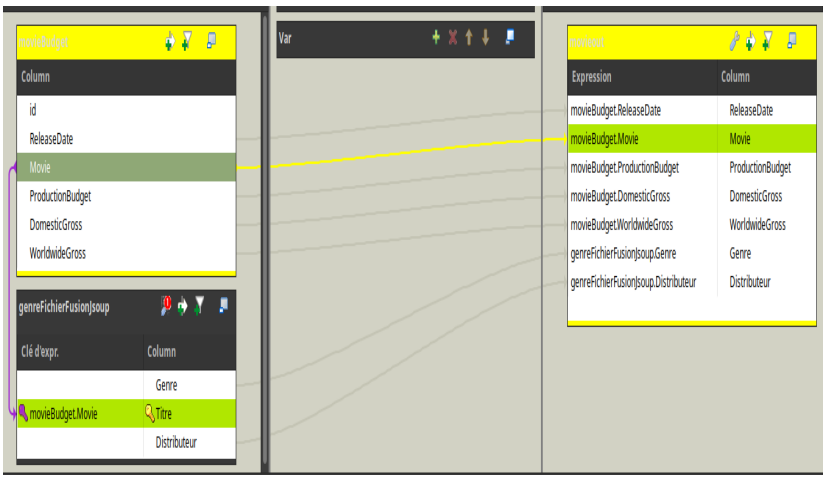

FIGURE 2 – configuration tMap

2.3 Sauvegarde des résultats de traitement dans la base de données :

Avec le composant tDBOutput on a crée une connexion à notre base de donnée "Mysql", puis on a cliquer sur tMap pour le lier à tDBOutput. La base de donnée recevra les données de sortie de tMap de la table "movieOut"

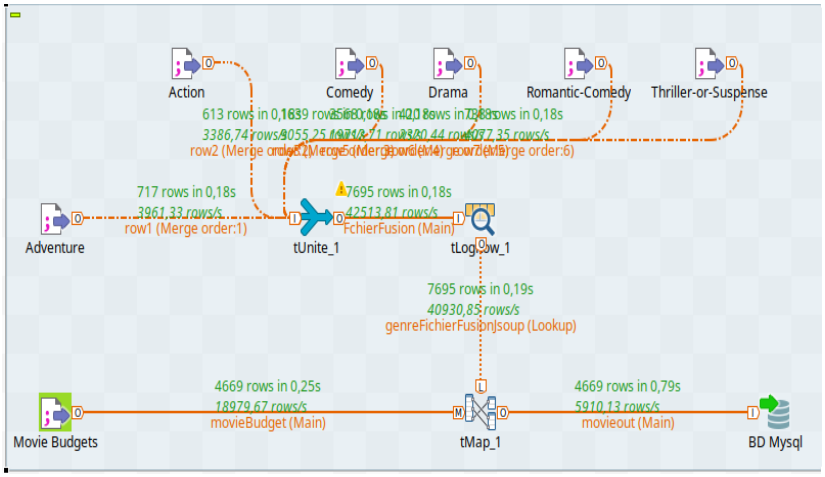

FIGURE 3 – resultat final

3 Le mapping entre le médiateur et les 3 sources

3.1 Source 1 : BD locale

Avec JDBC on fait une requête SQL classique pour obtenir l'année de sortie, les différents revenus du film, le budget, le genre, le distributeur.

SELECT * FROM movie WHERE Movie LIKE title

3.2 Source 2 : DBpedia

Avec l'API Jena, nous effectuons nos requêtes SPARQL puis stockons les résultats en mémoire. Nous effectuons 4 requêtes différentes pour obtenir le réalisateur d'un film, la liste des acteurs d'un film, la liste des producteurs d'un film, la liste des films où un acteur a joué. Ce sont les requêtes vu en TP. Il est primordial de renseigner l'ontologie suivante : PREFIX dbo : <http ://dbpedi a.org/ontology/>

SELECT ?nr WHERE { ?f a dbo :Film ; foaf :name "title"@en ; dbo :starring ?ac. ?ac foaf :name ?nr. }
SELECT ?nr WHERE { ?f a dbo :Film ; foaf :name "title"@en ; dbo :director ?d. ?d foaf :name ?nr. }
SELECT ?nr WHERE { ?f a dbo :Film ; foaf :name "title"@en ; dbo :producer ?d. ?d foaf :name ?nr. }
SELECT ?t WHERE { ?f a dbo :Film ; foaf :name ?t ; dbo :starring ?a. ?a foaf :name "actor"@en }

3.3 Source 3 : OpenMovieDatabase

Avec le service REST, nous envoyons notre requête GET avec comme paramètre t le titre du film et r le format de résultat souhaité (XML) plus la clé API générée. Nous stockons ensuite le résultat en mémoire.
Object o = XPath("http ://www.omdbapi.com/ ?apikey=6ea8b522&t=Titanic&r=xml","/root/movie/ @plot",XPathConstants.STRING) ;

4 La structure du programme du médiateur et son fonctionnement :

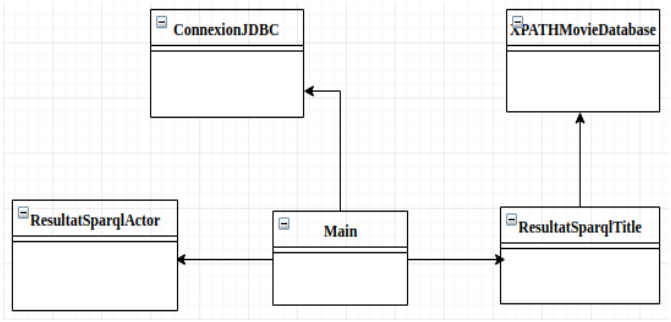

FIGURE 4 – UML

Le médiateur fonctionne dans un LOOP infini afin de pouvoir enchaîner les requêtes. On demande tout d'abord de choisir le type de requête (par titre ou par acteur) puis on demande de renseigner ce champ. Switch(type) :

— Si la requête est de type Titre, on va d'abord créer un objet Resultat, puis à l'aide de la source 1 (BD locale) on assigne date de sortie, genre, distributeur, budget, revenus aux Etats Unis, revenus mondiaux. Maintenant avec la source 2 (DBpedia) on requête pour obtenir le réalisateur et la liste des acteurs qu'on assigne à notre objet.
Finalement, on obtient avec la source 3 (OPD) l'intrigue du film qu'on affecte de même, puis on procèdre à l'affichage récapitulatif des informations obtenues.

— Si la requête est de type Acteur, on créé un objet Résultat2, qui contient une liste d'objets Résultats. On affecte le nom d'acteur à notre objet Résultat2, puis à l'aide la source DBpedia on obtient la liste des films dans lesquels il a joué.
Pour chacun de ces films, on créé un objet Résultat qu'on intègre dans Résultat2, et ensuite avec la BD Locale on obtient et affecte la date de sortie, le genre, le distributeur, puis avec la source DPBedia le réalisateur et les producteurs.
Ainsi, à l'aide d'une boucle sur la liste d'objets Résultats de Résultat2, nous affichons toutes les informations des films dans lequel l'acteur a joué.

5 Annexe

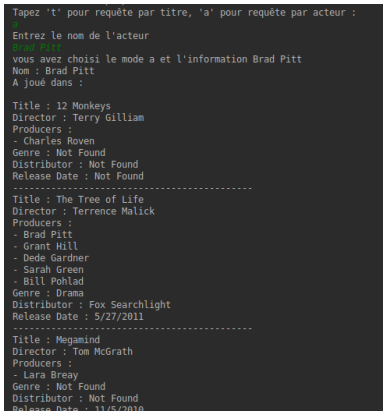

FIGURE 5 – Exemple resultat console : fig 1 : Actor "Brad Pitt", fig 2 : t=Title "Avatar"