Publication, partage et réutilisation de règles sur le Web de données

Oumy Seye, Catherine Faron Zucker, Olivier Corby, Alban Gaignard

Publication, partage et réutilisation de règles sur le Web de données

Oumy Seye[1], Catherine Faron-Zucker[1], Olivier Corby[2], Alban Gaignard[1]

[1] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
{seye,faron,gaignard}@i3s.unice.fr

[2] INRIA Sophia-Antipolis Méditerranée, 06900 Sophia Antipolis, France
olivier.corby@inria.fr

Résumé : L'objectif de notre travail présenté dans cet article est de favoriser la réutilisation de règles sur le Web, basée sur les principes du Web de données. En complément de données RDF, de schémas RDFS ou d'ontologies OWL, des règles peuvent être publiées et partagées sur le Web. Notre approche consiste à considérer des bases de règles comme des sources de données, représentées en RDF, qui peuvent être publiées, partagées et interrogées sur le Web de données, permettant ainsi la sélection et la réutilisation des règles pertinentes et utiles dans un contexte ou une application particuliers. Nous envisageons la sélection de règles selon des annotations qui les décrivent ou selon leur contenu, ou les deux. Nous avons implémenté et mis en œuvre notre approche avec le moteur Corese/KGRAM permettant le traitement de données centralisées ou distribuées sur le Web de données et nous avons conduit des expérimentations sur la sélection des règles de la sémantique de OWL pour des données basées sur des ontologies populaires.

Mots-clés : Web de données liées, règles, SPARQL, RDF

1 Introduction

Le partage et la réutilisation de connaissances sont les objectifs principaux du Web de données liées. Un ensemble de modèles tels que RDF, RDFS et OWL, ont été développés pour permettre aux humains et agents logiciels de publier et accéder aux données. Les schémas RDFS et les ontologies OWL sont des standards pour représenter les connaissances d'un domaine sur le Web de données. La définition de règles d'inférence constitue un moyen complémentaire ou alternatif pour capturer la sémantique sur le Web de données. En complément de données RDF, de schémas RDFS ou d'ontologies OWL, des règles peuvent être publiées et partagées sur le Web. Dans ce contexte l'objectif principal de notre travail présenté dans cet article est de favoriser la réutilisation de règles sur le Web, en se basant sur les principes du Web de données. Dans notre approche nous considérerons des bases de règles comme des sources de données particulières qui, comme toutes sources de données, peuvent être publiées, partagées et interrogées sur le Web de données, permettant ainsi la sélection et la réutilisation des règles pertinentes et utiles dans un contexte ou une application particuliers.

Un premier scénario de réutilisation de règles publiées sur le Web est le suivant. Pour traiter une source de données RDF particulière, un utilisateur souhaite sélectionner des règles selon l'organisme qui les publie, ou bien selon leur date de publication, ou leur sujet, c'est-à-dire, plus généralement, selon des critères qui peuvent être attachés aux règles dans des méta-données, et non plus seulement selon le contenu des règles.

Un second scénario, complémentaire du précédent, est relatif à la prise en compte des connaissances de domaine lors de l'exploitation d'une source de données RDF. Pour cela, l'utilisateur met en œuvre un moteur implémentant la sémantique du schéma RDFS ou de l'ontologie OWL

associés aux données, et recherche des règles de domaine relatives à ses données. Il recherche donc sur le Web, parmi toutes les règles d'inférence publiées, celles qui sont susceptibles de s'appliquer aux données qu'il souhaite exploiter, c'est-à-dire celles dont les termes sont ceux du schéma ou de l'ontologie sur lesquels reposent ses données.

Enfin, un troisième scénario, complémentaire du précédent, est la sélection la plus fine possible des règles susceptibles de s'appliquer à une source de données considérée. Pour réduire le temps des traitements ultérieurs d'une source de données, un utilisateur cherche à identifier le sous-ensemble des classes et propriétés effectivement utilisées dans les données considérées et à réduire la base de règles considérées aux seules règles susceptibles de s'appliquer sur ses données. Par exemple, dans une application sur des sources de données de grande taille, la première étape peut être la sélection des règles de la sémantique du modèle pertinentes pour l'ontologie particulière considérée. Ainsi, dans le cas d'une ontologie qui comporterait uniquement des classes primitives, beaucoup de règles de la sémantique de RDFS ou de OWL deviennent inutiles, c'est-à-dire ne s'appliquent pas, et l'économie de la tentative de leur application devient précieuse dans le cas d'une source de données de grande taille.

Dans cette perspective, pour répondre à de tels scénarios, nous proposons de considérer les règles comme des données, qu'il s'agit donc de publier dans le langage RDF, le standard du Web de données. Cette publication sur le Web permet leur réutilisation, basée sur la recherche automatique de règles pertinentes pour une application donnée ou un contexte spécifique. Cette recherche repose sur l'interrogation du contenu des règles et/ou des méta-données qui peuvent leur être associées, avec des requêtes SPARQL, le standard du Web de données pour interroger des données RDF. En d'autres termes, pour répondre au problème de la publication et la réutilisation de règles sur le Web, nous l'envisageons comme un problème classique en ingénierie des connaissances de partage et de réutilisation de connaissances et nous proposons une approche basée sur (1) la représentation en RDF à la fois du contenu et de méta-données associées aux règles, (2) leur publication sur le Web de données et (3) la construction automatique de bases de règles spécifiques à un contexte ou une application particuliers basée sur l'interrogation de sources de données RDF représentant des règles à l'aide de requêtes SPARQL.

Dans la section suivante, nous présentons le langage de règles que nous adoptons et justifions notre choix par rapport aux alternatives possibles. Dans la sections 3, nous présentons notre approche de la sélection de règles pertinentes dans un contexte donné, selon leur annotation ou leur contenu. Dans la section 4 nous présentons une implémentation de scénarios de sélection de règles dans des sources de données distribuées et leur application sur des sources de données RDF distribuées, avec le moteur sémantique Corese/KGRAM.

2 Publication de règles sur le Web de données : choix des langages SPARQL et SPIN

RIF [1] (acronyme de *Rule Interchange Format*) est le format de règles recommandé par le W3C pour échanger des règles sur le Web. Cependant, RIF reste peu utilisé. Le langage SPARQL [2] recommandé par le W3C pour l'interrogation de données RDF précède RIF et est également utilisé comme langage de règles dans de nombreux travaux sur le Web sémantique.

SELECT et ASK sont les formes de requêtes SPARQL les plus connues : une requête de la

1. http://www.w3.org/TR/rif-overview/
2. http://www.w3.org/TR/rdf-SPARQL-query/

forme ASK permet de demander si un appariement existe entre le graphe requête et le graphe RDF ; une requête de la forme SELECT permet de demander les valeurs des variables indiquées dans la clause SELECT pour lesquelles la clause WHERE de la requête s'apparie avec le graphe RDF interrogé.

La forme CONSTRUCT permet de produire un nouveau graphe RDF en remplaçant les variables du graphe de la clause CONSTRUCT par les valeurs pour lesquelles le graphe requête de la clause WHERE s'apparie avec le graphe RDF interrogé. Une telle requête peut être vue comme une règle (la prémisse représentée par la clause WHERE et la conclusion par la clause CONSTRUCT) et son traitement comme l'application d'une règle en chaînage avant pour enrichir le graphe RDF. Voici par exemple la représentation en SPARQL de la règle « tout homme est mortel » :

```
CONSTRUCT {?x rdf:type bio:Mortal} WHERE {?x rdf:type bio:Human}
```

Parmi les travaux qui utilisent SPARQL comme langage de règles, dans (Angles & Gutierrez, 2008; Polleres, 2007), les auteurs établissent une correspondance entre SPARQL et Datalog (nr-Datalog$^{\neg}$: sans récursion, avec négation), le langage de requêtes et de règles des bases de données déductives.

Dans (Schenk & Staab, 2008) les auteurs utilisent SPARQL comme langage de règles pour définir de nouvelles données RDF à partir de sources de données existantes. La forme CONSTRUCT de SPARQL est utilisée dans des réseaux de graphes pour effectuer la correspondance des patrons de graphes sur des graphes d'entrée.

Le framework R2R qui permet de publier des mappings sur le Web et de traduire des données du Web vers un schéma local est basé sur la forme CONSTRUCT de SPARQL pour exprimer des transformations de données (Bizer & Schultz, 2010).

Avec SPARQL++, (Polleres *et al.*, 2007) utilisent le langage SPARQL comme un langage de règles pour exprimer des alignements entre vocabulaires RDF et proposent pour cela certaines extensions de la forme de requêtes CONSTRUCT.

Avec SPIN [3] (acronyme de SPARQL *Inferencing Notation*), H. Knublauch invite à considérer SPARQL comme un langage de règles et de contraintes (les formes CONSTRUCT, UPDATE et ASK) et propose une notation en RDF. SPIN est une member Submission au W3C [4] depuis 2011.

SPIN est le format de règles que nous avons adopté, qui permet de publier des règles SPARQL sur le Web de données. Nous avons développé dans le moteur Corese/KGRAM un parser pour traduire des règles SPARQL en SPIN et un pretty-printer pour produire des requêtes dans la syntaxe concrète de SPARQL à partir de leur représentation en SPIN. Nous stockons la représentation SPIN de chaque règle SPARQL dans un graphe RDF nommé. Ainsi, tous les énoncés relatifs à une règle sont regroupés dans un même graphe, ce qui facilite la gestion, la recherche et la récupération de règles. Voici les représentation en SPARQL et en SPIN/RDF de la règle exprimant que si quelqu'un a un parent qui a un frère, alors celui-ci est son oncle.

```
prefix ex: <http://www.example.org/humans#>
CONSTRUCT {?x ex:hasUncle ?z}
WHERE {?x ex:hasParent ?y. ?y ex:hasBrother ?z }
```

3. http://spinrdf.org/

4. http://www.w3.org/Submission/spin-sparql/

```
@prefix sp: <http://spinrdf.org/sp#> .
@prefix ex: <http://www.example.org/humans#> .
_:b1  sp:varName "y"^^xsd:string .
_:b2  sp:varName "z"^^xsd:string .
_:b3  sp:varName "x"^^xsd:string .
[] a sp:Construct ;
   sp:templates ([ sp:subject sp:varName _:b3 ;
                   sp:predicate ex:hasUncle ;
                   sp:object _:b2 ]) ;
   sp:where ([ sp:subject sp:varName _:b3 ;
               sp:predicate ex:hasParent ;
               sp:object _:b1 ]
             [ sp:subject _:b1  ;
               sp:predicate ex:hasBrother ;
               sp:object _:b2 ]) .
```

Cependant, notre approche et notre implémentation se généralisent à tout langage de règles muni d'une syntaxe RDF ou qui peut être traduit dans le langage SPARQL. Notamment, dans (Seye *et al.*, 2012), nous avons décrit un dialecte RIF qui peut être traduit en SPARQL. Nous avons implémenté ce dialecte avec le moteur sémantique Corese/KGRAM et nous avons déployé un service en ligne pour la traduction des règles RIF-SPARQL en SPARQL. Ainsi, nous sommes capables de publier en RDF des règles RIF de ce dialecte en les traduisant d'abord dans le langage SPARQL.

Une approche similaire pourrait également être adoptée pour SWRL [5] (acronyme de *Semantic Web Rule Language*), un langage de règles relativement utilisé bien que non standardisé (*member Submission* au W3C depuis 2004). Il est muni d'une syntaxe RDF qui permettrait de publier sur le Web de données les règles écrites dans ce langage.

3 Sélection de règles : interrogation en SPARQL de leurs représentations RDF

La publication de règles en RDF sur le Web de données permet leur partage et leur réutilisation. Ces *données* RDF peuvent en effet être recherchées de façon standard, avec des requêtes SPARQL pour sélectionner des règles intéressantes à réutiliser dans un contexte spécifique.

Dans (González-Moriyón *et al.*, 2012) les auteurs ont exploré la réutilisation de règles RIF et proposé l'outil RIF Assembler qui permet de sélectionner et réutiliser des règles RIF en interrogeant les méta-données de ces règles, en les traduisant en RDF selon l'interprétation commune de RIF en RDF. Cependant, en RIF, les méta-données ne sont pas obligatoires et les règles non annotées ne peuvent pas être réutilisées avec RIF Assembler. De plus, seules les méta-données sont exploitées dans cette approche, et pas le contenu des règles.

Nous proposons une approche *générale* et *unifiée* de la réutilisation de règles SPARQL représentées en RDF, basée sur leur sélection par interrogation de leur contenu aussi bien que des métadonnées associées.

5. http ://www.w3.org/Submission/SWRL/

3.1 Sélection basée sur l'interrogation des méta-données associées aux règles

Les règles peuvent être sélectionnées en interrogeant des métadonnées qui peuvent leur être associées, dès lors qu'elles sont identifiées par un URI : nous identifions chaque règle par l'URI du graphe nommé RDF contenant sa représentation SPIN et cette URI peut être décrite en RDF. Nous répondons ainsi au premier scénario présenté en introduction.

Les métadonnées de règles peuvent par exemple contenir des informations sur la source des règles, l'auteur, le titre, le sujet, etc. Ces métadonnées lient les règles avec d'autres schémas et données du Web. Dans la recommandation RIF du W3C, il est suggéré d'utiliser le Dubling Core et des propriétés des modèles RDFS et OWL pour annoter les règles. Par exemple, la requête suivante permet de rechercher des règles sur le benfluorex parmi les données qui pourraient être publiées par l'agence nationale de sécurité du médicament et des produits de santé (ANSM), dont certaines pourraient être représentées par des règles SPARQL.

```
PREFIX  sp: <http://spinrdf.org/sp#>
PREFIX  drug: <http://www.example.org/drug#>
PREFIX  dc: <http://dublincore.org/documents/dcmi-namespace/#>
SELECT DISTINCT(kg:pprintWith(pp:spin, ?x) as ?res)
WHERE { ?x a sp:Construct ;
          dc:source <https://icrepec.afssaps.fr/Public> ;
          dc:subject drug:Benfluorex } }
```

Remarquons que dans cette requête, les règles solutions sont associées à la variable `?x` et la fonction `kg:pprintWith` appliquée sur ces solutions appelle le pretty-printer de Corese pour produire la représentation dans la syntaxe concrète de SPARQL des règles solutions, à partir de leur représentation en SPIN/RDF (Corby & Faron-Zucker, 2014). Les règles solutions sont ainsi directement utilisables par tout moteur de règles SPARQL, et en particulier le moteur de règles de Corese/KGRAM.

Le modèle Open Annotation [6] semble également bien adapté pour l'annotation de règles, qui fait une distinction entre le corps (`oa:body`) et le but (`oa:target`) d'une annotation de ressource, et permet ainsi de distinguer méta-données et description de contenu. La notion de sélecteur (`oa:Selector`) permet de décrire séparément différentes parties d'une ressource et permet, dans le cas d'une règle, de distinguer des annotations portant sur sa prémisse ou sa conclusion.

3.2 Sélection basée sur l'interrogation du contenu des règles

Publier des règles SPARQL en SPIN permet de lier la représentation de leur contenu avec d'éven-tuelles méta-données et avec d'autres données du Web : en particulier, elles sont directement liées avec des données RDF(S) ou des ontologies OWL qui peuvent être utilisées lors de la sélection de règles par interrogation de leurs contenus. Il s'agit ici de répondre aux scénarios 2 et 3 présentés en introduction.

Par exemple, la requête SPARQL suivante permet de sélectionner toutes les règles dont la prémisse contient des propriétés représentant des relations familiales.

6. http ://www.openannotation.org/spec/core/

```
PREFIX sp: <http://spinrdf.org/sp#>
PREFIX ex: <http://www.example.org/humans#>
SELECT DISTINCT (kg:pprintWith(pp:spin, ?x) as ?res)
WHERE { ?x a sp:Construct
        ?x sp:where ?m
        ?m (!sp:void)+ ?s
        ?s sp:predicate ?z
        ?z rdfs:subPropertyOf* ex:hasFamilyRelationShip }
```

Plus généralement, il est intéressant de pouvoir sélectionner des règles dans le contenu desquelles apparaissent des classes ou propriétés appartenant à une ontologie donnée. En effet, étant donnée une source de données RDF, il est inutile de chercher à appliquer des règles qui n'utilisent pas le même vocabulaire : elles ne s'appliqueraient pas sur les données. Dans le cas de sources de données de grande taille, un tel filtrage des règles susceptibles de s'appliquer peut devenir crucial pour la faisabilité des inférences. La requête suivante permet de sélectionner les règles qui utilisent des termes appartenant à un schéma RDFS chargé.

```
PREFIX sp: <http://spinrdf.org/sp#>
SELECT DISTINCT (kg:pprintWith(pp:spin, ?x) as ?res)
WHERE {
   SELECT DISTINCT ?resource
   WHERE {
      { ?resource rdf:type rdfs:Class }
       UNION
      { ?resource rdf:type rdf:Property }
     ?x a sp:Construct .
     ?x sp:where ?m .
     ?m (! sp:void)+ ?s .
     ?s ?p ?resource  } }
```

Une autre sélection, plus restrictive, pourrait être implémentée par une requête recherchant les règles pour lesquelles *toutes* les classes et propriétés apparaissant dans la prémisse appartiennent à un vocabulaire donné.

De telles présélections de règles ne remplacent pas les techniques d'optimisation des raisonneurs basées sur l'analyse des dépendances telle que celle présentée dans (Baget, 2004). Il s'agit de construire pour une ontologie une base des règles pertinentes. Nous entendons par règles pertinentes celles susceptibles de s'appliquer sur les données et donc de produire de nouvelles données inférées. Il s'agit ici des règles contenant au moins un terme de l'ontologie. Lors de l'application de la base de règles ainsi créée à une source de données particulière, les techniques d'optimisation des raisonneurs basées sur l'analyse des dépendances de règles peuvent s'appliquer.

3.3 Sélection basée sur l'ajustement de la sémantique du modèle RDFS ou OWL à la sémantique du vocabulaire des données

Dans une application sur des sources de données de grande taille, une première étape de sélection des règles de la sémantique du modèle pertinentes pour l'ontologie particulière considérée peut être précieuse. Il s'agit ici d'un cas particulier du scénario 3. Par exemple dans le cas d'une ontologie ne comportant que des classes RDFS primitives, la plupart des règles de la

sémantique de RDFS ou celles de OWL ne s'appliqueront pas. Or la connaissance a priori des règles pertinentes peut être cruciale dans le cas du traitement d'une source de données de grande taille.

Nous avons implémenté la sémantique de RDFS en écrivant une base de 14 règles et la sémantique de OWL 2 RL en construisant une base de 71 règles d'inférence. Nous avons défini deux requêtes SPARQL génériques pour sélectionner les règles SPIN de la sémantique de RDFS ou de OWL 2 RL. Ces deux requêtes sélectionnent les règles dont la prémisse contient des ressources appartenant au méta-modèle de RDFS et/ou de OWL. Voici une version simplifiée d'une telle requête :

```
SELECT DISTINCT  (kg:pprintWith(pp:spin, ?r1) as ?res)
WHERE {
    # ?resource matches URIs in the OWL or RDFS meta-model
    SELECT DISTINCT ?resource
    WHERE {
      GRAPH ?g {
        ?o a owl:Ontology
        { ?resource ?p ?y} union {?x ?resource ?y} union {?x ?p ?resource}
        filter ( ?resource in (rdf:Property, rdf:type)
                    || strstarts(?resource, owl:)
                    || strstarts(?resource, rdfs:) )
      }
    }
    # ?r1 contains a resource from the above ontology
    ?r1 a sp:Construct
    ?r1 sp:where ?w1
    ?w1 (! sp:nil)+ ?x
    ?x ?p ?resource
    VALUES ?p { sp:subject sp:predicate sp:object }
}
```

La requête finale contient également les règles qui dépendent des précédentes, c'est-à-dire que l'application des précédentes peut rendre à leur tour déclenchables : la prémisse de ces dernières peut être appariée avec la conclusion de l'une des premières. La requête de sélection complète comporte 80 lignes que nous ne reproduisons pas ici.

Nous montrons dans la section suivante que la présélection de règles de la sémantique de RDFS ou de OWL pertinentes pour un schéma ou une ontologie et donc susceptibles d'être appliquées sur une source de données reposant sur ce vocabulaire peut permettre de réduire considérablement le temps d'application des règles sur les données interrogées.

3.4 Expérimentations

Nous avons construit des bases de règles SPIN/RDF implémentant la sémantique de RDFS et OWL 2 RL et nous montrons dans cette partie comment sélectionner dans ces bases les règles pertinentes pour des ontologies populaires du Web de données et quelle réduction du coût des inférences cela représente lors de l'application de ces bases de règles à une source de données basée sur une de ces ontologies.

3.4.1 Sélection des règles de la sémantique de RDFS et OWL pertinentes pour une ontologie donnée

FOAF

L'ontologie FOAF [7] permet de décrire des personnes et leurs relations. En utilisant la requête de sélection présentée dans la section 3.3, nous sélectionnons 42 règles pertinentes pour cette ontologie parmi les 71 règles de la sémantique de OWL 2 RL. Pour implémenter la sémantique de RDFS, la requête idoine sélectionne 10 règles parmi les 14 de la base complète.

DBpedia

L'ontologie DBpedia [8] est une ontologie générale pour représenter en RDF des données issues de Wikipedia. En utilisant la même requête de sélection, 40 règles de la sémantique de OWL 2 RL sont sélectionnées parmi les 71 initiales comme pertinentes pour cette ontologie. Pour implémenter la sémantique de RDFS, 6 règles sont extraites parmi les 14 règles de la base complète.

INSEE

L'ontologie géographique de l'INSEE [9] permet de décrire les données issues du Code officiel géographique (COG) concernant notamment les régions, les départements, les arrondissements, les cantons et les communes. Pour cette ontologie, la phase de sélection de règles permet de réduire l'ensemble de règles à appliquer à 52 règles de la sémantique de OWL 2 RL pertinentes parmi les 71 de la base complète.

Pour chacune de ces ontologies, l'application des règles d'inférence RDFS et OWL 2 RL, avec et sans sélection de règles, engendrera le même nombre de triplets inférés. Cependant, la phase de sélection de règles pertinentes permet de réduire significativement le temps de calcul de ces inférences. La section suivante illustre plus précisément les gains obtenus.

3.4.2 Mesure de gain de temps lors de l'application de règles pré-sélectionnées

Dans ces expérimentations, nous nous appuyons sur la base des 71 règles SPARQL que nous avons écrites pour implémenter la sémantique de OWL 2 RL, le jeu de données RDF DBpedia-person, et le moteur de règles en chainage avant de Corese/KGRAM. Nous avons mesuré le temps d'exécution de l'ensemble des règles, avec et sans sélection, sur une machine dotée de 32 Go de RAM et de deux CPUs Intel quad-core cadencés à 2.2 GHz.

Le tableau 1 illustre les temps moyens d'application des règles sur cinq exécutions. Ces résultats montrent que la sélection de règles apporte un gain de temps de plus de 22% lors de l'application des règles OWL 2 RL sur les données DBpedia-person (1,7 million de triplets).

7. `http://xmlns.com/foaf/spec`
8. `http://downloads.dbpedia.org/3.9/dbpedia_3.9.owl.bz2`
9. `http://rdf.insee.fr/def/geo/insee-geo-onto.ttl`

Données DBpedia-person	*Toutes les règles*	*Règles pertinentes*
Nombre de règles appliquées	71	40
Nombre de triplets initiaux	1745628	1745628
Nombre de triplets initiaux et inférés	3835867	3835867
Temps moyen de calcul des inférences (s)	852,3	659,5
Ecart-type (s)	6,9	15,4

TABLE 1 – Réduction du temps de calcul des inférences OWL 2 RL lorsque les règles sont pré-sélectionnées.

4 Sélection et application de règles distribuées sur des données distribuées

Nous avons mis en œuvre notre approche de publication et de réutilisation de règles en utilisant le moteur sémantique Corese/KGRAM qui permet d'interroger des sources de données distribuées et d'appliquer des règles d'inférence sur celles-ci.

4.1 Recherche sémantique en environnement distribué

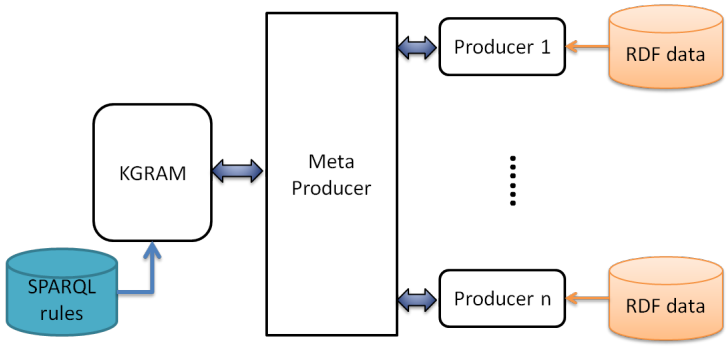
Corese/KGRAM est un moteur sémantique conçu pour l'interrogation de graphes RDF (Corby *et al.*, 2004). Corese/KGRAM implémente le langage de requête SPARQL 1.1 et permet donc d'interroger des données RDF avec des requêtes de la forme SELECT ou ASK mais aussi d'effectuer des mises à jour sur les données RDF avec des requêtes de la forme CONSTRUCT ou des opération INSERT/DELETE (SPARQL UPDATE).

Corese/KGRAM repose sur une interface `Producer` pour l'énumération de triplets RDF. Dans le cas où les données sont réparties, un `Metaproducer` itère sur les sources disponibles et énumère, en parallèle et de manière transparente, des triplets provenant de ces sources. Finalement, une interface d'invocation à distance permet la fédération de sources de données RDF distribuées sur le Web de données (Corby *et al.*, 2012).

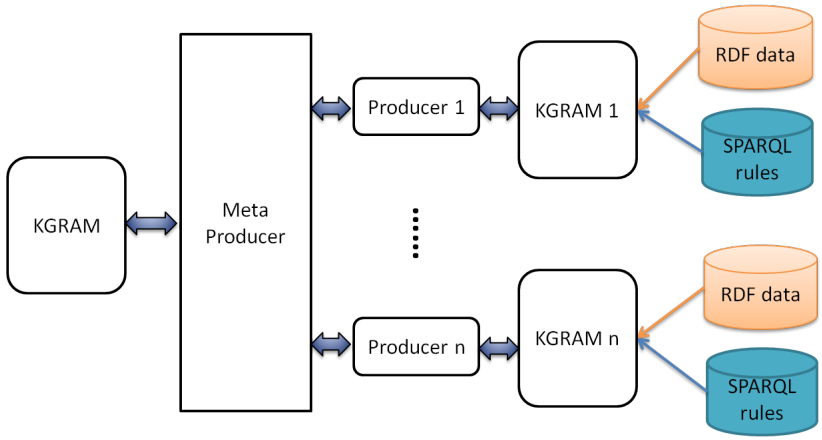
4.2 Application de règles d'inférence en environnement distribué

Avec son méta-producteur, Corese/KGRAM peut s'appuyer directement sur son moteur de règles pour raisonner sur des données multi-sources. La Figure 1 illustre deux scénarios mettant en jeu des règles et des données réparties. Le premier scénario montre l'application d'une base de règles centralisée, sur des données distribuées. Le second scénario illustre l'application de règles d'inférences elles-mêmes réparties, sur des données également réparties. Ce scénario fait sens lorsque un ensemble de règles est spécifique à un jeu de données, mais peut également s'appliquer sur d'autres sources de données.

De plus, comme nous représentons les règles en SPIN/RDF, la sélection de règles en amont de leur application peut également se faire dans un environnement distribué (scénarios 2 de la Figure 1). Des sources de données SPIN/RDF distribuées sont interrogées pour sélectionner des règles pertinentes et construire une base de règles à appliquer à des sources de données distribuées.



Scenario 1: a centralized SPARQL rule base and distributed RDF sources



Scenario 2: distributed RDF sources with associated SPARQL rule bases

FIGURE 1 – Scénarios de distribution des données et des règles

4.3 Preuve de concept

En combinant ainsi la sélection des règles représentées en SPIN/RDF et leur application à des sources de données RDF, Corese/KGRAM permet de répondre à des scénarios complexes du Web de données, intégrant la sélection et l'application de règles sur des données du Web.

Nous avons mené des premières expériences sur les données du tutoriel de Corese/KGRAM. Nous avons traduit (en utilisant notre pretty printer SPIN) la base des 25 règles SPARQL du tutoriel en un ensemble de 25 graphes SPIN/RDF que nous avons mis en ligne. Nous avons divisé les données RDF en deux sources de données RDF que nous avons mises en ligne : `human1.rdf` et `human2.rdf`. Nous avons sélectionné un *sous-ensemble* du schéma RDFS `human.rdfs` que nous avons mis en ligne dans une source de données RDFS supplémentaire `test.rdfs`.

Nous avons alors utilisé Corese/KGRAM pour interroger les 25 sources données SPIN/RDF et sélectionner les seules règles dont l'hypothèse fait référence à des ressources décrites dans le schéma `test.rdfs`.

Enfin, nous avons centralisé les 7 règles ainsi sélectionnées, étant donné le sous-ensemble de l'ontologie où nous nous limitons, et nous avons utilisé le moteur de règles de Corese/KGRAM pour appliquer ces règles sur les données RDF distribuées entre les sources `human1.rdf` et `human2.rdf`.

5 Conclusion et perspectives

Dans cet article, nous avons présenté une approche pour la publication, le partage et la réutilisation de règles d'inférence sur le Web de données liées. Nous avons choisi des règles SPARQL que nous exploitons dans leur syntaxe SPIN/RDF. Nous avons montré à travers l'utilisation du moteur sémantique Corese/KGRAM que la publication et la réutilisation de règles peuvent ainsi reposer de manière unifiée sur les modèles et techniques du Web de données. La réutilisation de règles repose sur l'interrogation de leur description RDF dans le langage SPARQL. Nous avons présenté différents scénarios de construction automatique de bases de règles pertinentes dans un contexte donné ; nous avons conduit des expérimentations sur la sélection des règles de sémantique de RDFS et de OWL 2 RL pertinentes pour des vocabulaires populaires du Web de données et montré le gain en temps que permet une telle sélection.

Le moteur Corese/KGRAM permet d'implémenter les différents scénarios d'interrogation et d'application de règles multi-sources. Nous avons précisé plusieurs scénarios typiques de réutilisation de règles avec des données liées et nous les avons expérimentés avec un jeu de données de taille réduite. Cependant, l'application de règles d'inférence sur des sources de données distribuées peut s'avérer extrêmement coûteuse. Par exemple, par nature, la base de règles OWL 2 RL est générique. Elle met en jeu certaines règles qui sont très peu sélectives, et qui mènent, dans le cas de sources de données conséquentes à des communications réseau très coûteuses.

Dans la continuité de ces travaux, nous envisageons d'étudier la sélectivité des règles et l'optimisation du moteur de règles (mise en cache) pour proposer des inférence sur des sources multiples de données et de règles dans des temps raisonnables.

Dans la continuation de ces travaux, nous avons commencé à nous intéresser à la traçabilité des règles lorsqu'elles sont sélectionnées sur le Web de données et la mise à jour de bases de règles selon leur contenu ou à les métadonnées qui leur sont associées. Il peut s'agir de modifier

les règles, par exemple en les généralisant en remplaçant une classe ou une propriété par une classe ou une propriété plus générale, de supprimer des règles, d'ajouter des annotations.

Remerciements

Nous remercions Maxime Lefrançois pour l'écriture de la base de règles SPARQL implémentant la sémantique de OWL 2 RL.

Références

ANGLES R. & GUTIERREZ C. (2008). The Expressive Power of SPARQL. In *Proc. of the 7th Int. Semantic Web Conf., ISWC 2008, Karlsruhe, Germany*, volume 5318 of *LNCS*, p. 114–129 : Springer.

BAGET J. F. (2004). Improving the forward chaining algorithm for conceptual graphs rules. In A. PRESS, Ed., *Proc. of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning, KR 2004, Whistler, Canada*, p. 407–414.

BIZER C. & SCHULTZ A. (2010). The r2r framework : Publishing and discovering mappings on the web. In *Proc. of the 1st Int. Workshop on Consuming Linked Data, COLD 2010, Shanghai, China* : CEUR-WS.org.

CORBY O., DIENG-KUNTZ R. & FARON-ZUCKER C. (2004). Querying the semantic web with the corese search engine. In *Proc. of the 16th European Conf. on Artificial Intelligence, ECAI 2004, Valencia, Spain*, p. 705–709 : IOS Press.

CORBY O. & FARON-ZUCKER C. (2014). SPARQL Template : un langage de Pretty Printing pour RDF. In *Actes des 25èmes Journées francophones d'Ingénierie des Connaissances, IC 2014, Clermont Ferrand, France*.

CORBY O., GAIGNARD A., FARON-ZUCKER C. & MONTAGNAT J. (2012). KGRAM Versatile Data Graphs Querying and Inference Engine. In *Proc. of IEEE/WIC/ACM Int. Conf. on Web Intelligence, WI 2012, Macau, China* : IEEE Computer Society.

GONZÁLEZ-MORIYÓN G., POLO L., BERRUETA D., TEJO-ALONSO C. & IGLESIAS M. (2012). Assembling rule mashups in the semantic web. In *Proc. of the 9th Extended Semantic Web Conf., ESWC 2012, Heraklion, Crete, Greece*, volume 7295 of *LNCS*, p. 590–602 : Springer.

POLLERES A. (2007). From SPARQL to rules (and back). In *Proceedings of the 16th Int. Conf. on World Wide Web, WWW 2007, Banff, Alberta, Canada*, p. 787–796 : ACM.

POLLERES A., SCHARFFE F. & SCHINDLAUER R. (2007). SPARQL++ for mapping between RDF vocabularies. In *Proc. of the 6th Int. Conf. on Ontologies, DataBases, and Applications of Semantics, ODBASE 2007, Vilamoura, Portugal*, volume 4803 of *LNCS*, p. 878–896 : Springer.

SCHENK S. & STAAB S. (2008). Networked Graphs : a Declarative Mechanism for SPARQL rules, SPARQL Views and RDF Data Integration on the Web. In *Proc. of the 17th Int. Conf. on World Wide Web, WWW 2008, Beijing, China*, p. 585–594 : ACM.

SEYE O., FARON-ZUCKER C., CORBY O. & FOLLENFANT C. (2012). Bridging the Gap between RIF and SPARQL : Implementation of a RIF Dialect with a SPARQL Rule Engine. In *Proc. of Artificial Intelligence meets the Web of Data Workshop at ECAI 2012, Montpellier, France*.