

Modèle de Connaissances et Web Sémantique

Rapport de Projet

Groupe:
ALLEMAND Fabien
COUTURE Louise



1 Construction de l'Ontologie

L'objectif de ce projet est de créer une ontologie OWL permettant de représenter le plus précisément possible les informations sur des chansons. Pour cela, nous utilisons le logiciel Protégé développé par l'Université de Stanford ainsi que l'ontologie FOAF (Friend Of A Friend) comme base de notre ontologie.

Nous avons tout d'abord créé des classes primitives correspondant aux chansons, aux groupes, aux musiciens (et leurs spécialités) et aux récompenses.



Figure 1: Hiérarchie des chansons



Figure 2: Hiérarchie des groupes



Figure 3: Hiérarchie des musiciens



Figure 4: Hiérarchie des récompenses

Il est important de préciser que la plupart des liens sont simples. Par exemple: `aNationalité` permet d'associer un élément de type `Person` à une `Nationalité`. Cependant, afin de traduire l'appartenance d'un artiste à plusieurs groupes, avec des rôles et des périodes données, nous avons du introduire une classe intermédiaire nommée `ParticipationDansGroupe`. Cette classe peut contenir des assertions qui définissent les dates d'entrée et de sortie du groupe ainsi que le rôle d'un artiste dans ce groupe. Le lien avec le groupe en lui même n'est fait qu'au travers du lien `aGroupe` qui relie une `ParticipationDansGroupe` à un `Groupe`.

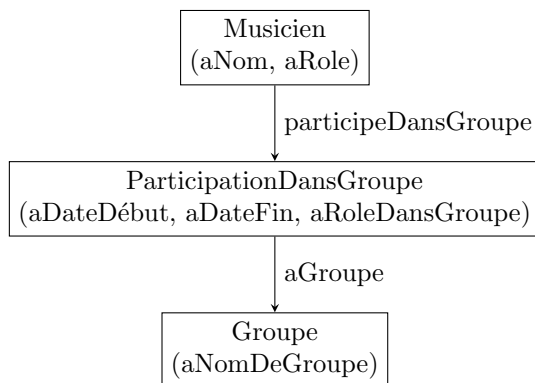


Figure 5: Description du lien entre un Musicien et un Groupe

Nous avons aussi ajouté diverses classes définies pour répondre à nos besoins.

2 Interrogation de l'Ontologie

Nous avons par la suite pu interroger l'ontologie à l'aide de requêtes SPARQL et SQWRL. Par commodité, les requêtes demandées sont écrites dans le fichier `AllemandCouture_Requetes.md`. Voici deux exemples de requêtes qui étaient demandées.

La première est une requête SPARQL qui retrouve les groupes ayant publié moins de 4 albums et affiche le groupe ainsi que le nombre d'albums publiés. On remarque l'utilisation d'une instruction **GROUP BY** pour que la fonction **count** regroupe le calcul par groupe. L'instruction **HAVING** agit comme un filtre, ici, élimine les groupes ayant publié au moins 4 albums.

```

SELECT ?g (count(distinct ?a) as ?count)
WHERE {?a rdf:type ch:Album .
      ?g ch:aPubli ?a }
GROUP BY (?g)          # Regrouper le r sultat par groupe
HAVING (?count < 4) # Conserver les groupes dont le r sultat est inf rieur 4
  
```

La deuxième requête permet de lister chaque participation d'un artiste dans un groupe. On observe en sortie le nom de l'artiste et le groupe. Les trois premières instructions permettent de lier chaque artiste à un groupe et les deux dernières récupèrent le nom de l'artiste et du groupe.

```

Musicien(?m) ^ participeDansGroupe(?m, ?p) ^ aGroupe(?p, ?g) ^ aNom(?m, ?n) ^
aNomDeGroupe(?g, ?ng) -> sqwrl:select(?n, ?ng)
  
```