

Ré-écriture de requêtes dans un système d'intégration sémantique

Cheikh Niang^{*,**}, Béatrice Bouchou^{*}, Moussa Lo^{**}, Yacine Sam^{*}

^{*}Université François Rabelais Tours, Laboratoire d'Informatique
3 place Jean Jaurès, 41000 Blois — {prénom.nom}@univ-tours.fr

^{**}Université Gaston Berger, Laboratoire d'Analyse Numérique et d'Informatique
Route de Ngalléle, BP 234 Saint-louis, Sénégal — moussa.lo@ugb.edu.sn

Résumé. Nous décrivons la deuxième phase de réalisation d'un système d'intégration qui minimise l'intervention humaine habituellement nécessaire. Après la phase de construction semi-automatique du schéma (ontologie) global décrite dans de précédents articles, nous présentons ici le processus de ré-écriture de requêtes globales en des requêtes adressées aux sources.

1 Introduction

L'intégration de données vise à combiner des données issues de sources différentes afin de permettre leur exploitation à travers une interface d'interrogation appelée schéma global. Le travail que nous présentons ici est une des phases de réalisation d'un système d'intégration de type médiation (approche non matérialisée) sémantique. Ce dernier est un triplet $\mathcal{J} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ (Lenzerini (2002)) où, dans notre proposition, le schéma global \mathcal{G} et les schémas locaux \mathcal{S} sont décrits par des ontologies. Nous construisons le schéma global semi-automatiquement à partir des schémas sources, ce qui permet de définir naturellement un mapping GAV. Pour autant, comme montré dans Niang et al. (2011), notre construction incrémentale de \mathcal{G} induit que l'ajout et le retrait de sources ont un coût de mise à jour de \mathcal{G} très limité. Nous montrons ici qu'en plus cela n'a *aucun effet sur le mapping* que nous définissons, car ce mapping *n'est calculé qu'au moment où une requête doit être évaluée*, en fonction de l'état courant de \mathcal{G} et *uniquement pour ce qui concerne la requête*.

Notre contribution vise une intégration dynamique au sein de communautés web. Une source peut rejoindre le système intégré le temps d'échanger ses données ou de contribuer à une tâche particulière et peut se retirer à tout moment. Les collaborations qui se multiplient dans de nombreux domaines suscitent la constitution de solides ontologies de référence, publiquement accessibles. C'est dans ce cadre que nous avons mis au point dans Niang et al. (2011) notre construction du schéma global, en exploitant une ontologie de référence. Ce schéma global est une TBox spécifiée en *DL-Lite_A*, une logique de description (LD) (Calvanese et al. (2007)) connue pour traiter des requêtes dans un système d'intégration avec une complexité LOGSPACE en la taille des données interrogées. Nous exploitons dans cet article cette LD pour la **ré-écriture** d'une requête q , exprimée dans les termes de \mathcal{G} , en des sous-requêtes ciblant des sources susceptibles d'y répondre.

2 Processus de ré-écriture de requêtes

Notre proposition dans Niang et al. (2011) pour la construction semi-automatique d'une ontologie globale pouvant servir de support d'intégration dans un système de médiation, synthétisée dans la figure 1, est scindée en deux phases. D'abord, une ontologie intermédiaire appelée *Accord* (notée A dans la figure 1) est générée pour chaque source. Elle décrit la partie de l'ontologie locale concernée par le processus d'intégration. Cette phase s'appuie sur des techniques d'appariement d'ontologies : ontologie source OL et ontologie de référence OM . Ensuite, chaque Accord est automatiquement et incrémentalement concilié dans l'ontologie globale OG . La conciliation est faite en liant sémantiquement les concepts locaux via une taxonomie calculée à partir de l'ontologie de référence OM . Les sources restent ainsi indépendantes les unes vis-à-vis des autres tout en étant liées sémantiquement dans l'ontologie globale.

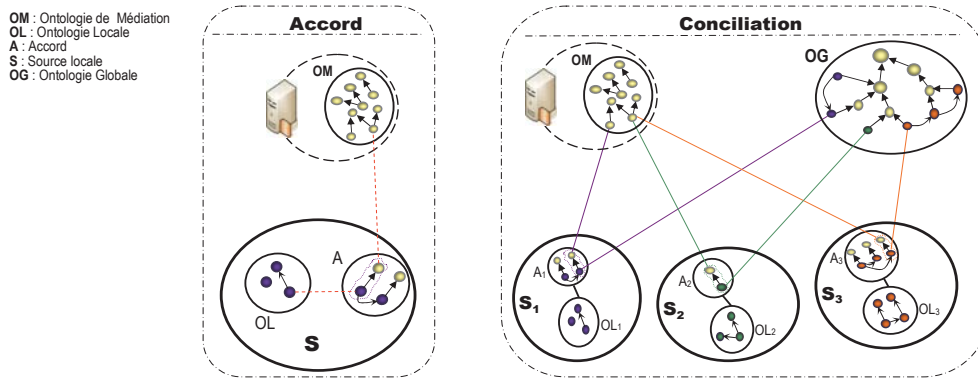


FIG. 1 – Processus de construction de l'ontologie globale.

Notre proposition a été d'abord construite dans le cadre du projet *SIC-Sénégal* ayant pour but de permettre à plusieurs partenaires travaillant sur le bassin du fleuve Sénégal de partager et intégrer leurs données, entre autres agricoles. Dans l'exemple donné dans le tableau 1¹, l'ontologie globale \mathcal{G} est représentée sous forme d'une TBox \mathcal{T}_g telle que $\mathcal{T}_g = \langle \{\mathcal{T}_{a_i}\}, \mathcal{T}_m \rangle$, où (i) $\{\mathcal{T}_{a_i}\}$ est l'ensemble des TBox des accords A^{s_i} construits à partir des ontologies locales des sources S_i , et (ii) \mathcal{T}_m est calculée à partir de l'ontologie de référence pour concilier les différents \mathcal{T}_{a_i} dans \mathcal{T}_g . Ce tableau 1 montre un extrait de \mathcal{T}_g conciliant \mathcal{T}_{a_1} et \mathcal{T}_{a_2} , des deux accords A^{s_1} et A^{s_2} . Les expressions de \mathcal{T}_{a_1} et \mathcal{T}_{a_2} expriment le fait que les sources S_1 et S_2 , à partir desquelles ont été respectivement construits A^{s_1} et A^{s_2} , détiennent des informations sur (1) des quantités de production de variétés de culture, ici : tomate, oignon et sorgho, et (2) des statistiques agricoles concernant des tomates et du riz paddy, lesquels ont un attribut prix. Les expressions de \mathcal{T}_m , située en bas du tableau, concilient \mathcal{T}_{a_1} et \mathcal{T}_{a_2} par le fait que, par exemple : oignon de \mathcal{T}_{a_1} et tomate de \mathcal{T}_{a_2} sont des légumes, sorgho de \mathcal{T}_{a_1} et riz de \mathcal{T}_{a_2} sont des céréales, les céréales et les légumes sont des produits végétaux, mais sont disjoints.

1. Faute d'espace, nous n'allons pas introduire ni la syntaxe ni la sémantique de la LD utilisée (i.e., *DL-Lite_A*). Le lecteur peut se référer à Calvanese et al. (2006) ou Niang et al. (2011) pour cela.

$\mathcal{T}_{a_1}(\mathcal{A}_1)$	$\mathcal{T}_{a_2}(\mathcal{A}_2)$
$\exists \text{ concerne} \sqsubseteq \text{Produit}$	$\exists \text{ porteSur} \sqsubseteq \text{Statistique_agricole}$
$\exists \text{ concerne}^- \sqsubseteq \text{Varietes_culture}$	$\exists \text{ porteSur}^- \sqsubseteq \text{Tomate}$
$\text{Tomate} \sqsubseteq \text{Varietes_culture}$	$\exists \text{ porteSur}^- \sqsubseteq \text{Riz_paddy}$
$\text{Oignon} \sqsubseteq \text{Varietes_culture}$	$\text{Tomate} \sqsubseteq \delta(\text{prix})$
$\text{Sorgho} \sqsubseteq \text{Varietes_culture}$	$\text{Riz_paddy} \sqsubseteq \delta(\text{prix})$
$\text{Varietes_culture} \sqsubseteq \delta(\text{production})$	$\rho(\text{prix}) \sqsubseteq \text{xsd:string}$
$\text{Legume} \sqsubseteq \text{Produit_vegetal}$	$\text{Cereale} \sqsubseteq \text{Produit_vegetal}$
$\text{Legume} \sqsubseteq \neg \text{Cereale}$	$\text{Riz} \sqsubseteq \text{Cereale}$
$\text{Tomate} \sqsubseteq \text{Legume}$	$\text{Sorgho} \sqsubseteq \text{Cereale}$
$\text{Oignon} \sqsubseteq \text{Legume}$	$\text{Sorgho} \sqsubseteq \neg \text{Riz}$
$\text{Tomate} \sqsubseteq \neg \text{Oignon}$	$\text{Riz_paddy} \sqsubseteq \text{Riz}$

TAB. 1 – Extrait d'une TBox globale \mathcal{T}_g conciliant les Accords de deux sources.

Considérons dans ce qui suit (i) l'ontologie globale \mathcal{G} , dont \mathcal{T}_g est celle montrée dans le tableau 1 ; (ii) les deux sources S_1 et S_2 dont les accords \mathcal{A}^{s_1} et \mathcal{A}^{s_2} sont ceux conciliés par \mathcal{T}_g , et (iii) la requête conjonctive suivante exprimée dans les termes de \mathcal{G} :

$$q(x, a, b) \leftarrow \text{Legume}(x), \text{prix}(x, a), \text{production}(x, b)$$

Cette requête demande le prix et la production de légumes. Nous décrivons dans la suite comment nous pouvons y répondre en reportant l'interrogation sur les sources S_1 et S_2 .

2.1 Définition de règles servant au calcul des mappings

En partant uniquement des informations présentes dans \mathcal{T}_{a_1} et \mathcal{T}_{a_2} des Accords \mathcal{A}^{s_1} et \mathcal{A}^{s_2} (tableau 1), ni S_1 ni S_2 ne peuvent répondre à q , ne serait-ce que parce que le prédicat *Legume* de q n'est défini dans aucun de leur vocabulaire. Cependant, si on considère \mathcal{T}_m - taxonomie située dans la partie basse du tableau 1 et reliant les concepts des \mathcal{T}_{a_i} , montrant dans notre exemple que les oignons et les tomates sont des légumes - il devient envisageable que S_1 et S_2 puissent permettre de construire une réponse pour q .

La première étape de notre processus de ré-écriture est d'utiliser les relations sémantiques définies dans $\mathcal{T}_g = \langle \{\mathcal{T}_{a_i}\}, \mathcal{T}_m \rangle$ afin de ramener les atomes d'une requête q , exprimée sur \mathcal{T}_g , vers les sources qu'ils impliquent (S_i est représentée par \mathcal{T}_{a_i}). Pour ce faire, nous définissons un ensemble de règles sémantiques qui permettent par un raisonnement par **chainage avant** de déduire de \mathcal{T}_g les expressions de mapping correspondant aux atomes de la requête. Ces expressions sont de la forme : $G(v) \supseteq S_i(\Phi(w))$, où $G(v)$ et $S_i(\Phi(w))$ sont, respectivement, la tête et le corps du mapping. La tête représente un atome dans lequel G est un prédicat pouvant être un concept/attribut/rôle atomique de \mathcal{T}_g et v un tuple de variables. Dans le corps, $\Phi(w)$ est une requête conjonctive, d'arité égale au nombre de variables de v , sur la source S_i .

Appliquées en chainage avant, les règles fournies en haut de la prochaine page permettent de calculer un ensemble M_g d'expressions de mapping pour chaque atome g de q . A chaque fois, l'atome g est utilisé comme déclencheur du raisonnement et \mathcal{T}_g comme base de faits initiale. Le résultat est un ensemble d'expressions M_g de la forme $g(v) \supseteq S_i(\Phi(w))$.

Les règles (1) à (3) correspondent au cas où l'atome est un concept atomique $A(x)$. Dans (1), $\text{concept}(A, \mathcal{T}_{a_i})$ est vrai si A apparaît en position de concept dans une expression de \mathcal{T}_{a_i} .

Ré-écriture de requêtes dans un système d'intégration sémantique

$$\begin{aligned}
A(x), \text{concept}(A, \mathcal{T}_{a_i}) &\rightarrow A(x) \supseteq S_i(A(x)) & (1) \\
A(x), (A_1 \sqsubseteq A) \in \mathcal{T}_g &\rightarrow A_1(x) & (2) \\
A(x), (A_1 \sqsubseteq A) \in \mathcal{T}_g, A_1(x) \supseteq S_i(\Phi(x)) &\rightarrow A(x) \supseteq S_i(\Phi(x)) & (3) \\
U(x, y), (A \sqsubseteq \delta(U)) \in \mathcal{T}_{a_i} &\rightarrow U(x, y) \supseteq S_i(U(x, y), A(x)) & (4) \\
U(x, y), (U_1 \sqsubseteq U) \in \mathcal{T}_g &\rightarrow U_1(x, y) & (5) \\
U(x, y), (U_1 \sqsubseteq U) \in \mathcal{T}_{a_i}, U_1(x, y) \supseteq S_i(\Phi(x)) &\rightarrow U(x, y) \supseteq S_i(\Phi(x)) & (6) \\
U(x, y), (A_1 \sqsubseteq A) \in \mathcal{T}_{a_i}, U(x, y) \supseteq S_i(U(x, y), A(x)) &\rightarrow U(x, y) \supseteq S_i(U(x, y), A_1(x)) & (7) \\
P(x, _), (\exists P \sqsubseteq A) \in \mathcal{T}_{a_i} &\rightarrow P(x, _) \supseteq S_i(P(x, _), A(x)) & (8) \\
P(_, y), (\exists P^- \sqsubseteq A) \in \mathcal{T}_{a_i} &\rightarrow P(_, y) \supseteq S_i(P(_, y), A(y)) & (9) \\
P(x, y), (P_1 \sqsubseteq P) \in \mathcal{T}_g &\rightarrow P_1(x, y) & (10) \\
P(x, y), (P_1 \sqsubseteq P) \in \mathcal{T}_{a_i}, P_1(x, y) \supseteq S_i(\Phi(x, y)) &\rightarrow P(x, y) \supseteq S_i(\Phi(x, y)) & (11) \\
P(x, _), (A_1 \sqsubseteq A) \in \mathcal{T}_{a_i}, P(x, _) \supseteq S_i(P(x, _), A(x)) &\rightarrow P(x, _) \supseteq S_i(P(_, _), A_1(x)) & (12) \\
P(_, y), (A_1 \sqsubseteq A) \in \mathcal{T}_{a_i}, P(_, y) \supseteq S_i(P(_, y), A(y)) &\rightarrow P(_, y) \supseteq S_i(P(_, y), A_1(y)) & (13)
\end{aligned}$$

Dans ce cas, S_i peut renseigner sur A , ce qu'indique l'expression de mapping en conclusion de la règle. Ainsi toutes les sources contenant ce concept seront considérées par le processus de ré-écriture. (2) et (3) permettent de rechercher des sous-concepts qui seraient dans les sources. Grâce à elles, en partant de l'atome $Legume(x)$ on peut arriver à $Tomate(x)$ de S_2 et $Oignon(x)$ et $Tomate(x)$ de S_1 . Les règles (4) à (7) peuvent se comprendre comme les trois premières mais s'appliquent à un attribut atomique $U(x, y)$. Elles servent à rechercher d'éventuels sous-attributs au sein d'une même source. La règle (7) permet, dans une même source, d'interroger l'attribut d'un concept sur ses sous-concepts. Les règles (8) à (13) concernent un rôle atomique $P(x, y)$, les 2 premières fonctionnant comme (1) pour le concept ou (4) pour l'attribut : elles s'appliquent s'il y a dans une \mathcal{T}_{a_i} une expression impliquant ce rôle. (10) et (11) permettent de rechercher des sous-rôles (dans une même source) quand (12) et (13) servent à interroger des rôles sur des sous-concepts (dans une même source). Nous donnons ci-après les mappings obtenus en appliquant les règles (1) à (13) pour chaque atome de la requête q :

$$\begin{aligned}
M_{Legume} : & \quad m_1 : Legume(x) \supseteq S_1(Tomate(x)) \\
& \quad m_3 : Legume(x) \supseteq S_1(Oignon(x)) \\
& \quad m_2 : Legume(x) \supseteq S_2(Tomate(x)) \\
M_{prix} : & \quad m_1 : prix(x, a) \supseteq S_2(prix(x, a), Tomate(x)) \\
& \quad m_2 : prix(x, a) \supseteq S_2(prix(x, a), Riz_paddy(x)) \\
M_{production} : & \quad m_1 : production(x, b) \supseteq S_1(production(x, b), Varietes_culture(x)) \\
& \quad m_2 : production(x, b) \supseteq S_1(production(x, b), Oignon(x)) \\
& \quad m_3 : production(x, b) \supseteq S_1(production(x, b), Tomate(x))
\end{aligned}$$

Les mappings de M_{prix} ne ramènent qu'à S_2 qui est capable de fournir le prix d'instances de tomate et de riz, tandis que les mappings de $M_{production}$ indiquent que la source S_1 est capable d'apporter des réponses sur la production de variétés de cultures, d'oignons et de tomates. Les réponses qui peuvent être obtenues à partir de S_1 et S_2 sont partielles puisque S_1 ne peut fournir que la production (de tomate ou d'oignon) et S_2 que le prix (de tomate ou de riz). C'est en combinant les différents mappings que l'on aura une réponse globale pour q , c'est l'objet de la prochaine sous-section.

2.2 Dépliage de requêtes globales et validation des ré-écritures obtenues

Le dépliage d'une requête $q(x) \leftarrow g_1(z_1), \dots, g_n(z_n)$ par rapport à un ensemble de mappings GAV $g_i(x_i) \supseteq q_i(x_i, y_i)$ est la requête u obtenue en remplaçant dans q chaque atome $g_i(z_i)$ par $q_i(\psi_i(x_i, y_i))$, où ψ_i est une fonction qui associe x_i à z_i , et y_i à de nouvelles variables. Chaque dépliage u sert à calculer une partie du résultat que le système intégré peut fournir pour q . L'union de ces parties constitue l'ensemble des réponses à q . L'ensemble des dépliages u de q est obtenu en calculant le produit cartésien $R = M_{g_1} \times \dots \times M_{g_n}$. Chaque requête q_r de ce produit est une ré-écriture *candidate*. En considérant les mappings obtenus précédemment pour les atomes de la requête q , résumés dans le tableau 2, certaines des 18 ré-écritures *candidates* de q sont les suivantes :

$Legume(x)$	$Prix(x,a)$	$Production(x,b)$
$S_1(Tomate(x))$	$S_2(prix(x, a), Tomate(x))$	$S_1(production(x, b), Varietes_culture(x))$
$S_1(Oignon(x))$	$S_2(prix(x, a), Riz_paddy(x))$	$S_1(production(x, b), Tomate(x))$
$S_2(Tomate(x))$		$S_1(production(x, b), Oignon(x))$

TAB. 2 – Résumé des sources pouvant répondre pour chaque atome de la requête q .

$$\begin{aligned}
q_{r_1}(x,a,b) &\leftarrow S_1(Tomate(x)), S_2(prix(x,a), Tomate(x)), S_1(production(x,b), Varietes_culture(x)) \\
q_{r_2}(x,a,b) &\leftarrow S_1(Tomate(x)), S_2(prix(x,a), Tomate(x)), S_1(production(x,b), Tomate(x)) \\
q_{r_{13}}(x,a,b) &\leftarrow S_2(Tomate(x)), S_2(prix(x, a), Tomate(x)), S_1(production(x,b), Varietes_culture(x)) \\
q_{r_{14}}(x,a,b) &\leftarrow S_2(Tomate(x)), S_2(prix(x, a), Tomate(x)), S_1(production(x,b), Tomate(x)) \\
q_{r_{15}}(x,a,b) &\leftarrow S_2(Tomate(x)), S_2(prix(x, a), Tomate(x)), S_1(production(x,b), Oignon(x))
\end{aligned}$$

Une ré-écriture *candidate* ne constitue pas forcément une ré-écriture valide de q . A titre d'exemple, $q_{r_{15}}$ qui demande pour S_1 la production de tomates et pour S_2 le prix d'oignons n'est pas valide. Les tomates et les oignons sont certes des légumes, mais l'expression $Tomate \sqsubseteq \neg Oignon$ déclarée dans la TBox \mathcal{T}_g indique que les tomates et les oignons sont disjoints.

Pour vérifier la consistance des dépliages, nous utilisons l'algorithme *Consistent* de Calvanese et al. (2007) qui construit des requêtes booléennes q_{unsat} à partir des inclusions négatives présentes dans \mathcal{T}_g , qui représentent des contraintes d'intégrité servant à assurer la consistance des réponses apportées par le système d'intégration. Considérons ci-après les requêtes q_{unsat} obtenues à partir de \mathcal{T}_g du tableau 1 et qui concernent les atomes de q :

$$\begin{aligned}
q_{unsat1} &\leftarrow Tomate(x), Oignon(x) - \text{car } Tomate \sqsubseteq \neg Oignon \text{ est dans } \mathcal{T}_g \\
q_{unsat2} &\leftarrow Tomate(x), Riz_paddy(x) - \text{inféré de } Legume \sqsubseteq \neg Cereale \\
q_{unsat3} &\leftarrow Oignon(x), Riz_paddy(x) - \text{idem}
\end{aligned}$$

Afin de vérifier la consistance d'une ré-écriture *candidate* q_r , nous remplaçons d'abord dans q_r les requêtes sur les sources par les requêtes correspondantes sur le schéma global. Dans notre contexte il suffit de retirer toute mention des sources. Soit q'_r le résultat de cette étape, il faut ensuite appliquer l'algorithme *Consistent* en évaluant les requêtes q_{unsat} sur le corps de q'_r considéré comme une instance canonique. Si le résultat de l'évaluation d'une des requêtes q_{unsat} est *true* alors la ré-écriture q_r est invalide. A titre d'exemple, soit la requête $q'_{r_{15}}$ suivante, obtenue à partir de $q_{r_{15}}$:

$$q'_{r_{15}}(x,y,z) \leftarrow Tomate(x), prix(x,y), Tomate(x), production(x,z), Oignon(x)$$

En évaluant q_{unsat1} sur $\{Tomate(a), prix(a,b), production(a,c), Oignon(a)\}$ on obtient *true*, donc q_{r15} est invalide. En appliquant ce processus sur toutes les ré-écritures *candidates* de q nous obtenons les ré-écritures valides suivantes :

$$\begin{aligned} q_{r1}(x,a,b) &\leftarrow S_1(Tomate(x), S_2(prix(x,a), Tomate(x)), S_1(production(x,b), Varietes_culture(x))) \\ q_{r2}(x,a,b) &\leftarrow S_1(Tomate(x), S_2(prix(x,a), Tomate(x)), S_1(production(x,b), Tomate(x))) \\ q_{r13}(x,a,b) &\leftarrow S_2(Tomate(x), S_2(prix(x,a), Tomate(x)), S_1(production(x,b), Varietes_culture(x))) \\ q_{r14}(x,a,b) &\leftarrow S_2(Tomate(x), S_2(prix(x,a), Tomate(x)), S_1(production(x,b), Tomate(x))) \end{aligned}$$

Ces requêtes peuvent être simplifiées en unifiant les atomes équivalents d'une même requête impliquant la même source (algorithme de test d'inclusion de requêtes de Millstein et al. (2003)). Après simplification, les requêtes q_{r2} et q_{r14} étant équivalentes à la requête q_{r1} peuvent être supprimées. La ré-écriture de q se réduit au final aux deux requêtes q_{r1} et q_{r13} simplifiées :

$$\begin{aligned} q_{r2}(x,a,b) &\leftarrow S_1(Tomate(x), production(x,b)), S_2(Tomate(x), prix(x,a)) \\ q_{r13}(x,a,b) &\leftarrow S_2(Tomate(x), prix(x,a)), S_1(Varietes_culture(x), production(x,b)) \end{aligned}$$

3 Conclusion

Dans cet article, nous avons décrit le processus de ré-écriture de requêtes exprimées dans les termes d'une ontologie globale en des sous-requêtes ciblant les sources. L'originalité de cette ré-écriture est qu'elle repose sur un calcul dynamique des mappings, dont la complexité est au pire en $O(n \times (k \times l)^2)$, où k est le nombre de \mathcal{T}_{a_i} impliqués par l'atome g , l est la taille moyenne des \mathcal{T}_{a_i} et n est le nombre d'atomes dans la requête. Ceci est possible grâce aux caractéristiques du schéma global servant de support d'interrogation, construit semi-automatiquement, pour lequel l'ajout et le retrait d'une source ont un effet limité sur le schéma.

Références

- Calvanese, D., G. D. Giacomo, D. Lembo, M. Lenzerini, et R. Rosati (2007). Tractable Reasoning and Efficient Query Answering in Description Logics : The *L-Lite* Family. *J. Autom. Reasoning* 39(3), 385–429.
- Calvanese, D., D. Lembo, M. Lenzerini, A. Poggi, et R. Rosati (2006). Linking data to ontologies : The description logic dl-litea. In *In Proc. of OWLED 2006*.
- Lenzerini, M. (2002). Data integration : A theoretical perspective. In *PODS*, pp. 233–246.
- Millstein, T. D., A. Y. Halevy, et M. Friedman (2003). Query containment for data integration systems. *J. Comput. Syst. Sci.* 66(1), 20–39.
- Niang, C., B. Bouchou, M. Lo, et Y. Sam (2011). Automatic building of an appropriate global ontology. In *ADBIS*, pp. 429–443.

Summary

We present in this article the second realization phase of our data integration system minimizing human intervention. After the proposition in a previous work of an approach allowing the construction of a global ontology (GO), we focus here on the process allowing to rewrite queries specified over the GO to queries targeting local ontologies present in the system.