

Réécriture de patrons de requêtes fondée sur des alignements d'ontologies



Pascal GILLET

Cassia Trojahn - Ollivier Haemmerlé

Institut de Recherche en Informatique de Toulouse -
Equipe MELODI

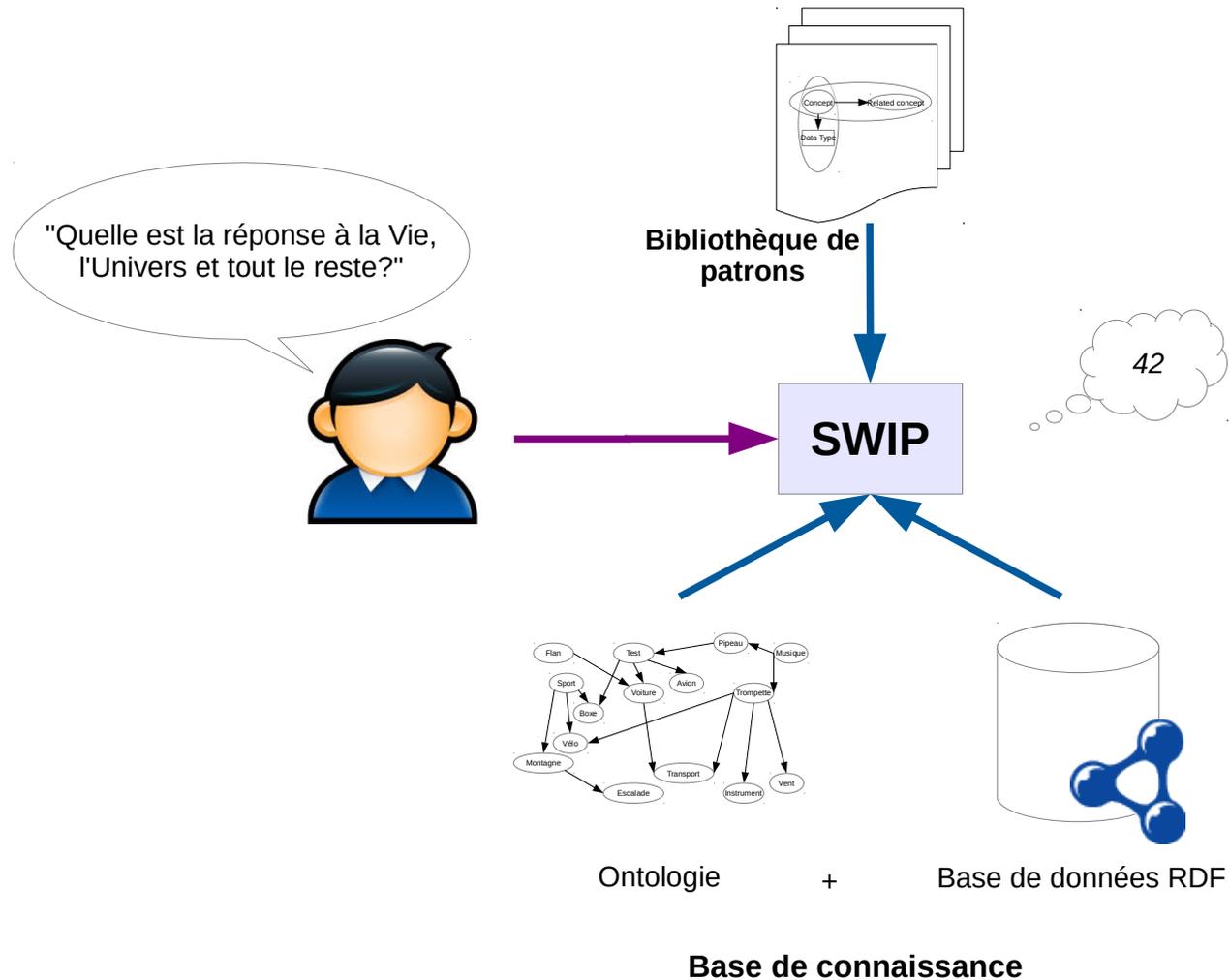


Université
de Toulouse



Systeme SWIP

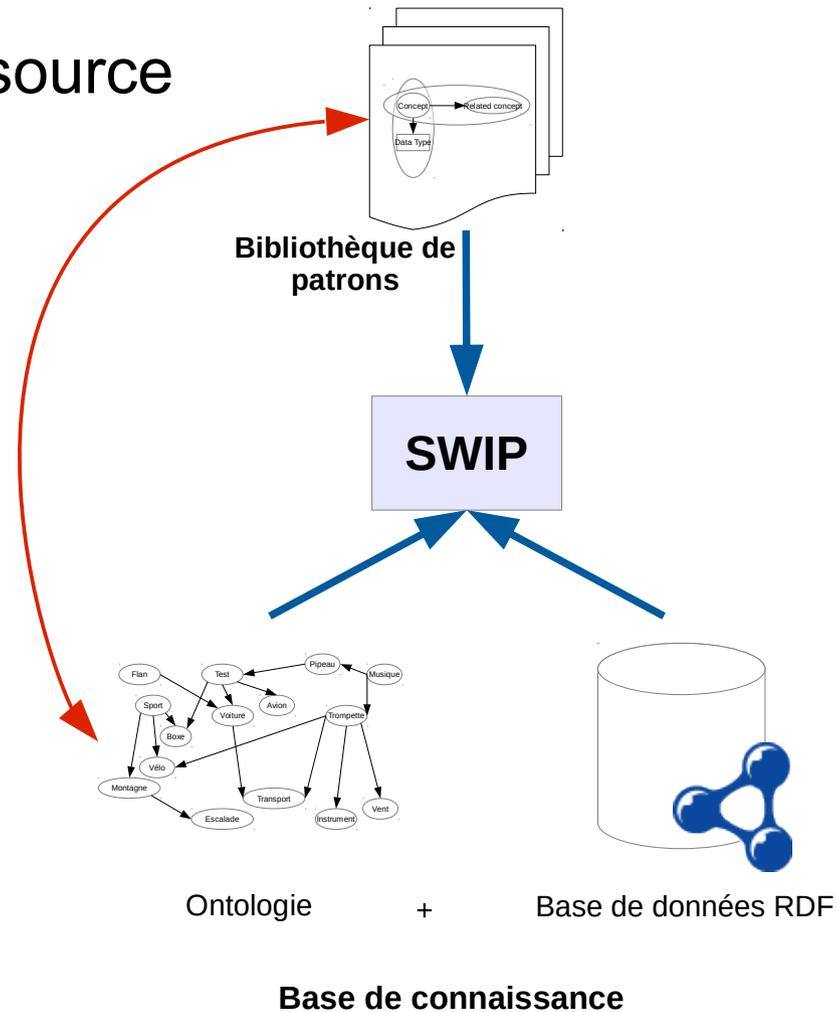
- **Contexte:** système SWIP fondé sur des patrons de requêtes



Systeme SWIP

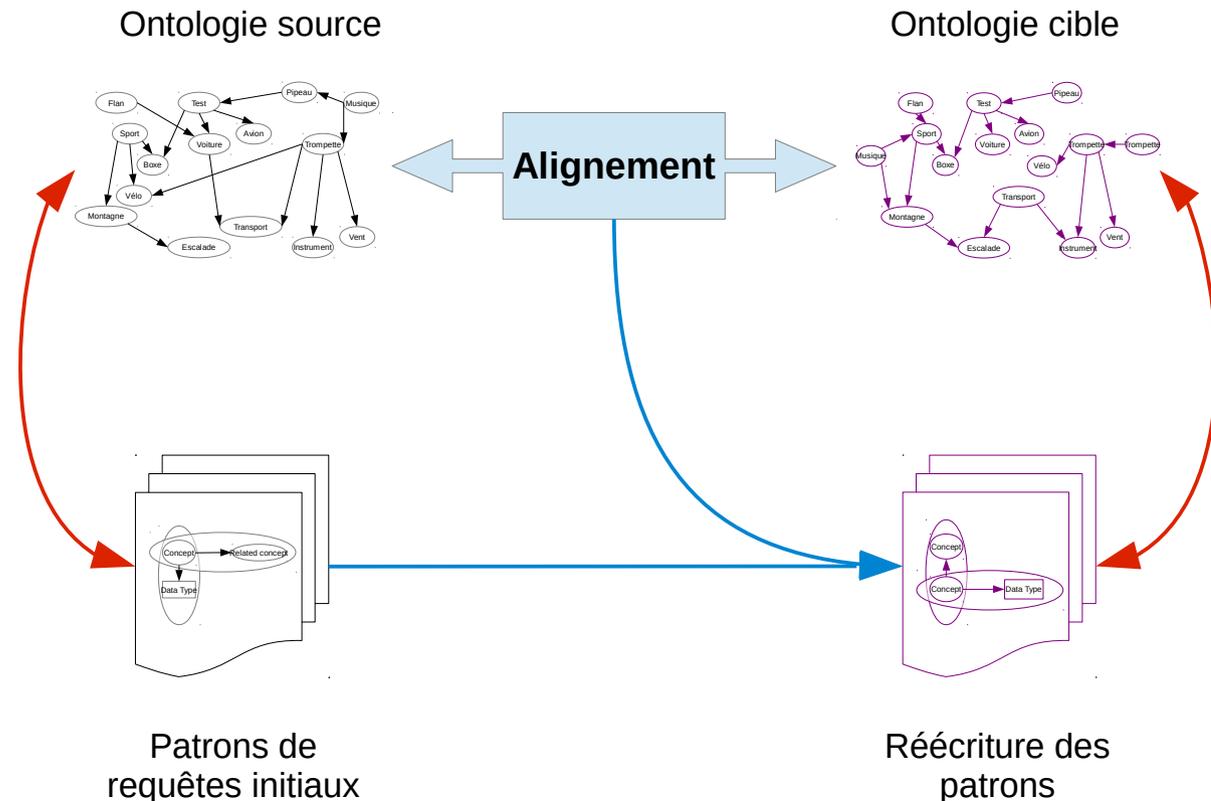
■ Problématique:

- Patrons spécifiques à la source de données
- Patrons construits manuellement



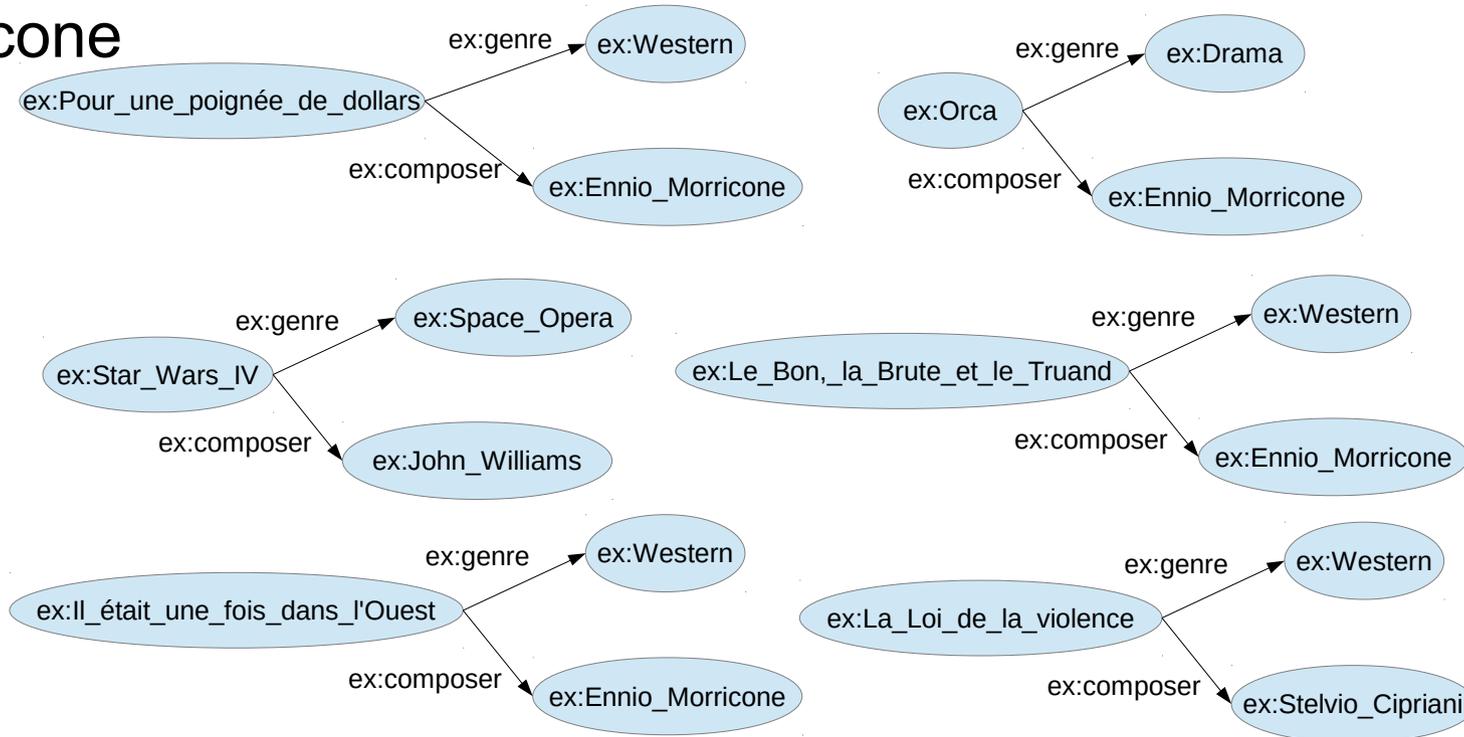
Objectif

- Proposer un mécanisme automatique pour générer des patrons de requêtes dans le système SWIP
 - Patrons initiaux
 - Alignements d'ontologies



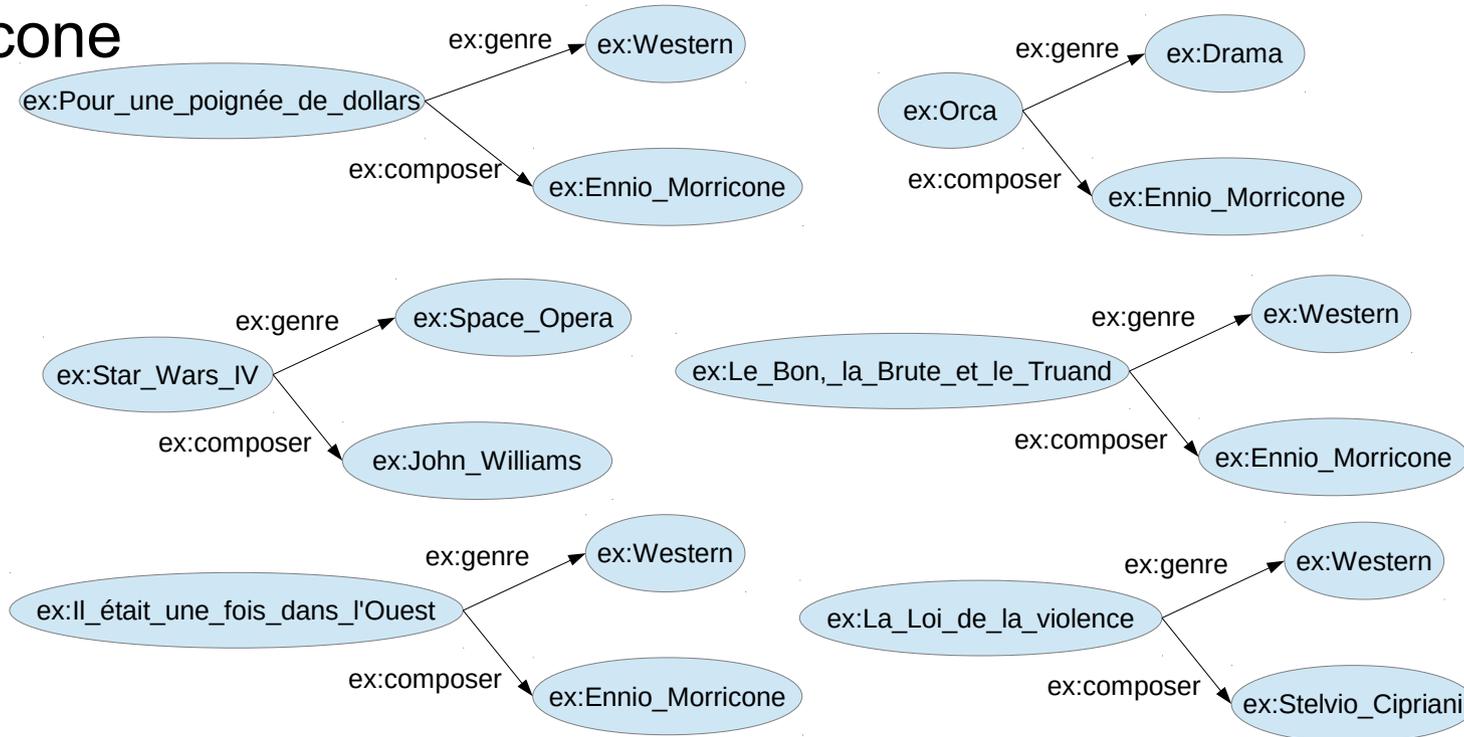
Interrogation de graphes RDF (exemple)

- Liste des Westerns dont la musique a été composée par Ennio Morricone



Interrogation de graphes RDF (exemple)

- Liste des Westerns dont la musique a été composée par Ennio Morricone

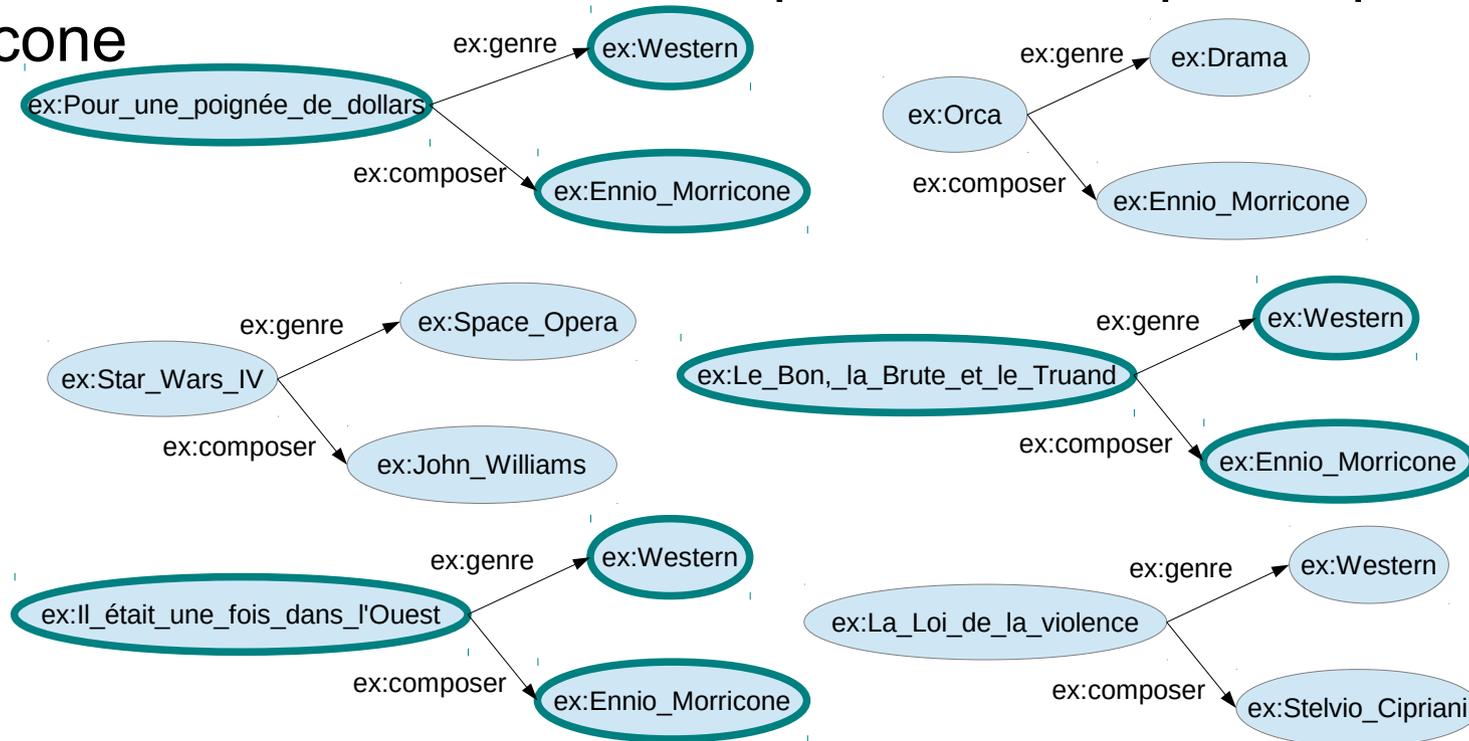


PREFIX ex: <http://example.org/>

```
SELECT ?film WHERE {  
  ?film ex:genre ex:Western .  
  ?film ex:composer ex:Ennio_Morricone .  
}
```

Interrogation de graphes RDF (exemple)

- Liste des Westerns dont la musique a été composée par Ennio Morricone



PREFIX ex: <http://example.org/>

```
SELECT ?film WHERE {  
  ?film ex:genre ex:Western .  
  ?film ex:composer ex:Ennio_Morricone .  
}
```

?film

ex:Pour_une_poignée_de_dollars

ex:Le_Bon_la_Brute_et_le_Truand

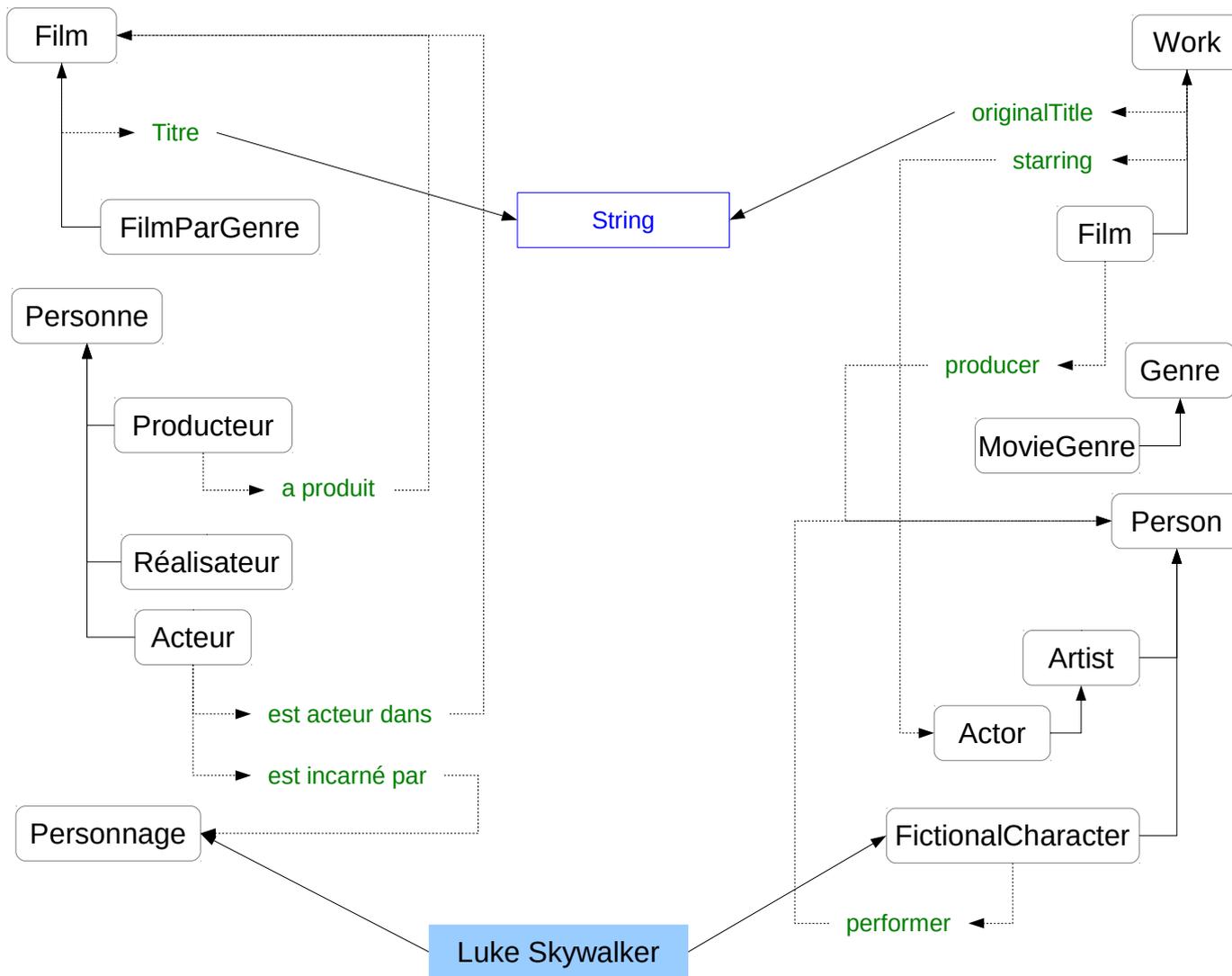
ex:Il_était_une_fois_dans_l'Ouest

Alignement d'ontologies

- **Alignement:** *Un alignement A entre deux ontologies O_1 et O_2 est un ensemble de correspondances entre des entités de O_1 et de O_2 .*
 - Un alignement est directionnel

- **Correspondance:** $\{e, e', r, n\}$
 - e et e' sont des entités de O_1 et O_2
 - r est une relation d'équivalence (\equiv), ou de subsomption (\leq, \geq)
 - n est une mesure de confiance

Alignement d'ontologies (exemple)

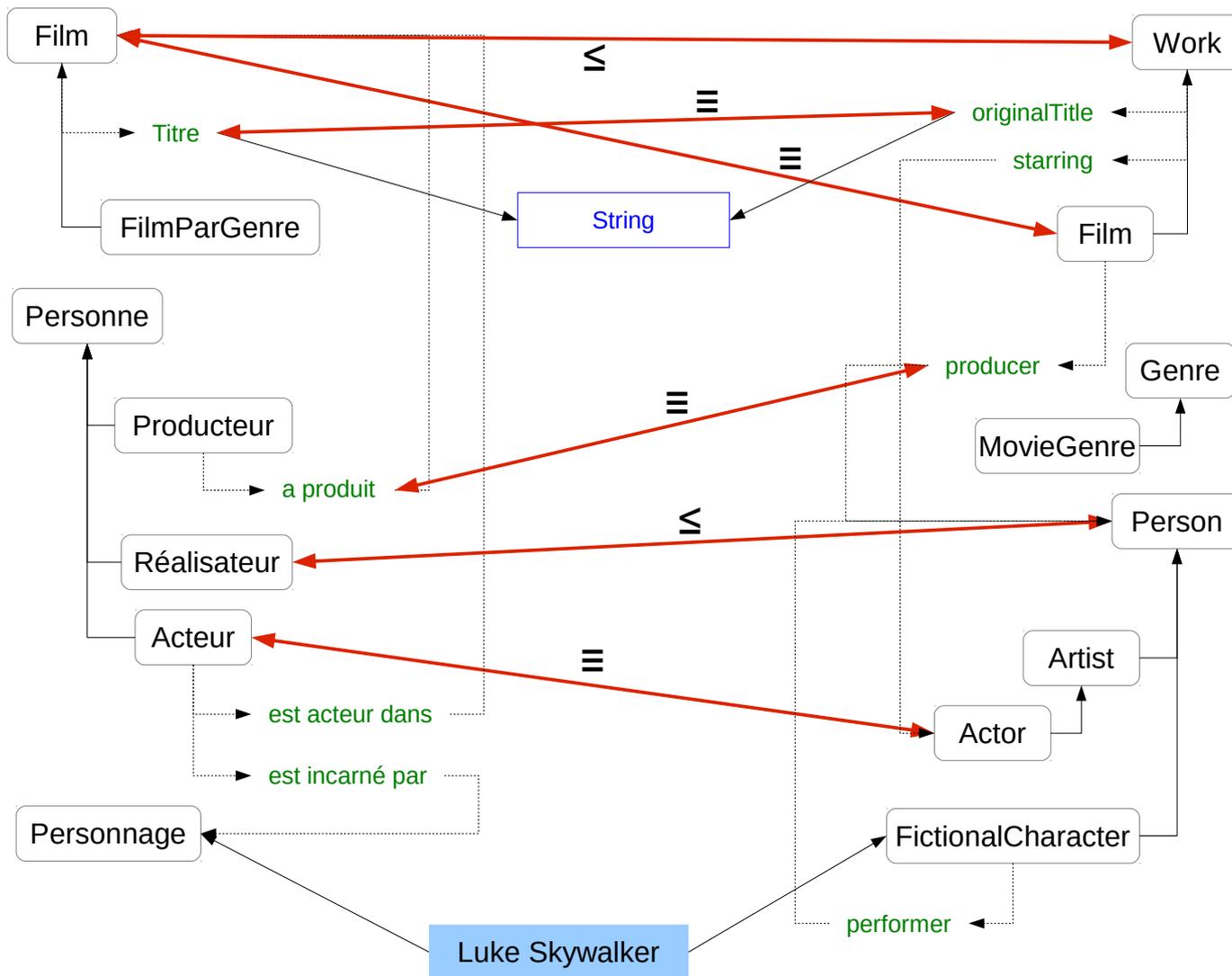


O_1 : Cinéma IRIT

O_2 : DBpedia

Réécriture de patrons de requêtes fondée sur des alignements d'ontologies

Alignement d'ontologies (exemple)



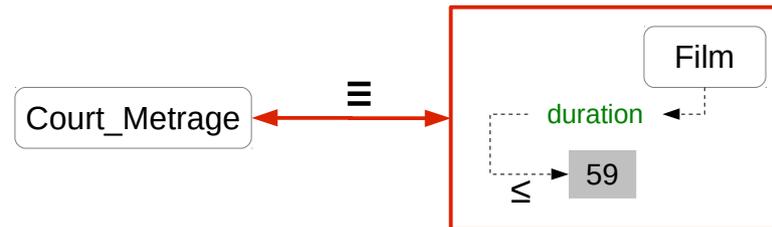
O_1 : Cinéma IRIT

O_2 : DBpedia

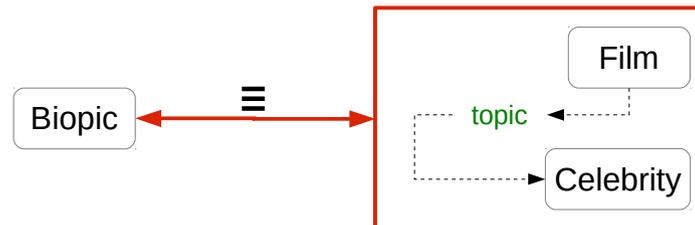
Réécriture de patrons de requêtes fondée sur des alignements d'ontologies

Correspondances complexes

- Les correspondances simples peuvent ne pas suffire:
 - Manque d'expressivité
- Correspondances **complexes**:



- $\forall x, O_1:\text{Court_metrage}(x) \equiv O_2:\text{Film}(x) \wedge O_2:\text{duration}(x, y) \wedge y \leq 59$



- $\forall x, O_1:\text{Biopic}(x) \equiv O_2:\text{Film}(x) \wedge O_2:\text{Celebrity}(y) \wedge O_2:\text{topic}(x, y)$

- **On peut passer de DL à FOL, et inversement (si le fragment DL est respecté)**

Alignement d'ontologies (Etat de l'Art)

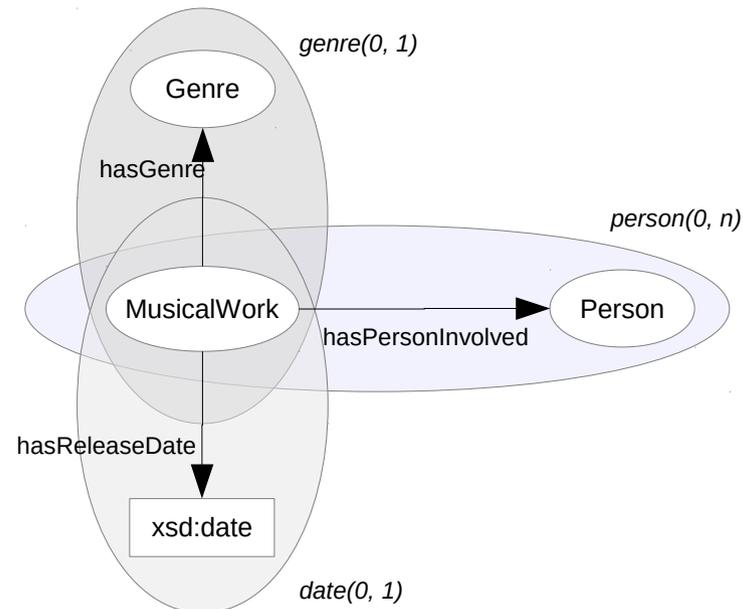
- Classification générale des approches pour les correspondances simples [Rahm & Bernstein, 2001]:

Méthodes terminologiques	Labels, commentaires
Méthodes sémantiques	Interprétation logique
Structure interne	Ensemble des valeurs possibles des propriétés (attributs et relations). Propriétés transitives, symétriques, etc.
Structure externe	Position des entités dans les hiérarchies
Similarité des instances	Intersection des ensembles d'instances
Approches combinées	

- Correspondances complexes:
 - Patrons de correspondances complexes [Scharffe, 2009, Ritze et al., 2009]

Systeme SWIP: principe des patrons

- Objectif: rendre plus efficace l'étape de construction finale de la requête SPARQL
- Prototype d'une famille de requêtes typiques
- Un patron est un graphe RDF



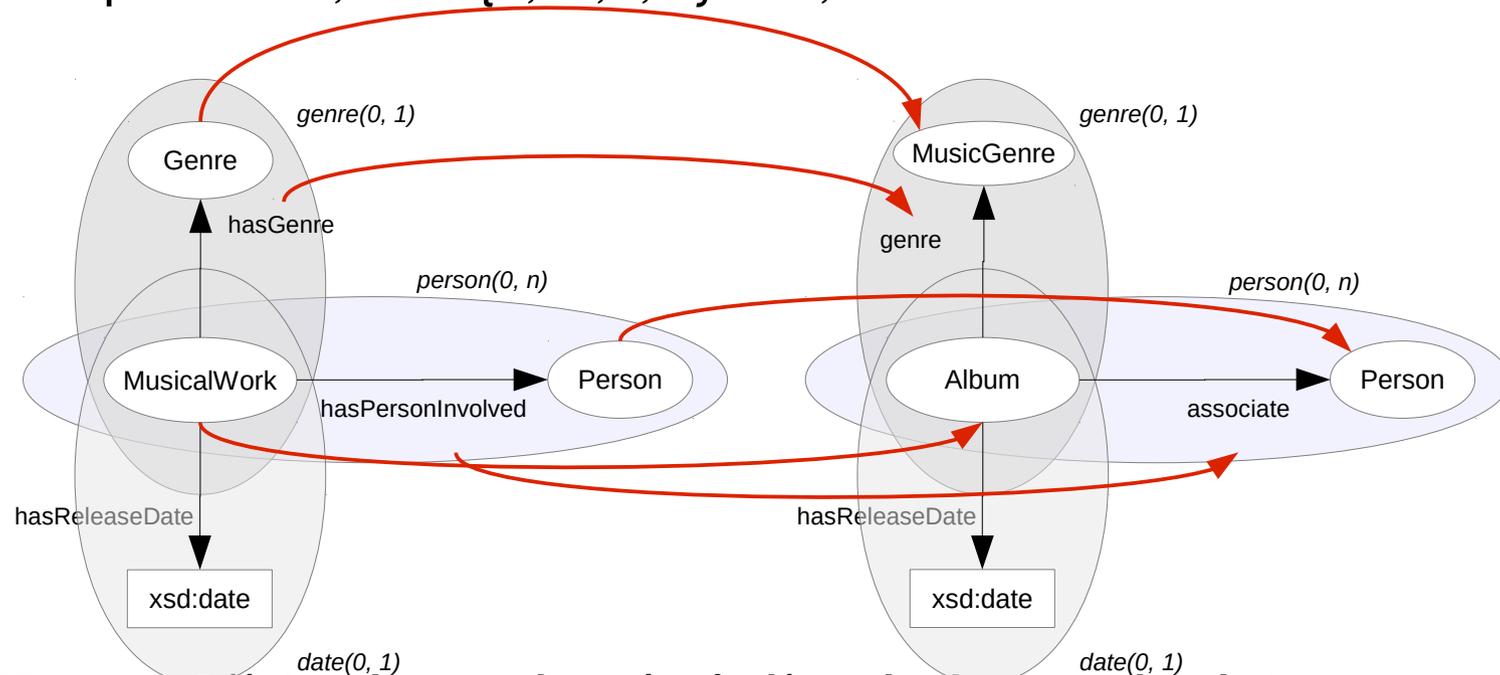
Approches pour la réécriture des patrons (i)

- **Objectif:** trouver une fonction de transformation T t.q.
 $T(A, P) = P'$, avec A un alignement, P et P' des ensembles de patrons

- **Correspondances simples:**

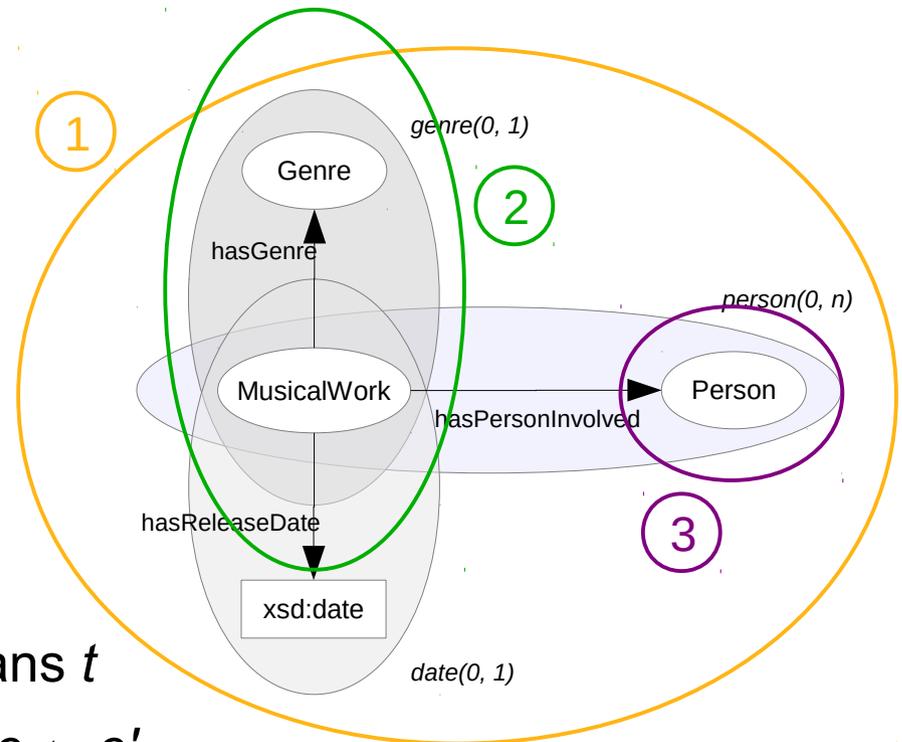
$$P' \leftarrow P$$

Pour chaque $e \in P'$, Si $\exists \{e, e', r, n\} \in A$, Alors $e \leftarrow e'$



Approches pour la réécriture des patrons (ii)

- **Correspondances complexes:** tout sous-graphe dans P a potentiellement une correspondance dans A
 - Sous-graphe de plus haut niveau: sous-patron
 - Sous-graphe de plus bas niveau: entité (concept ou propriété)
- $P' \leftarrow P$
- **Pour** chaque sous-patron $sp \in P'$
 - **Si** $\exists \{sp, sp', r, n\} \in A$
 - $sp \leftarrow sp'$
 - **Sinon Pour** chaque triplet $t \in sp$
 - **Si** $\exists \{t, t', r, n\} \in A$
 - $t \leftarrow t'$
 - **Sinon Pour** chaque entité e dans t
 - **Si** $\exists \{e, e', r, n\} \in A$ **Alors** $e \leftarrow e'$



Expérimentations – Correspondances simples (i)

- SWIP fournit des patrons pour **Music** et **Cinéma IRIT**
- Croisement avec d'autres ontologies: DBpedia, JEDFILM, Movie, BBC Program
- Outils testés: 24 systèmes d'alignement (OAEI) ne produisant que des **correspondances simples**
- Merge des alignements / paire d'ontologies

Paire d'ontologies	Alignement obtenu			Transformation des patrons	
	#correspond.	#entités sources distinctes	Couverture ontologie source	#entités remplacées	#sous-patrons remplacés
Music - DBPedia (249 * 2213)	207	168	67%	25/60 soit 41%	2/27 (5 patrons)
...					

- **Problème d'incomplétude de la transformation !**

Expérimentations – Correspondances complexes (ii)

- Aujourd'hui: très peu d'outils produisent des correspondances complexes !
- **ComplexMapping** [Ritze et al., 2009]
 - Patrons de correspondances complexes
 - Exemple de correspondance entre **Music** et **DBpedia**:

$\forall x,z$ (catalogue_number(x,z) \leftrightarrow $\exists y$ (crewMember(x,y) \wedge numberOfVisitors(y,z))) ???

Expérimentations – Correspondances complexes (iii)

- **Correspondances complexes réalisées à la main** pour Cinéma IRIT→DBpedia, et Music→DBpedia :
 - Découpage des patrons en sous-graphes (en général, **sous-graphe = sous-patron**)
 - Recherche d'un énoncé logique équivalent dans l'ontologie cible
- Exemples entre **Cinéma IRIT** et **DBpedia**:
 - $\text{Artiste} \sqcap \exists \text{estRecompenseA.CesarDuCinema} \equiv \text{Artist} \sqcap \exists \text{cesarAward}(\text{Award} \sqcap \exists \text{event.FilmFestival})$
 - $\text{Artiste} \sqcap \exists \text{acteurDoublePar.Doubleur} \equiv \text{Person} \sqcap \exists \text{dubber-} .\text{Artist}$
 - $\forall x \text{FilmLocalise}(x) \rightarrow \exists yz \text{Film}(x) \wedge \text{originalLanguage}(x, y) \wedge \text{language}(x, z) \wedge y \neq z$
 - $\text{Artiste} \sqcap \exists \text{estRecompensePour.Film} \sqsubseteq \text{Artist} \sqcap \exists \text{nominee-} .\text{Film}$
- **Résultat obtenu**: 2 nouveaux jeux de patrons pour DBpedia sur les domaines du cinéma et de la musique
 - **Taux de transformation $\approx 90\%$**
 - Music: 20/23 sous-graphes, Cinéma IRIT: 45/51 sous-graphes

Conclusion

- Approches de réécriture automatique de patrons de requêtes:
 - Correspondances simples insuffisantes
 - Correspondances complexes nécessaires
 - Alignements complexes réalisés à la main
 - Nouvelle ressource pour l'évaluation des approches d'alignements complexes
- Evolutions SWIP à court terme:
 - Support de la disjonction et des instances dans les patrons

Perspectives

- Proposer une approche pour un système d'alignement capable de générer des correspondances complexes avec les méthodes suivantes:
 - Définition de (nouveaux) patrons de correspondances complexes
 - Détection de correspondances entre instances
 - Interaction avec l'utilisateur
 - Apprentissage automatique de nouveaux patrons