

Génération automatique de correspondances sémantiques entre schémas Une approche par réécriture de requêtes

Karima Toumani, Hélène Jaudoin, Michel Schneider

▶ **To cite this version:**

Karima Toumani, Hélène Jaudoin, Michel Schneider. Génération automatique de correspondances sémantiques entre schémas Une approche par réécriture de requêtes. INFORSID, May 2007, Perros-Guirec, France. hal-01556467

HAL Id: hal-01556467
https://inria.hal.science/hal-01556467

Submitted on 5 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Génération automatique de correspondances sémantiques entre schémas

Une approche par réécriture de requêtes

Karima Toumani* — **Hélène Jaudoin** ** — **Michel Schneider** *

* *Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS)*
Université Blaise-Pascal Clermont-Ferrand II
24 avenue des Landais 63177 Aubière cedex
{ktoumani, michel.schneider}@isima.fr

** *IRISA-ENSSAT*
6, rue Kerampont, 22305 Lannion
jaudoin@enssat.fr

RÉSUMÉ. *Nous nous intéressons au problème de génération automatique de correspondances sémantiques entre des schémas de sources de données autonomes. Nous proposons pour cela d'exploiter des correspondances définies de manière centralisée entre les schémas des sources et un schéma global, pour inférer des correspondances entre les schémas des sources. Nous utilisons le cadre des logiques de description pour formaliser ce problème. Nous identifions deux sous-problèmes: le premier correspond au problème de recherche de réécritures maximalement contenues dans une requête, tandis que le second constitue une nouvelle instance du problème de la réécriture de requêtes. Pour la résolution de ce dernier, nous distinguons deux cas : les langages fermés par la négation et les langages possédant la propriété de subsomption structurelle.*

ABSTRACT. *we are interested in the problem of automatic generation of mappings between data sources schemas, starting from mappings between these schemas and a global ontology. We formalize the problem in description logics and we show that it brings back to a problem of rewriting queries using views. We identify two subproblems: the first is equivalent to a well known problem of rewriting queries using views, the second constitues a new way of rewriting queries problem. We distinguish two cases to solve the second problem.*

MOTS-CLÉS : *correspondances sémantiques de schémas, logiques de description, réécriture de requêtes, les hypergraphes*

KEYWORDS: *schema mappings, description logics, rewriting queries, hypergraphs*

1. Introduction et motivation

L'exploitation de l'information disséminée sur un réseau est devenu une préoccupation majeure de nombreux secteurs d'activités comme l'industrie, l'agriculture et le commerce [ABI 03]. Les problèmes sous-jacents de partage et d'intégration d'information ont intéressé la communauté de recherche en bases de données depuis une vingtaine d'années et continuent aujourd'hui de faire l'objet d'actives investigations [LEV 01, LEN 02, ABI 03]. Les travaux dans ce domaine ont visé à développer des techniques et des outils permettant un accès transparent à des sources de données disséminées sur un réseau. Ils ont notamment conduit à deux grandes classes de systèmes d'intégration : les systèmes de médiation basés sur le paradigme *médiateur/wrapper* et les systèmes pair à pair (P2P). Dans le premier type d'architecture, un composant central, appelé *médiateur*, joue le rôle d'interface entre les utilisateurs et les sources de données. Le médiateur est composé d'un *schéma global*, qui offre une vue unifiée des sources de données et sur lequel les requêtes sont exprimées. Bien que ces systèmes soient efficaces pour des applications comportant peu de sources de données, ils sont peu adaptés au nouveau contexte d'intégration soulevé par le web car ils reposent sur un schéma global unique. Les systèmes pair à pair sont formés d'un ensemble de sources de données autonomes (pairs) tel que chaque pair est associé à un schéma qui représente son domaine d'intérêt. Chaque schéma de pair constitue un point d'entrée dans le système pair à pair. En d'autres termes, les requêtes sont exprimées sur le schéma local du pair. La résolution des requêtes dans les systèmes d'intégration pair à pair ou médiation nécessite l'existence de mappings[1] centralisés au niveau du médiateur établis entre le schéma global et les schémas des sources de données pour l'approche médiation et entre les schémas des pairs pour l'approche pair à pair. Les mappings relatifs au pair local sont stockés au niveau du pair lui même. A partir de ces mappings, le médiateur, respectivement le pair, analyse les requêtes et les reformule en sous-requêtes exécutables par les sources ou les pairs.

L'établissement des correspondances sémantiques entre différents schémas de données est une tâche difficile. Elle constitue actuellement une des étapes les plus coûteuse lors du développement d'un système d'intégration de type médiation ou pair à pair. Dans cet article, nous nous intéressons au problème de la génération automatique de correspondances sémantiques entre des schémas de sources de données autonomes. Nous les appelons mappings *décentralisés* par opposition aux mappings centralisés stockés dans le médiateur. L'idée majeure de cet article est d'exploiter des mappings centralisés existants pour *inférer* des mappings décentralisés (c.f. figure 1). En effet, on remarque que différents efforts ont contribué à développer des mappings centralisés par exemple : *(i)* entre des schémas de sources de données et un schéma global dans le cadre des systèmes de médiation classique, ou *(ii)* entre des applications web et une ontologie globale du domaine (i.e., des ontologies qui définissent les concepts de référence dans des domaines particuliers). Concernant ce dernier cas, l'essor actuel du web, et particulièrement le web sémantique, a conduit au développement de

1. Dans cet article, nous utilisons aussi le terme mappings pour désigner les correspondances sémantiques

plus en plus d'applications qui reposent sur des ontologies globales de domaines. Par exemple, le standard **ISCO88** donne une classification des professions tandis que le consortium **RosettaNet** propose des standards pour les échanges commerciaux. L'objectif est alors d'exploiter de tels mappings centralisés vers des ontologies métiers pour découvrir des mappings décentralisés. La découverte des mappings décentralisés a deux intérêts majeurs *(i)* dépasser les limites de s'abstraire d'un schéma global unique des systèmes de médiation en les transformant en systèmes extensibles de type P2P, ou *(ii)* dans le contexte du web sémantique où des application sont décrites selon une ontologie métier, de créer un cadre d'intégration distribué. Ces mappings décentralisés peuvent constituer les entrées d'algorithmes de résolution de requêtes distribuées des systèmes pair à pair comme SomeWhere et PIAZZA.

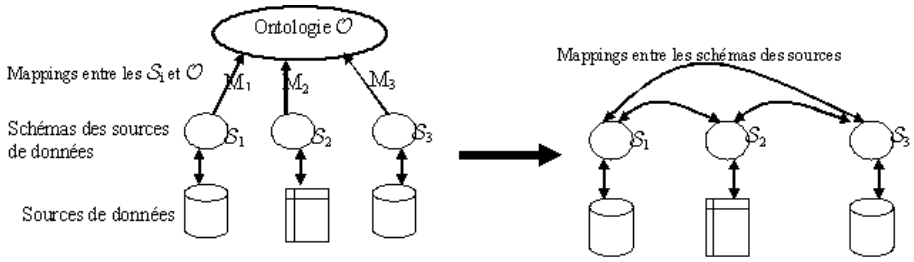


Figure 1. *Processus général*

Par exemple, considérons une ontologie[2] $\mathcal{O}$ concernant le domaine de l'enseignement et qui contient, entre autres, les deux concepts suivants :

– $Enseignant$ est une personne qui enseigne une matière,

– $TravailleurSecondaire$ est une personne qui travaille dans un établissement d'enseignement secondaire,

Supposons maintenant que trois sources de données différentes (ou pairs) P_1, P_2 et P_3 aient défini les concepts de leurs schémas respectifs en termes de l'ontologie $\mathcal{O}$ comme suit :

– le concept $Professeur$ du pair P_1 est équivalent dans l'ontologie $\mathcal{O}$ à des enseignants qui travaillent dans un établissement d'enseignement secondaire (i.e., l'intersection des concepts $Enseignant$ et $TravailleurSecondaire$ de l'ontologie $\mathcal{O}$),

– le concept $Personnel$ de P_2 est une spécialisation (i.e., sous-concept) du concept $Enseignant$ de l'ontologie $\mathcal{O}$, et

– le concept $Employé$ de P_3 est équivalent au concept $TravailleurSecondaire$ de l'ontologie $\mathcal{O}$.

2. Dans la suite de cet article, nous utilisons les termes ontologie et schéma de manière interchangeable.

A partir de ces correspondances, il est alors possible de déduire que d'un point de vue sémantique le concept $Professeur$ de P_1 est une généralisation (i.e., super-concept) de l'intersection du concept $Personnel$ du pair P_2 et du concept $Employé$ du pair P_3. Ainsi, une requête sur le pair P_1 s'intéressant, par exemple, à la liste de tous les professeurs peut être reformulée dans le cadre d'un système pair à pair en deux sous-requêtes interrogeant respectivement les pairs P_2 (pour le $Personnel$) et P_3 (pour $Employé$).

Dans cet article, nous étudions ce problème de découverte de mappings en nous appuyant sur un cadre formel des logiques de description [3]. Ces logiques constituent aujourd'hui des formalismes de représentation de connaissances importants et sont, par exemple, à la base des langages du web sémantique. Les principales contributions de notre papier sont les suivantes :

– Nous formalisons le problème de découverte des mappings décentralisés à partir d'un ensemble de mappings centralisés dans le cadre des logiques de description,

– Nous proposons une technique pour générer de façon automatique ce type de mappings. Pour cela, nous montrons que le problème de découverte de mappings se ramène à un problème de réécriture de requêtes en utilisant les vues. Plus précisément, nous identifions deux sous-problèmes : *(i)* le premier est équivalent à un problème de réécriture classique de recherche des réécritures maximalement contenues dans une requête comme étudié dans [GOA 04, BEE 97, LEV 01], et *(ii)* le second problème consiste à rechercher les réécritures minimales qui contiennent (subsument) une requête. Nous montrons que ce dernier problème constitue une nouvelle instance du problème de la réécriture de requêtes en termes de vues et nous distinguons deux cas pour sa résolution. Pour les langages fermés par la négation, le problème se ramène au problème de réécriture classique, tandis que pour les langages ayant la propriété de subsomption structurelle, une technique basée sur la théorie des hypergraphes a été proposée pour le résoudre.

Le reste de cet article est organisé de la manière suivante : la section 2 introduit les notions de base concernant les logiques de description. La section 3 montre comment le problème de la génération de mappings peut se ramener à un problème de réécriture en définissant deux types de problèmes de réécriture M_{max} et M_{min}. La résolution du problème M_{min} est détaillée dans la section 4. Finalement, dans la section 5 nous positionnons notre approche par rapport aux travaux connexes et nous concluons en donnant quelques perspectives.

2. Introduction aux logiques de description

Les logiques de description (LD) [BAA 03] sont une famille de formalismes de représentation de connaissances et de raisonnement basée sur un sous-ensemble de la logique du premier ordre. Elles fournissent un ensemble de constructeurs qui permet

3. Ces travaux sont supporté par l'ANR, ANR-05-MMSA-0007

de représenter un domaine d'application en termes de classes d'individus (prédicats unaires) appelés concepts, et de rôles (prédicats binaires entre individus). Par exemple, en utilisant le constructeur de conjonction de concept $\sqcap$, la restriction de la cardinalité minimale d'un rôle ($\geq\ nR$) où n est un entier positif et R est un nom de rôle et le quantificateur universel qualifié $\forall R.C$, où R est un rôle et C est un concept, on peut décrire le concept $Enseignant$ comme suit : $personne \sqcap \forall donne.Cours \sqcap (\geq\ 1donne)$. Ce concept dénote l'ensemble des individus du domaine qui sont des personnes et qui donnent au moins un cours.

En réalité, il existe plusieurs logiques de description différentes, chacune étant définie par l'ensemble de constructeurs autorisés dans cette logique. Par exemple, la figure 2 donne les constructeurs de deux logiques de description : $\mathcal{FL}_0$ et $\mathcal{ALN}$. $\mathcal{FL}_0$ est une logique simple qui contient le concept universel (noté $\top$), la conjonction de concepts ($\sqcap$) et la quantification universelle qualifiée ($\forall R.C$). Le langage $\mathcal{ALN}$ est plus expressif car il contient, en plus des constructeurs de $\mathcal{FL}_0$, le concept inconsistant (noté $\bot$), la négation des concepts atomiques (i.e., des descriptions simples formées uniquement d'un nom de concept) ainsi que les contraintes de cardinalités sur les rôles ($\geq n\ R$ et $\leq n\ R$, où n est un entier positif et R un nom de rôle).

La sémantique des descriptions repose sur la notion d'interprétation. Plus précisément, une interprétation $\mathcal{I} = (\triangle^{\mathcal{I}}, .^{\mathcal{I}})$ consiste en un ensemble non vide $\triangle^{\mathcal{I}}$, le domaine de l'interprétation, et une fonction $.^{\mathcal{I}}$ qui associe chaque concept A à un sous-ensemble de $\triangle^{\mathcal{I}}$ et chaque rôle R à une relation binaire $R^{\mathcal{I}}$, sous-ensemble de $\triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$. La fonction d'interprétation s'étend aux concepts complexes décrits dans une logique de description particulière. Par exemple, toute interprétation d'un concept $\mathcal{FL}_0$ (resp. $\mathcal{ALN}$) doit respecter la sémantique des constructeurs de ce langage telle que donnée dans la table 2 ci-dessous.

Sémantique			$\mathcal{FL}_0$	$\mathcal{ALN}$
$\top^{\mathcal{I}}$	=	$\Delta^{\mathcal{I}}$	X	X
$\bot^{\mathcal{I}}$	=	$\emptyset$		X
$(\neg A)^{\mathcal{I}}$	=	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$		X
$(C \sqcap D)^{\mathcal{I}}$	=	$C^{\mathcal{I}} \sqcap D^{\mathcal{I}}$	X	X
$(\forall R.C)^{\mathcal{I}}$	=	$\{x \in \Delta^{\mathcal{I}} \mid \forall y : (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$	X	X
$(\leq nR))^{\mathcal{I}}$	=	$\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}}\} \leq n\}$		X
$(\geq nR))^{\mathcal{I}}$	=	$\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}}\} \geq n\}$		X

Figure 2. *Syntaxe et sémantique des constructeurs de concepts pour les logiques de description $\mathcal{FL}_0$ et $\mathcal{ALN}$, où n dénote un entier positif, A un concept atomique, C et D des concepts, R un rôle et le symbole $\sharp$ dénote la cardinalité d'un ensemble.*

Etant donnée la sémantique introduite précédemment, les notions de satisfiabilité, de subsumption et d'équivalence entre deux concepts sont définies comme suit :

– Un concept C est satisfiable ssi il existe une interprétation I telle que $C^I \neq \emptyset$. On dit alors que I est une interprétation valide ou un modèle pour C . Un concept C est dit inconsistant, noté $C \equiv \bot$ ssi il n'admet pas de modèle.

– Un concept C est subsumé par un concept D, noté $C \sqsubseteq D$, ssi $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}\ \forall I$.

– Un concept C est équivalent à un concept D, noté $C \equiv D$, ssi $C \sqsubseteq D$ et $D \sqsubseteq C$.

Au niveau intentionnel, les logiques de description permettent de décrire un domaine d'application sous forme d'une terminologie ou Tbox (i.e., un schéma ou une ontologie). Soit C un nom de concept et D un concept. $C \equiv D$ (resp., $C \sqsubseteq D$) est un axiome terminologique appelé *définition* (resp., *spécification primitive*). Une terminologie $\mathcal{T}$ est un ensemble fini d'axiomes terminologiques tels qu'aucun nom de concept n'apparaît plus d'une fois dans la partie gauche d'un axiome terminologique. Un concept C qui apparaît en partie gauche d'une définition (resp, d'une spécification primitive) est appelé *concept défini* (resp, *concepts primitifs*). Dans cet article, on considère que les terminologies sont acycliques, ce qui veut dire qu'aucun nom de concept n'apparaît directement ou indirectement dans sa propre définition (resp., dans sa spécification primitive).

La sémantique d'une terminologie est obtenue en étendant la notion d'interprétation comme suit. Une interprétation $\mathcal{I}$ satisfait un axiome terminologique $C \equiv D$ (resp., $C \sqsubseteq D$) ssi $C^{\mathcal{I}} = D^{\mathcal{I}}$ (resp., $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$). Une interprétation $\mathcal{I}$ satisfait (ou est un modèle pour) une terminologie $\mathcal{T}$ ssi $\mathcal{I}$ satisfait tout axiome terminologique dans $\mathcal{T}$. Soient C et D deux concepts d'une terminologie $\mathcal{T}$, les notions de subsumption et d'équivalence peuvent être étendues aux terminologies comme indiqué ci-dessous.

– C est subsumé par le concept D selon une terminologie $\mathcal{T}$ noté ($C \sqsubseteq_{\mathcal{T}} D$) ssi $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de $\mathcal{T}$.

– C et D sont équivalents selon une terminologie $\mathcal{T}$ noté ($C \equiv_{\mathcal{T}} D$) ssi $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de $\mathcal{T}$.

Dans les logiques de description, il est possible de ramener le test de subsumption par rapport à une terminologie à un test de subsumption selon une terminologie vide. En effet, étant donné deux concepts C et D d'une ontologie $\mathcal{T}$, il est possible de montrer que $C \sqsubseteq_{\mathcal{T}} D$ ssi $\breve{C} \sqsubseteq \breve{D}$ où $\breve{C}$ et $\breve{D}$ sont respectivement obtenus à partir de C et D par un processus d'aplatissement de la terminologie qui consiste à remplacer de manière récursive dans $\mathcal{T}$ chaque nom de concept par sa définition.

2.1. *Réécriture de concept en utilisant une terminologie*

Nous nous intéressons ici au problème de réécriture de requêtes en utilisant les vues tel qu'il a été étudié notamment dans la cadre des systèmes d'intégration d'information [LEV 01]. L'objectif dans ce contexte est de rechercher les *'réécritures'* maximalement contenues dans une requête donnée. Dans le cadre des logiques de description, ce problème peut-être défini comme suit. Soient deux terminologies $\mathcal{T}$ et $\mathcal{V}$. La terminologie $\mathcal{V}$ est constituée uniquement de définitions qui sont exprimées en utilisant les concepts de $\mathcal{T}$. Soit une requête (i.e., concept) Q exprimée à l'aide des concepts de $\mathcal{T}$. Une réécriture de Q en utilisant la terminologie $\mathcal{T} \cup \mathcal{V}$ est un concept Q' qui vérifie les deux conditions suivantes :

(i) Q' est décrit en utilisant uniquement les noms de concepts définis qui apparaissent dans $\mathcal{V}$, et

(ii) $Q' \sqsubseteq_{\mathcal{T} \cup \mathcal{V}} Q$

En plus, Q' est dite une *réécriture maximalement contenue* dans Q s'il n'existe pas une réécriture Q'' de Q tel que $Q' \sqsubseteq_{\emptyset} Q''$ et $Q'' \not\sqsubseteq Q'$.
Notons qu'ici le test de subsomption entre Q' et Q'' est effectué par rapport à une terminologie vide, ce qui permet de considérer, lors du test, les concepts définis de $\mathcal{V}$ comme étant des concepts de base. Ceci permet de capturer l'hypothèse du monde ouvert (incomplétude des vues) qui est généralement admise dans les systèmes d'intégration [LEV 01].
Le problème de calculer les réécritures maximalement contenues dans une requête a été étudié dans la la littérature pour différents types de langages. Dans le cadre des logiques de description, ce problème est décidable même pour des langages très expressifs tel que le langage $\mathcal{ALCNR}$ [BEE 97].

3. Génération de mappings entre schémas de sources de données

Dans cette section, nous présentons le problème de la génération de mappings décentralisés à partir de mappings centralisés et le formalisons dans le cadre des logiques des description. Plus précisément, nous cherchons à inférer des relations logiques, comme la subsomption ou l'équivalence, entre deux requêtes exprimées en termes des schémas de sources de données. Pour cela, nous déclinons le problème de la découverte de ces mappings en deux problèmes de réécriture : le premier correspond au problème de recherche de réécriture maximalement contenues dans une requête, le second constitue en revanche une nouvelle instance du problème de réécriture de requêtes.

3.1. *Définition du problème*

Nous considérons comme entrées de notre problème : (i) une ontologie globale, représentée sous forme d'une terminologie en logique de description, (ii) un ensemble de sources de données et un ensemble de mappings entre les schémas des sources et l'ontologie globale. Par exemple, dans la Figure 3, on dispose d'une ontologie $\mathcal{O}$, de trois sources de données et de trois mappings M_1, M_2, M_3 entre les concepts des schémas S_1, S_2 et S_3 de ces sources et l'ontologie $\mathcal{O}$. Le mapping M_1 exprime le fait que le concept $S_1.Enseignant$ du schéma S_1 représente dans les termes de l'ontologie $\mathcal{O}$ le concept $EnseignantUniversite$, i.e., des enseignants qui travaillent à l'université. Le mapping M_2 dénote que le concept $S_2.Professeur$ est équivalent à celui du concept $Enseignant$ de $\mathcal{O}$ tandis que le mapping M_3 traduit le fait que le concept $S_3.Employe$ désigne dans $\mathcal{O}$ le concept des travailleurs à l'université. Notre objectif est d'exploiter ces mappings centralisés pour inférer des correspondances sémantiques entre les schémas locaux des sources de données eux-mêmes. Par exemple,

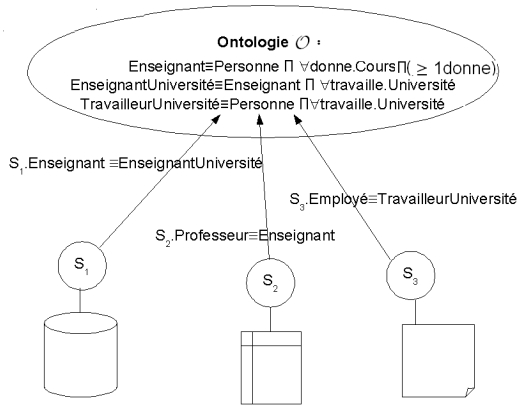


Figure 3. *Exemple de mappings centralisés*

à partir des mappings centralisés de la figure 3, on obtient les mappings de la figure 4, i.e., un mapping entre les schémas S_1 et S_2 puis un mapping entre S_1 et S_3 et enfin un mapping qui implique les trois schémas locaux. Ces mappings sont décentralisés dans le sens où ils spécifient des correspondances entre des schémas de sources uniquement. Le mapping $S_2.Professeur \sqcap S_3.Employe \equiv S_1.Enseignant$ explicite que le concept $S_1.Enseignant$ du schéma S_1 est équivalent à la conjonction des concepts $S_2.Professeur$ et $S_3.Employe$ issus des schémas S_2 et S_3. De même, le mapping $S_1.Enseignant \sqsubseteq S_2.Professeur$ de la figure 4 met en évidence le fait que le concept $S_1.Enseignant$ est subsumé par le concept $S_2.Professeur$.

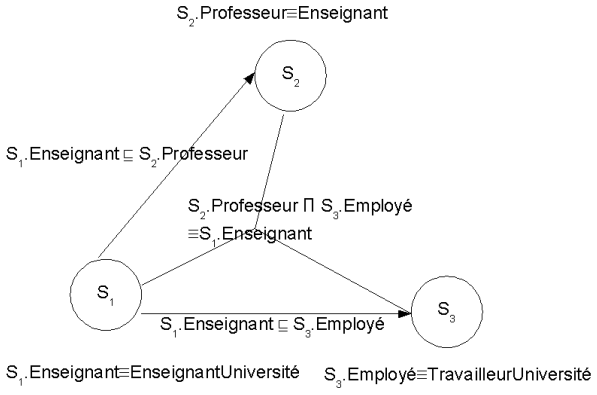


Figure 4. *Exemple de mappings générés*

3.2. *Formalisation dans le cadre des LD*

Dans la suite de l'article, on suppose disposer (c.f., Figure 3) :

– d'une ontologie $\mathcal{O}$ représentée par une terminologie dans une logique de description $\mathcal{L}$,

– d'un ensemble de mappings centralisés $\mathcal{M}$. Ces mappings sont décrits par une terminologie contenant uniquement des définitions exprimées à l'aide des concepts apparaissant dans $\mathcal{O}$. Chaque définition de concepts de la terminologie $\mathcal{M}$ désigne un mapping centralisé. Autrement dit, la terminologie $\mathcal{M}$ définit les concepts issus des différents schémas locaux en termes de l'ontologie $\mathcal{O}$. Dans la Figure 3, les concepts des schémas locaux sont les noms de concepts $S_1.Enseignant$, $S_2.Professeur$ et $S_3.Employe$ et la terminologie $\mathcal{M}$ est formée des trois mappings M_1, M_2 et M_3.

On désigne également par Q, Q' et Q_i où i est un entier naturel des expressions de requêtes qui réfèrent uniquement les noms de concepts définis dans $\mathcal{M}$.

Un mapping décentralisé est une correspondance sémantique entre deux expressions de requête Q_1 et Q_2. On distingue deux types de mappings : l'équivalence et la subsomption. Plus formellement, on définit un mapping décentralisé entre des schémas de sources de données comme suit :

Définition 1 (Mapping décentralisé) *Soient $\mathcal{O}$ une ontologie et $\mathcal{M}$ un ensemble de mappings centralisés. Soient Q_1 et Q_2 des expressions de requêtes en termes de $\mathcal{M}$. Toute assertion M de la forme suivante :*

– $M : Q_1 \equiv_{\mathcal{M}} Q_2$

– $M : Q_1 \sqsubseteq_{\mathcal{M}} Q_2$

est un mapping décentralisé de schémas.

Le premier mapping $Q_1 \equiv_{\mathcal{M}} Q_2$ est un mapping d'équivalence et stipule que les deux expressions de requêtes Q_1 et Q_2 sont sémantiquement équivalentes, i.e., Q_1 et Q_2 représentent le même concept. Le deuxième mapping $Q_1 \sqsubseteq_{\mathcal{M}} Q_2$ est un mapping de subsomption et dénote que l'expression de requête Q_2 décrit un concept qui est plus général que celui décrit par l'expression Q_1.

Tous les mappings n'ont pas le même intérêt d'un point de vue sémantique. Par exemple, dans la figure 3, d'après la définition 1, l'assertion $M_{d_1} : S_1.Enseignant \sqcap S_3.Employe \sqsubseteq S_2.Professeur$ constitue également un mapping entre les schémas des sources S_1, S_2 et S_3. Cependant, le mapping M_{d_1} est redondant de part la présence du mapping $M_{d_2} : S_1.Enseignant \sqsubseteq S_2.Professeur$ dans le sens où il n'apporte aucune information supplémentaire par rapport au mapping M_{d_2}. De plus, dans un contexte d'intégration où il s'agit de résoudre une requête à partir des mappings existants entre des schémas des sources de données, on souhaite pour des raisons d'efficacité, exploiter uniquement les mappings conduisant à un maximum de réponses. En effet, si une requête recherche des professeurs, i.e., $Q \equiv S_2.Professeur$, la propagation de Q dans un système pair à pair implique l'exploitation du mapping M_{d_1}

ou celle du mapping M_{d_2}. Le mapping M_{d_2} conduit à la recherche des tuples satisfaisant le concept local $S_1.Enseignant$ et suffit localement à répondre à Q. Dans le cas du mapping M_{d_1}, il est nécessaire de rechercher les tuples satisfaisant le concept local $S_1.Enseignant$ puis les tuples satisfaisant le concept $S_3.Employe$ et d'en faire l'intersection. L'ensemble des tuples obtenus par ce biais ne peut être qu'un sous-ensemble des tuples satisfaisant $S_1.Enseignant$. Ainsi dans la suite de l'article, nous focalisons sur les mappings non redondants.

Définition 2 (Mapping non redondant) *Soient $\mathcal{O}$ une ontologie et $\mathcal{M}$ une terminologie. Soient Q_1, Q_2, Q'_1 et Q'_2 des expressions de requêtes en termes de $\mathcal{M}$.*

Un mapping $M : Q_1 \sqsubseteq_{\mathcal{M}} Q_2$, est non redondant si et seulement si :

i) $\nexists Q'_1 \mid Q_1 \sqsubseteq_{\emptyset} Q'_1$ *et* $Q'_1 \sqsubseteq_{\mathcal{M}} Q_2$ *et*

ii) $\nexists Q'_2 \mid Q'_2 \sqsubseteq_{\emptyset} Q_2$ *et* $Q_1 \sqsubseteq_{\mathcal{M}} Q'_2$.

3.3. *De la découverte de mappings à la réécriture de requêtes*

Soit une ontologie $\mathcal{O}$ et un ensemble de mappings $\mathcal{M}$. Nous nous intéressons à présent au calcul des mappings décentralisés non redondants. De plus, nous nous concentrons uniquement sur les mappings de subsomption car par définition, leur découverte permet de trouver les mappings d'équivalence lorsqu'ils existent. Pour générer ce type de mappings, nous exprimons le problème de la découverte des mappings $Q_1 \sqsubseteq Q_2$ comme un problème de réécriture d'une requête en termes d'une terminologie $\mathcal{T}$. Plus précisément, selon le sens de lecture des assertions, ce problème peut se décliner en deux problèmes de réécriture de requêtes en termes de vues. Dans la première approche, nous supposons disposer d'une expression de requête Q en termes d'un sous-ensemble E de $\mathcal{M}$. Puis nous considérons Q comme étant la spécification de la partie droite d'un mapping possible. Le problème est alors de définir la partie gauche du mapping, i.e., de découvrir s'il existe une correspondance sémantique entre les concepts de Q et des concepts issus d'autres schémas locaux. Une approche possible consiste à réécrire l'expression Q en une expression Q' en termes de $\mathcal{T} \backslash E$ qui soit subsumée par Q. Ce problème noté $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ est défini comme suit :

Problème 1 ($M_{max}(\mathcal{O}, \mathcal{M}, Q)$) *Soient $\mathcal{O}$ une ontologie et $\mathcal{M}$ un ensemble de mappings centralisés. Soit Q une expression de requêtes en termes de $E \subseteq \mathcal{M}$. Le problème $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ consiste à trouver l'ensemble des expressions de requêtes Q' en termes de $\mathcal{M} \backslash E$ telles que $Q' \sqsubseteq_{\mathcal{M}} Q$ où le mapping $Q' \sqsubseteq_{\mathcal{M}} Q$ est non redondant.*

La seconde approche de découverte des mappings consiste à supposer le terme de gauche Q de l'assertion connu puis de calculer Q', le terme de droite du mapping. Plus précisément, Q' est une reformulation de Q en terme de $\mathcal{M} \backslash E$ qui subsume Q. Ce deuxième problème noté $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ est défini comme suit :

Problème 2 (*$M_{min}(\mathcal{O}, \mathcal{M}, Q)$*) *Soient $\mathcal{O}$ une ontologie et $\mathcal{M}$ un ensemble de mappings centralisés. Soit Q une expression de requêtes en termes de $E \subseteq \mathcal{M}$. Le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ consiste à trouver l'ensemble des expressions de requêtes Q' en termes de $\mathcal{M} \backslash E$ telles que $Q \sqsubseteq_{\mathcal{M}} Q'$ où le mapping $Q \sqsubseteq_{\mathcal{M}} Q'$ est non redondant.*

Le premier problème $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ de découverte de mappings entre des expressions de requêtes en termes de $\mathcal{M}$ peut être considéré comme une instance du cadre général de la réécriture de requêtes en utilisant une terminologie. En effet, les solutions du problème $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ sont des mappings de la forme $Q' \sqsubseteq Q$ et sont non redondants. Ainsi, Q' réfère uniquement des concepts définis dans la terminologie $\mathcal{M}$. De plus Q' est subsumée par Q et par définition de la redondance, Q' est maximal. Par conséquent, Q' satisfait la définition d'une réécriture maximale telle que décrite dans la section 2.1. Par la suite, on note toute solution du problème $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ par l'assertion $Q' \sqsubseteq_{max} Q$.

Le deuxième problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ peut être vu comme une nouvelle instance du problème de réécriture de requête dans le sens où des généralisations de la requête Q plutôt que des spécialisations sont recherchées. Par la suite, on note toute solution du problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ par l'assertion $Q \sqsubseteq_{min} Q'$ et on dit aussi que Q' est une réécriture minimale qui subsume Q.

La section suivante s'attache à résoudre le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$. Deux cas sont distingués : (i) le cas des langages fermés par la négation et (ii) le cas des langages ayant la propriété de la subsomption structurelle. Le problème $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ ne sera pas détaillé puisqu'il est équivalent à un problème de réécriture largement étudié dans la littérature [BEE 97, LEV 01, GOA 04].

4. Calcul des mappings $M_{min}(\mathcal{O}, \mathcal{M}, Q)$

Nous commençons par proposer une solution pour les LD fermées par la négation et nous focalisons par la suite, pour les LD n'autorisant pas la négation, sur la résolution du problème dans le cas des LD possédant la propriété de subsomption structurelle.

4.1. *Cas des LD fermées par rapport à la négation*

Pour les LD autorisant la négation, la résolution du problème de découverte de réécritures minimales est équivalente à la résolution de sa contraposée et revient donc au problème $M_{max}(\mathcal{O}, \mathcal{M}, Q)$ comme indiqué dans le théorème suivant :

Théorème 3 *Le problème de trouver les solutions de $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ est équivalent à résoudre le problème $M_{max}(\mathcal{O}, \mathcal{M}, \neg Q)$*
$Q \sqsubseteq_{min} Q' \Leftrightarrow \neg Q' \sqsubseteq_{max} \neg Q$

La démonstration du théorème repose sur l'axiome suivant :
Dire que $Q \sqsubseteq_{min} Q'$ est équivalent à dire que $\neg Q' \sqsubseteq_{max} \neg Q$.

Ainsi, d'après le théorème précédent, il est possible de résoudre le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ en réutilisant des algorithmes existants de calcul des réécritures maximalement contenues.

4.2. *Cas des LD à subsomption structurelle*

La transformation précédente du problème étant impossible dans les LD qui ne sont pas fermées par la négation, nous proposons d'étudier le problème pour une autre famille de LD qui sont les LD à subsomption structurelle. Nous proposons dans ce cadre de réduire le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ de recherche de réécritures minimales à un problème de calcul des transversaux minimaux d'un hypergraphe. Dans la suite, nous rappelons les caractéristiques des LD à subsomption structurelle, puis nous formalisons le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ à l'aide des hypergraphes et nous donnons une approche pour le résoudre.

4.2.1. *Caractéristiques des LD à subsomption structurelle*

Nous rappelons ci-dessous la forme clausale d'une description en LD.
Soit $\mathcal{L}$ une logique de description. Une clause dans $\mathcal{L}$ est une description A qui respecte la propriété suivante :
$(A \equiv B \sqcap A') \Rightarrow (B \equiv \top) \wedge (A' \equiv A) \vee (A' \equiv \top) \wedge (B \equiv A)$. Chaque conjonction $A_1 \sqcap ... \sqcap A_n$ de clauses peut être représentée par l'ensemble de clauses $\hat{A} = \{A_1, ..., A_n\}$.
Un ensemble de clauses $\hat{A} = \{A_1, ..., A_n\}$ est dit réduit si :

– soit $n = 1$.

– soit aucune clause ne subsume la conjonction des autres clauses, i.e, :
$\forall i \mid i \in \{1, ..., n\}, A_i \not\sqsupseteq (\hat{A} \backslash A_i)$.

On dit que l'ensemble $\hat{A}$ [4] est une *forme clausale réduite* (RCF).
Nous définissons maintenant la notion de logique à subsomption structurelle telle qu'elle a été donnée par [TEE 94].

Définition 4 *(Propriété de subsomption structurelle) On dit qu'une logique de description $\mathcal{L}$ possède la propriété de subsomption structurelle si et seulement si pour toute description $A \equiv A_1 \sqcap ... \sqcap A_n$ et toute description $B \equiv B_1 \sqcap ... \sqcap B_m$ dans $\mathcal{L}$, données par leur RCF, on a :*
$B \sqsubseteq A \Leftrightarrow \forall A_j \in \hat{A}, \exists B_i \in \hat{B} \mid B_i \sqsubseteq A_j$
En d'autres termes, si $B \sqsubseteq A$ alors $\hat{A} \subseteq \hat{B}$ ($\{A_j\} \subseteq \{B_i\}$).

---

4. Dans le reste de l'article, nous désignons par $\hat{A}$ la forme clausale réduite (RCF) du concept A.

Pour un langage à subsomption structurelle, chercher les réécritures Q' qui subsument une expression de requête Q ($Q \sqsubseteq Q'$) revient à chercher les Q' telles que $\hat{Q'} \subseteq \hat{Q}$.

Exemple 1 *$Q \equiv A \sqcap B \sqcap C$, $Q_1 \equiv A \sqcap B$, $Q_2 \equiv A$ et $Q_3 \equiv A \sqcap D$*
$\{A, B\} \subseteq \{A, B, C\} \Rightarrow Q \sqsubseteq Q_1$ et $\{A\} \subseteq \{A, B, C\} \Rightarrow Q \sqsubseteq Q_2$
Q_1 et Q_2 sont des réécritures qui subsument Q.

Lemme 1 *Chercher les réécritures minimales qui subsument une expression de requête Q revient à chercher la réécriture Q' qui subsume Q telle que $\hat{Q'}$ soit maximale par rapport à l'inclusion (i.e., $\nexists Q'' \mid Q \sqsubseteq Q''$ et $\hat{Q'} \subseteq \hat{Q''}$)*

Démonstration
La démonstration de ce lemme se base sur la définition de la propriété de subsomption structurelle. En effet, $Q \sqsubseteq Q' \Leftrightarrow \forall B_j \in \hat{Q'}, \exists A_i \in \hat{Q} : A_i \sqsubseteq B_j$. Autrement dit, $\Leftrightarrow \hat{Q'} \subseteq \hat{Q}$. Nous déduisons donc qu'une description minimale contient le plus grand nombre de clauses.

4.2.2. *Modélisation du problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ avec les hypergraphes*

Nous montrons maintenant que le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ de découverte de réécritures minimales revient au problème de recherche des transversaux minimaux d'un hypergraphe. Nous commençons par rappeler les définitions des hypergraphes.

Définition 5 *(Hypergraphe et transversaux [EIT 95])*
Un hypergraphe $\mathcal{H}$ est un couple (Σ, Γ) où Σ est un ensemble d'éléments $\Sigma = \{S_1, ..., S_n\}$ et $\Gamma = \{\epsilon_1, ..., \epsilon_n\}$ un ensemble de sous-ensembles de Σ. Les éléments de Σ sont appelés sommets et les éléments de Γ sont appelés arêtes.
Un ensemble $\mathcal{R} \subseteq \Sigma$ est un transversal de $\mathcal{H}$ si pour chaque $\epsilon \in \Gamma$, $\mathcal{R} \cap \epsilon \neq \emptyset$. Un transversal $\mathcal{R}$ est minimal si aucun sous-ensemble strict $\mathcal{R}'$ de $\mathcal{R}$ n'est un transversal. L'ensemble des transversaux minimaux d'un hypergraphe $\mathcal{H}$ est noté Tr($\mathcal{H}$).

Soit une instance du problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$, Q et $\mathcal{M}$ sont exprimées dans une logique de description $\mathcal{L}$ ayant la propriété de subsomption structurelle, avec $\mathcal{M}$ l'ensemble des mappings (cf. section 3.2) tel que $\mathcal{M} = \{M_i, i \in \{1, .., n\}\}$ et $Q \equiv \sqcap_{j=1}^{k} A_j$. Nous supposons que Q et les M_i sont donnés sous leurs formes clausales réduites (RCF).
La résolution du problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ consiste à rechercher les réécritures Q' qui subsument Q (i.e, $\hat{Q'} \subseteq \hat{Q}$, d'après la définition 4) et qui soient minimales (i.e, qui couvrent le maximum de clauses dans Q, d'après le lemme 1). Pour cela, nous construisons l'hypergraphe $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$ =(Σ, Γ) de la manière suivante :

– Chaque $M_i \in \mathcal{M} \mid \hat{M_i} \subseteq \hat{Q}$ devient un sommet S_{M_i} de l'hypergraphe $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$

– Chaque clause $A_j \in \hat{Q} \mid A_j \in \{\hat{M}_i \cap \hat{Q}\}$ et $\hat{M}_i \subseteq \hat{Q}$, devient une arête dans $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$, notée ϵ_{A_j} avec $\epsilon_{A_j} = \{S_{M_i}\}$.

Exemple 2 *Considérons une instance du problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ où $\mathcal{O}$, Q et $\mathcal{M}$ sont exprimées dans la logique $\mathcal{FL}_0$. Les concepts $TravailleurUniversité$, $TravailleurSecondaire$ et $TravailleurExpérimenté$ sont des concepts primitifs décrivant respectivement les travailleurs à l'université, dans un établissement secondaire et les travailleurs expérimentés.*

$M_1 : S_1.Professeur \equiv Personne \sqcap \forall donne.Cours \sqcap TravailleurUniversité \sqcap TravailleurExpérimenté$, $M_2 : S_2.Enseignant \equiv Personne \sqcap \forall donne.Cours$,
$M_3 : S_3.Employé \equiv TravailleurSecondaire \sqcap \forall donne.Cours$,
$M_4 : S_4.Personnel \equiv Personne \sqcap TravailleurUniversité$ *et*
$M_5 : S_5.Travailleur \equiv TravailleurUniversité$.
Q est une requête qui recherche les professeurs, $Q \equiv S_1.Professeur$

Nous construisons l'hypergraphe $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$ de l'exemple 2 de la manière suivante ; Chaque clause dans les $M_i \mid \hat{M}_i \subseteq \hat{Q}$ devient une arête dans $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$ et les M_i correspondantes deviennent des sommets comme suit, $\Sigma = \{S_{M_2}, S_{M_4}, S_{M_5}\}$ et les arêtes, $\epsilon_1 = \epsilon_{Personne}$, $\epsilon_2 = \epsilon_{\forall donne.Cours}$, $\epsilon_3 = \epsilon_{TravailleurUniversité}$.

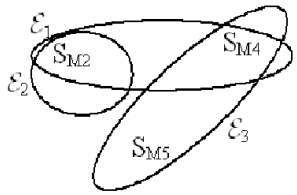


Figure 5. *L'hypergraphe $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$ construit à partir de Q et de $\mathcal{M}$ dans l'exemple 2*

Théorème 6 *Résoudre le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ est équivalent à trouver les transversaux minimaux de l'hypergraphe $\mathcal{H}_{\mathcal{M}_\mathcal{Q}}$.*

Ce théorème donne une méthode pratique pour résoudre le problème $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ en le ramenant à un problème de calcul des transversaux minimaux d'un hypergraphe. Ainsi les algorithmes de calcul de transversaux minimaux d'un hypergraphe comme ceux de [BER 89] peuvent être utilisés dans notre contexte.

Les transversaux minimaux de l'hypergraphe donné par la figure 5 sont $\{S_{M_2}, S_{M_4}\} et \{S_{M_2}, S_{M_5}\}$. Par conséquent, les réécritures minimales subsumant Q sont $Q' \equiv S_2.Enseignant \sqcap S_4.Personnel$ et $Q' \equiv S_2.Enseignant \sqcap S_5.Travailleur$.

5. Discussion

La génération automatique (ou semi-automatique) de correspondances sémantiques n'est pas un problème nouveau. Parmi les travaux existants [RAH 01, MAD 05, SHV 05] se dégagent deux classes de problème. Une grande majorité vise à établir des *similarités* linguistiques ou structurelles entre des termes extraits de différents schémas. Les similarités obtenues ne sont pas directement exploitables à des fins de résolution de requêtes dans les systèmes d'intégration. La recherche des ces similarités constitue cependant une phase préliminaire indispensable à la découverte de la seconde catégorie de correspondances dites sémantiques [HER 01]. Récemment, un consensus sur la définition de telles correspondances, utiles dans le cadre des systèmes d'intégration généralisés pair-à-pair, a été établi [MAD 02, SER 04, HAL 04]. Une correspondance sémantique est une *assertion logique* de la forme e_1 *op* e_2 où e_1 et e_2 sont deux expressions de requêtes en termes des schémas de sources de données et où *op* désigne une relation logique [MAD 02]. Parmi les travaux portant sur la génération de correspondances sémantiques, les références [MAD 03, KOL 05] s'appuient sur le principe de réutilisation des mappings existants. Cependant, ces travaux focalisent sur un problème nouveau et très particulier qui est celui de la composition de mappings afin d'optimiser le traitement des requêtes dans les environnements distribués. Il s'agit à partir de mappings existants depuis une source A vers une source B puis d'une source B vers une source C de déterminer s'il est possible de déduire un mapping direct de A vers C équivalent à la réalisation successive des deux premiers.

Dans cet article, nous nous sommes intéressés au problème de génération automatique de correspondances sémantiques décentralisées entre des schémas de sources de données autonomes à partir de mappings centralisés existants issus de systèmes de médiation classiques ou d'applications du web sémantique. Le problème de la découverte de mappings décentralisés a été formalisé dans le cadre des logiques de description et peut s'exprimer comme un problème de réécriture de requêtes. Plus précisément, deux sous-problèmes de réécriture ont été mis en évidence. Le premier est équivalent à un problème de réécriture classique de recherche des réécritures maximalement contenues dans une requête tandis que le second problème consiste à rechercher les réécritures *minimales contenant* une requête. Le deuxième problème dans le cadre des logiques fermées par rapport à la négation se réduit au problème du calcul des réécritures maximalement contenues dans une requête. Les algorithmes existants peuvent alors être utilisés pour résoudre ce problème. En revanche, dans le cadre des logiques caractérisées par la propriété de subsomption structurelle, un algorithme basé sur la théorie des hypergraphes a été proposé pour calculer les réécritures minimales contenant la requête. Un prototype est en cours de réalisation. En terme de perspectives, nous nous intéressons à un autre type de mapping, des mappings de généralisation. Supposons dans un systèmes pair à pair, que deux pairs P_1 et P_2 stockent des étudiants de deux villes différentes, i.e., les $EtudiantsClermontois$ et les $EtudiantsLyonnais$. Ces deux concepts ne sont pas directement comparables mais une relation de généralisation les lie dans le sens où ce sont des étudiants. Cette correspondance peut être exploitée pour propager de P_1 vers P_2 une requête recherchant des étudiants.

6. Bibliographie

[ABI 03] ABITEBOUL S., AL., « The Lowell Database Research Self-Assessment », rapport, 2003.

[BAA 03] BAADER F., DONINI F., MCGUINNESS D., NARDI D., PATEL-SCHNEIDER P., « The description logic handbook », *theory, implementation, and applications*, Cambridge University Press, 2003.

[BEE 97] BEERI C., LEVY A., ROUSSET M., « Rewriting Queries Using Views in Description Logics », *PODS'97*, ACM Press, 1997, p. 99 - 108.

[BER 89] BERGE C., « Hypergraphs », vol. 45 of North Holland Mathematical Library, Elsevier Science Publishers B.V, 1989.

[EIT 95] EITER T., GOTTLOB G., « Identifying the minimal transversals of a hypergraph and related problems », vol. 24(6), 1995, p. 1278 - 1304.

[GOA 04] GOASDOUÉ F., ROUSSET M., « Answering queries using views : A KRDB perspective for the semantic Web », *ACM Trans. Inter. Tech.*, vol. 4, n° 3, 2004, p. 255–288, ACM Press.

[HAL 04] HALEVY A. Y., IVES Z. G., MADHAVAN J., MORK P., SUCIU D., TATARINOV I., « The Piazza Peer Data Management System. », *IEEE Trans. Knowl. Data Eng.*, vol. 16, n° 7, 2004, p. 787-798.

[HER 01] HERNÁNDEZ M. A., MILLER R. J., HAAS L. M., « Clio : A Semi-Automatic Tool For Schema Mapping. », *SIGMOD Conference*, 2001, page 607.

[KOL 05] KOLAITIS P. G., « Schema mappings, data exchange, and metadata management », *PODS '05*, ACM Press, 2005, p. 61–75.

[LEN 02] LENZERINI M., « Data integration : a theoretical perspective », *PODS'02*, ACM Press, 2002, p. 233 - 246.

[LEV 01] LEVY A., « Answering Queries Using Views : A Survey », *The VLDB Journal*, vol. 10, Springer-Verlag, 2001, p. 270 - 294.

[MAD 02] MADHAVAN J., BERNSTEIN P. A., DOMINGOS P., HALEVY A. Y., « Representing and reasoning about mappings between domain models », *Eighteenth national conference on Artificial intelligence*, American Association for Artificial Intelligence, 2002, p. 80–86.

[MAD 03] MADHAVAN J., HALEVY A., « Composing Mappings among Data Sources », , 2003.

[MAD 05] MADHAVAN J., BERNSTEIN P., DOAN A., HALEVY A., « Corpus-based Schema Matching », *(ICDE'05*, 2005, p. 57-68.

[RAH 01] RAHM E., BERNSTEIN P., « A survey of approaches to automatic schema matching », , 2001, p. 49-58.

[SER 04] SERAFINI L., TAMILIN A., « Local Tableaux for Reasoning in Distributed Description Logics », *DL'04*, 2004.

[SHV 05] SHVAIKO P., EUZENAT J., « A survey of schema-based matching approaches », *Journal on Data Semantics*, 2005.

[TEE 94] TEEGE G., « Making the Difference : A Subtraction Operation for Description Logics », DOYLE J., SANDEWALL E., TORASSO P., Eds., *Principles of Knowledge Representation and Reasoning : Proc. of the 4th International Conference (KR94)*, San Francisco, California, 1994, Morgan Kaufmann, p. 540-550.