

Construction d'une ontologie sur les séries télévisées

Florian FALKNER, Matthias FOYER

29/12/20

Modélisation

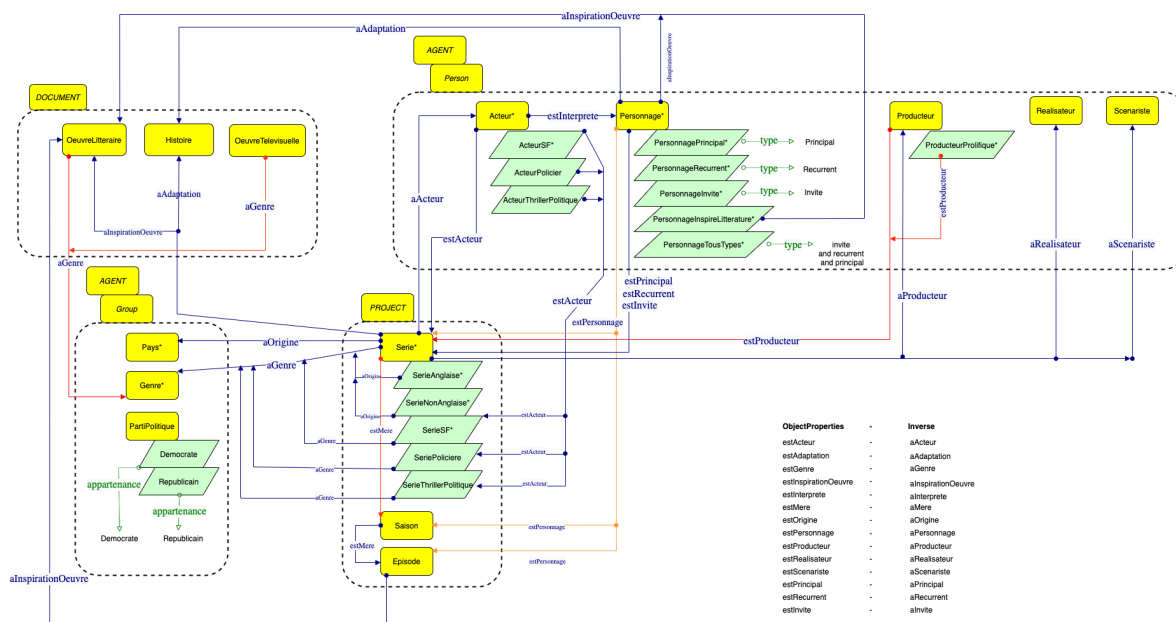


Figure 1: Vous retrouverez ce schéma dans son format d'origine dans l'archive de ce rendu

Classes primitives

Nous avons construit notre ontologie en nous basant au maximum sur les concepts FOAF pré-existants (représentés sur le schéma par les encadrés en pointillés).

Concernant les ajouts que nous avons choisi de faire :

- les classes **Pays**, **Genre** et **PartiPolitique** sont considérés comme des groupes que nous associons naturellement à la classe **Group** du schéma FOAF.
- Les classes **Acteur**, **Personnage**, **Producteur**, **Réalisateur** et **Scénariste** regroupent toutes des personnes. Nous l'associons à la classe **Person** héritée de FOAF.
- Nous considérons les classes **OeuvreLitteraire**, **Histoire** et **OeuvreTelevisuelle** comme appartenant à la classe **Document** FOAF.
- Enfin, les classes **Série**, **Saison** et **Episode** sont sous-classes de **Project** : nous nous plaçons du point de vue d'une maison de production pour qui il s'agit effectivement de projets.

Classes définies

Elles apparaissent en vert sur la figure 1.

En tout, elles sont 16 : sept ont été spécifiées dans le sujet, puis nous avons créé les 9 autres car elles nous semblaient pertinentes et nécessaires.

Parmi celles que nous avons ajoutées nous retrouvons :

- **ActeurPolicier** et **ActeurThrillerPolitique** qui se basent sur le même principe que **ActeurSF**, c'est à dire qu'il s'agit respectivement d'un acteur qui joue au moins dans une série policière et d'un acteur qui joue dans au moins un thriller politique
- **PersonnageRecurrent** et **PersonnageInvite** que nous avons ajoutées pour compléter l'ontologie qui doit pouvoir nous renseigner sur le rôle d'un personnage dans une série
- **SerieSF**, **SeriePolicier** et **SerieThrillerPolitique** qui catégorisent les séries selon leur genre tout comme **SerieAnglaise** et **SerieNonAnglaise**
- **Democrate** et **Republicain** qui représentent des partis politiques

Toutes les relations dans l'ontologie sont représentées sur la figure 1 à l'exception des relations inverses qui rendraient le schéma illisible. Les couleurs des propriétés ne jouent aucun rôle, elles permettent uniquement de rendre la lecture plus simple.

Propriétés

Propriété	Inverse
estActeur	aActeur
estAdapatation	aAdaptation
estGenre	aGenre
estInterprete	aInterprete
estMere	aMere
estOrigine	aOrigine
estPersonnage	aPersonnage
estProducteur	aProducteur
estRealisateur	aRealisateur
estScenariste	aScenariste
estPrincipal	aPrincipal
estRecurrent	aRecurrent
estInvite	aInvite
estInspirationOeuvre	aInspirationOeuvre

Il nous a paru nécessaire de créer une propriété transitive dans les deux sens pour les liens entre Serie, Saison et Episode. Cela nous permet d'assurer une continuité logique dans les appartenances entre épisodes, saisons et séries.

Data properties

- *type* : Cette propriété est ajoutée sur les personnages des séries pour désigner un personnage principal, récurrent ou invité dans une série. Un personnage doit avoir au moins 1 type et au plus 3. Le range a été spécifié pour interdire l'ajout d'autre chose que ces trois littéraux.
- *appartenance* : Cette propriété permet de rattacher un personnage à un parti politique, Démocrate ou Republicain dans notre exemple. Le range a été spécifié pour interdire l'ajout

d'autre chose que ces deux littéraux. Chaque personnage a au plus un rattachement à un parti. Cette propriété est fonctionnelle.

- *foaf:firstName* : Héritée de FOAF, représente le prénom.
- *foaf:LastName* : Héritée de FOAF, représente le nom.
- *foaf:name* : Héritée de FOAF, représente les titres des séries dans notre ontologie.

Des individus ont été ajoutés dans chacune des classes de l'ontologie. Pour qu'un individu puisse être instancié dans la classe définie **Democrate** ou **Republicain**, il faut que sa data property *appartenance* ait la valeur *democrate* ou *republicain*.

Les classes définies **ActeurPolicier**, **ActeurSF**, **ActeurThrillerPolitique** sont attribuées sous condition qu'un acteur joue respectivement dans une série policière, une série de science-fiction ou un thriller politique.

PersonnageInspireLitterature est attribuée aux personnages inspirés d'exactement une œuvre littéraire.

Des classes définies indiquent si un personnage est principal, récurrent, ou invité dans une série. S'il appartient à l'ensemble de ces classes définies il appartient aussi à la classe définie **Personnage-TousTypes**. Pour renseigner le rôle d'un personnage dans une saison spécifique, des propriétés *estPrincipal*, *estRecurrent* et *estInvite* lient le personnage aux saisons concernées.

Pour qu'une série apparaisse dans la classe définie **SerieAnglaise**, elle doit être reliée par la propriété *aOrigine* au pays "Royaume-Uni". Une série n'est définie **SerieNonAnglaise** que si elle a pour origine un autre pays que "Royaume-Uni". Pour cela, nous nous reposons sur une liste exhaustive de tous les pays que possède notre ontologie, mais excluant "Royaume-Uni". Ainsi, notre classe définie a pour condition "*aOrigine* some ({**tous les pays excepté le Royaume-Uni**})". Lorsqu'un pays est ajouté (il s'agit d'individus), il faudra alors le rajouter à cette liste dans la classe **SerieNonAnglaise** pour s'assurer que notre ontologie reste cohérente.

Conventions de nommages

Classes : Les noms de classes commencent par une majuscule, ne contiennent pas d'espace, sont accordés au singulier et en français. Chaque mot composant le nom commence par une majuscule. (*Notation CamelCase ou CamelBack*)

Propriétés : Les noms de propriétés commencent par une minuscule, ne contiennent pas d'espace, sont accordées au singulier et en français. Chaque mot composant le nom commence par une majuscule excepté le premier. Tous les noms de propriétés commencent par « est » ou « a » excepté pour les data properties qui sont en anglais pour respecter les conventions de FOAF.

Limites

La première limite que nous rencontrons concerne une inférence. En effet, un personnage ou un acteur n'est pas automatiquement ajouté à une série lorsqu'il est acteur/personnage d'une saison de cette même série. Il est nécessaire, lorsqu'on ajoute un personnage à une saison, de l'ajouter manuellement à la série correspondante. Cela est imputable à l'hétérogénéité des liens qui existent entre les sous-classes de **Person** et les sous-classes de **Project** (transitivité non-applicable).

Pour s'assurer de la cohérence de notre ontologie, lorsqu'un pays est ajouté il faut l'intégrer à la liste de pays de la classe définie **SerieNonAnglaise** s'il ne s'agit pas du Royaume-Uni.

Questions

Question 1

Parmis les concepts hérités du schéma FOAF nous avons conservé :

- Les classes : **Agent**, **Group**, **Person Document** et **Project**. Nous souhaitions utiliser la classe *LabelProperty*, mais elle s'est avérée instable.
- Les data properties : *firstName*, *lastName* et *name*. Nous n'avons pas conservé *title* car elle concerne la civilité et n'est pas applicable aux séries. De même, les propriétés *maker* et *made* n'ont pas été conservées car ce sont des mots trop généraux.

Question 2

Chaque personnage possède un lien avec les saisons dans lesquelles il apparaît (propriété *estPersonnage*). De cette façon, s'il disparaît pour une saison, il suffit de ne pas ajouter ce lien pour la saison concernée.

De plus, nous avons décidé de créer 3 propriétés supplémentaires : *estPrincipal*, *estRecurrent* et *estInvite*. Elles s'appliquent entre **Personnage** et **Saison** et nous permettent, pour chaque saison où le personnage apparaît, de lui attribuer la récurrence qui correspond.

Pour savoir si un personnage est présent sur toutes les saisons, il suffit de comparer le nombre de récurrences avec le nombre de saisons qui composent la série. Ceci peut se faire à l'aide d'une requête SPARQL en utilisant la fonction count (*c.f. sujet 2 question 10*).

Question 3

Sans utiliser de requête SPARQL ou SQWRL (auquel cas nous répondons à cette question dans le sujet 2), il suffit de lancer Reasoner, de se rendre dans l'onglet "Individuals by class", de sélectionner **OeuvreLitteraire** et de choisir l'œuvre en question. Il sera ensuite possible de voir à quels épisodes cette œuvre est reliée par la propriété *estInspirationOeuvre*.