
Accès personnalisé à des sources de données multiples : évaluation de deux approches de reformulation de requêtes

Dimitre Kostadinov, Mokrane Bouzeghoub, Stéphane Lopes

Laboratoire PRiSM
CNRS-Université de Versailles
45, avenue des Etats-Unis 78035 Versailles
prénom.nom@prism.uvsq.fr

RÉSUMÉ. Les systèmes de médiation actuels permettent un accès transparent à un ensemble de sources de données hétérogènes. Le but de la personnalisation est de faciliter l'expression du besoin de l'utilisateur et lui permettre d'obtenir des informations pertinentes en exploitant au mieux ses préférences regroupées dans un profil. Dans un contexte de médiation, la personnalisation doit tenir compte non seulement du profil des utilisateurs mais aussi de la description sémantique des sources de données, définie par des requêtes de médiation. Cet article décrit et évalue deux approches de reformulation de requêtes sur la base de deux métriques qui sont la couverture et la précision des prédicats utilisés dans l'enrichissement de la requête utilisateur.

ABSTRACT. Mediation systems allow transparent access to a set of heterogeneous data sources. Personalization is a mechanism that helps users to express their needs and enables them to obtain relevant results by exploiting a set of preferences in their profiles. In a mediation context, the personalization process must take into account not only the user profile but also the semantic description of the data sources given in mediation queries. This paper describes and evaluates two query reformulation techniques on the basis of two metrics which are coverage and precision of the predicates used in the query enrichment.

MOTS-CLÉS : personnalisation, accès adaptatif, reformulation de requêtes.

KEYWORDS: personalization, adaptative access, query reformulation.

CATEGORIE : Chercheur

Cette recherche a été partiellement soutenue par le Ministère Délégué à la Recherche et aux Nouvelles Technologies, dans le programme ACI Masses de Données, projet #MD-33 (APMD).

1. Introduction et contexte

Les systèmes d'information actuels donnent accès à des sources de données multiples, distribuées, autonomes et potentiellement redondantes. Une des principales limites de ces systèmes est leur incapacité à discriminer les utilisateurs en fonction de leurs centres d'intérêt, de leurs préférences et de leurs contextes de requêtage, et à leur délivrer des résultats pertinents selon leurs profils respectifs. Ces limites sont inhérentes aux systèmes de bases de données qui ont été conçus pour une utilisation dans des domaines d'applications fermés, où l'utilisateur connaît non seulement le schéma de la base, mais suppose aussi que tous les objets de son univers sont dans la base de données au moment de leur utilisation (hypothèse du monde fermé). De ce fait, la requête est une expression exacte de son besoin. Les systèmes d'intégration de données n'ont fait que généraliser cette approche à un ensemble de sources de données distribuées, sans remise en cause de l'hypothèse du monde fermée, prolongeant ainsi la survie des modes opératoires classiques. Par exemple, un système de médiation de données est perçu à travers son schéma global sur lequel l'utilisateur exprime ses requêtes qui sont ensuite réécrites ou décomposées pour être exécutées de façon tout à fait classique sur les sources de données participant à la médiation. L'évolution des sources de données, leur indisponibilité temporaire ou permanente, l'ajout de nouvelles sources ne sont pas pris en compte; tout se passe comme s'il y avait toujours un administrateur qui maintenait la vision classique d'une base de données, sans variation de sa sémantique initiale.

Cependant, la multiplicité des sources de données, leur évolutivité et la difficulté croissante de maîtriser leurs descriptions et leurs contenus (notamment dans les architectures P2P) font émerger de nouvelles pratiques qui s'apparentent plus à celles utilisées dans les systèmes de recherche d'information. Les utilisateurs ne connaissent pas forcément les sources de données qu'ils interrogent ni leur description et leur contenu. En conséquence, leurs requêtes ne traduisent plus un besoin précis mais une intention qui doit être affinée en fonction des sources de données disponibles au moment de l'interrogation.

Par ailleurs, ces utilisateurs ont de nouvelles exigences telles que la prise en compte de leur localité géographique, le média utilisé pour l'expression de leurs requêtes, leurs préférences récurrentes en termes de qualité des données, de présentation des résultats, de sécurité, etc. Ainsi, si ces préférences sont prises en compte, l'exécution de la même requête, exprimée par des utilisateurs différents, ne produit pas nécessairement les mêmes résultats. C'est ce qu'on appelle un *accès personnalisé à l'information*.

Pour répondre aux besoins de la personnalisation, différentes approches ont été adoptées: extension des langages de requêtes comme dans PreferenceSQL (Kießling, 2002) ou SQL/f (Rocacher et al., 2006), enrichissement de requêtes à l'aide de préférences définies dans un profil utilisateur (Koutrika et al., 2004), sélection des sources de données en fonction de leur qualité (Naumann et al., 1998). Les trois premières approches s'inscrivent dans le cadre de l'accès à une source de données unique. La troisième approche s'inscrit dans le cadre de systèmes multi source mais ne vise que la sélection des sources selon des critères de qualité. Par ailleurs, c'est une approche statique réalisée pour un ensemble de requêtes et non pour chaque requête.

Mais aucune de ces approches ne prend en compte la personnalisation dans sa globalité, tenant compte à la fois des profils des utilisateurs pour l'enrichissement des requêtes et de la sémantique des sources de données pour la réécriture de ces requêtes dans un contexte distribué.

Dans cet article, nous introduisons le problème de la personnalisation dans un système de médiation relationnel. Etant donné la volatilité des sources de données dans notre contexte, une approche *Local As View* (LAV) pour l'expression des requêtes de médiation nous semble bien adaptée. D'autre part, le principal problème dans ce contexte étant la surcharge d'informations résultant d'une requête, le but que nous nous fixons est de restreindre la réponse à une requête en ciblant de façon plus précise les besoins de l'utilisateur. Ce dernier point a pour conséquence que le résultat de la requête « personnalisée » sera toujours inclus dans le résultat de la requête initiale.

Le cœur de cette problématique est la *reformulation* de requêtes. Elle consiste à réinterpréter l'intention de l'utilisateur, exprimée dans sa requête, en une requête plus complète, tenant compte à la fois du profil utilisateur et de la description des sources de données.

Partant de deux algorithmes connus et réalisant respectivement la réécriture (Levy et al., 1996) et l'enrichissement (Koutrika et al., 2004) de requêtes, nous proposons deux approches de reformulation de requêtes basées sur les deux combinaisons possibles de ces algorithmes. Afin de comparer objectivement ces deux approches, nous définissons deux métriques (la *couverture* et la *précision*) adaptées des mesures communément utilisées en Recherche d'Information. Finalement, nous expérimentons les deux approches proposées sur un jeu de test afin de réaliser une étude comparative pour mettre en évidence les avantages et inconvénients de chacune d'elles.

La section 2 présente un exemple illustratif qui va servir de base à l'étude de ces deux approches. La section 3 rappelle les principes des algorithmes de réécriture et d'enrichissement de requêtes que nous utilisons. La section 4 décrit nos deux approches de médiation personnalisée, à l'aide de l'exemple. La section 5 définit deux métriques qui sont la couverture et la précision afin d'analyser et de comparer ces approches en se basant sur les résultats que nous avons obtenus sur un échantillon de requêtes. La section 6 conclut l'article en discutant de la généralisation de l'approche et des évolutions futures.

2. Exemple illustratif

Cette section décrit un exemple de système de médiation avec son schéma virtuel et les liens sémantiques (ou requêtes de médiation) qui le relient aux sources de données participantes. Dans la suite on utilisera S_v pour désigner le schéma virtuel et S_m pour désigner l'ensemble de requêtes de médiation $\{S_1, \dots, S_n\}$.

Exemple 1 : Schéma virtuel (S_v)

VOYAGE(idV, prix, lieuDep, lieuArr, nbJours, dateDep, heure,
typeSejour, formule, idT, idH)

TRANSPORT(idT, moyen, typeTrajet, confort)

HOTEL(idH, nbEtoiles, nom, region, ville, restaurant)

Notre exemple traite des voyages, des moyens de transport et des hôtels qu'un voyageur peut réserver pour des séjours professionnels ou d'agrément. Son schéma virtuel est composé des relations de l'exemple 1. Les instances de ce schéma sont calculées à partir de six sources de données (Exemple 1 (suite)), définies par des expressions LAV sur le schéma de médiation. Ces expressions, appelées aussi requêtes de médiation, vont servir à la réécriture des requêtes utilisateur, avant ou après leur enrichissement par les profils utilisateur.

Exemple 1 (suite) : Définitions des sources de données

```

S1:HOTELSDUMONDE(idH, nbEtoiles, nom, region, ville, restaurant) :-
    HOTEL(idH, nbEtoiles, nom, region, ville, restaurant).

S2:TRANSPORTAERIEN(idT, lieuDep, lieuArr, dateDep, heure, moyen,
    typeTrajet, confort) :-
    TRANSPORT(idT, 'avion', typeTrajet, confort),
    VOYAGE(idV, prix, lieuDep, lieuArr, nbJours, dateDep, heure,
    typeSejour, formule, idT, idH).

S3:SNCF(idT, lieuDep, lieuArr, dateDep, heure, moyen, typeTrajet, confort) :-
    TRANSPORT(idT, 'train', typeTrajet, confort),
    VOYAGE(idV, prix, lieuDep, lieuArr, nbJours, dateDep, heure,
    typeSejour, formule, idT, idH).

S4:VOYAGERPARTOUT(idT, lieuDep, lieuArr, dateDep, heure, moyen,
    typeTrajet, confort) :-
    TRANSPORT(idT, moyen, typeTrajet, confort),
    VOYAGE(idV, prix, lieuDep, lieuArr, nbJours, dateDep, heure,
    typeSejour, formule, idT, idH).

S5:PROMOVACANCES(idV, prix, lieuDep, lieuArr, nbJours, dateDep, heure,
    typeSejour, Formule, moyen, nom, nbEtoiles, restaurant, idT, idH) :-
    TRANSPORT(idT, moyen, typeTrajet, confort),
    VOYAGE(idV, prix, 'Paris', lieuArr, nbJours, dateDep, heure,
    typeSejour, formule, idT, idH),
    HOTEL(idH, nbEtoiles, nom, region, lieuArr, restaurant), prix<950.

S6:LYONVACANCES(idV, prix, lieuDep, lieuArr, nbJours, dateDep, heure,
    typeSejour, formule, moyen, nom, nbEtoiles, restaurant, idT, idH) :-
    TRANSPORT(idT, moyen, typeTrajet, confort),
    VOYAGE(idV, prix, 'Lyon', lieuArr, nbJours, dateDep, heure,
    typeSejour, formule, idT, idH),
    HOTEL(idH, nbEtoiles, nom, region, lieuArr, restaurant), prix<950.

```

3. Réécriture et enrichissement des requêtes

Cette section rappelle les mécanismes de réécriture et d'enrichissement de requêtes que nous avons retenus pour illustrer l'importance de la personnalisation dans un système d'intégration de données. Les deux algorithmes traitent des requêtes conjonctives exprimées sur un schéma virtuel relationnel.

3.1. Réécriture des requêtes

La réécriture de requêtes consiste à transformer la requête de l'utilisateur exprimée sur le schéma virtuel afin qu'elle puisse être évaluée sur les sources de données. Nous nous limiterons

à présenter la réécriture de requêtes dans un contexte LAV qui est le mieux adapté à une évolution rapide des sources de données.

La réécriture d'une requête dans une approche *Local As View* consiste à déterminer les sources contributives pour l'exécution de la requête utilisateur et à utiliser leurs définitions pour reformuler cette requête (Calvanese et al., 2001). Il existe deux classes principales d'algorithmes de réécriture: les algorithmes de règles inversées (Duschka et al., 1997) et les algorithmes à base de tas (*Bucket-based*) (Levy et al., 1996).

Ces deux types d'algorithmes pourraient être considérés pour notre étude. Cependant pour simplifier la présentation, nous nous limiterons à un algorithme du second type. Il comporte deux phases:

- Création d'un tas (*bucket*) pour chaque sous-but g de la requête Q . Un sous-but correspond à une relation virtuelle qui apparaît dans l'expression de Q . Ce tas contient la définition des sources contributives pour ce sous-but.
- Construction des réécritures candidates en ne gardant que les réécritures qui sont incluses dans la requête. Une requête Q_i est incluse dans une autre Q_j si pour toute base de données D , l'ensemble des tuples retournés par l'évaluation de Q_i sur D est un sous-ensemble des tuples retournés par Q_j . Chaque réécriture candidate est une requête conjonctive obtenue en prenant un élément de chaque tas.

Prenons une requête utilisateur Q_1 , exprimée sur le schéma virtuel de l'exemple 1, qui cherche des voyages à destination de Madrid pour une durée de 4 jours (Exemple 2).

Exemple 2 : Requête initiale

```
Q1 = SELECT idV, prix, V.lieuDep, T.moyen, T.confort
      FROM VOYAGE V, TRANSPORT T
      WHERE V.idT = T.idT AND V.lieuArr='Madrid' AND V.nbJours=4;
```

La première phase de réécriture de Q_1 correspond à la construction des 2 tas de sources contributives, $\{S_2, S_3, S_4\}$ et $\{S_5, S_6\}$ respectivement pour les relations virtuelles TRANSPORT et VOYAGE. La deuxième phase consiste à construire la réécriture de Q_1 comme l'union de toutes les réécritures candidates possibles obtenues en prenant une source contributive de chaque tas.

L'exemple 3 illustre la réécriture candidate de Q_1 par rapport à S_2 et S_5 qui définissent respectivement les sources TransportAerien et PromoVacances.

Exemple 3 : Exemple de réécritures candidates de Q_1

```
RW(idV, prix, lieuDep, moyen, confort):-  
TRANSPORTAERIEN(idT, 'Paris', 'Madrid', dateDep, heure, 'avion', typeTrajet,  
confort),  
PROMOVACANCES(idV, prix, 'Paris', 'Madrid', 4, dateDep, heure,  
typeSejour, formule, 'avion', nom, nbEtoiles, restaurant, idT, idH).
```

3.2. Enrichissement des requêtes

L'enrichissement d'une requête exploite le profil de l'utilisateur pour reformuler sa requête en y intégrant des éléments de son centre d'intérêt ou de ses préférences. Cette technique d'enrichissement, courante dans les langages à mots clés en recherche d'information, est très récente en bases de données. La méthode la plus récente et la plus aboutie est celle de (Koutrika et al., 2004; 2005). Dans cette méthode, le profil de l'utilisateur est composé d'un ensemble de prédicats pondérés. Le poids d'un prédicat exprime son intérêt relatif pour l'utilisateur. Il est spécifié par un nombre réel compris entre 0 et 1.

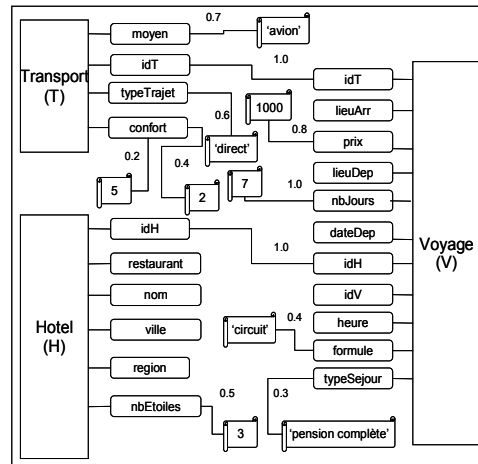
Dans l'exemple 4 ci-dessous, le profil décrit un utilisateur qui aime voyager en avion pour des séjours en pension complète de plus de 7 jours et pour un prix de moins de 1000 euro, préfère les vols directs dont le niveau de confort est entre 2 et 5, descend dans les hôtels de plus de 3 étoiles et n'aime pas les circuits touristiques. A chaque prédicat décrivant un élément du centre d'intérêt est associé un poids qui exprime son importance relative par rapport aux autres éléments de ce centre d'intérêt. Des préférences de jointure complètent le profil. Ce sont des prédicats de jointure orientés. Ces préférences expriment l'importance de la partie droite du prédicat si la partie gauche est incluse dans la requête. Le profil de l'utilisateur peut également être présenté sous forme d'un graphe où les nœuds sont de trois types (relations, attributs et valeurs) et les arcs correspondent aux prédicats du profil (Figure 1).

L'enrichissement d'une requête à l'aide d'un profil se fait en deux étapes : (i) recherche des prédicats de sélection pertinents (i.e. qui sont en relation avec la requête et qui ne sont pas contradictoires avec elle), et (ii) intégration de ces prédicats à la requête. Un prédicat est contradictoire avec la requête si, en l'ajoutant à celle-ci comme une conjonction, on obtient toujours un résultat nul. Par exemple, le prédicat « lieuArr='Venise' » est contradictoire avec le prédicat « lieuArr='Madrid' » dans Q_1 puisque ces deux prédicats ne peuvent pas être satisfaits simultanément.

Selon la méthode de (Koutrika et al., 2004), la recherche des prédicats pertinents consiste à trouver les chemins dans le graphe représentant le profil utilisateur, qui partent des nœuds attributs d'une relation apparaissant dans la requête et qui vont jusqu'aux nœuds valeurs. En prenant la requête utilisateur Q_1 de l'exemple 2, un tel chemin est celui représenté par les prédicats (b) et (g). Dans cet exemple, (g) est le prédicat pertinent et (b) permet de le lier à la requête Q_1 . Pour ajouter (g) à Q_1 , on doit ajouter également la relation HOTEL qui contient l'attribut sur lequel ce prédicat est exprimé. Dans ce cas le prédicat (b) permet de la joindre à VOYAGE qui est l'une des relations de Q_1 .

Exemple 4: Profil utilisateur P1 =

- ```
{
VOYAGE.idT=TRANSPORT.idT 1.0 (a)
VOYAGE.idH=HOTEL.idH 1.0 (b)
VOYAGE.nbJours>7 1.0 (c)
VOYAGE.prix<1000 0.8 (d)
TRANSPORT.moyen='avion' 0.7 (e)
TRANSPORT.typeTrajet='direct' 0.6 (f)
HOTEL.nbEtoiles>3 0.5 (g)
VOYAGE.formule<>'circuit' 0.4 (h)
TRANSPORT.confort>2 0.4 (i)
VOYAGE.typeSejour='pens. complete' 0.3 (j)
TRANSPORT.confort<5 0.2 (k)
}
```



**Figure 1.** Représentation textuelle et graphique d'un profil (Koutrika et al., 2004)

L'intégration des prédicats du profil à la requête est guidée par les trois paramètres suivants:  $K$  qui est le nombre de prédicats du profil devant être pris en compte selon leur poids par exemple;  $M$  qui est le nombre de prédicats parmi les  $K$  qui doivent *obligatoirement* être satisfaits (ça correspond aux  $M$  prédicats de plus grand poids parmi les top  $K$ ) et  $L$  qui est le nombre minimal de prédicats parmi les  $K-M$  restants que chaque tuple du résultat doit satisfaire.

Pour illustrer ce principe, considérons la requête  $Q_1$  de l'exemple 2 et fixons les paramètres  $K$ ,  $L$  et  $M$  respectivement à 5, 2 et 2 sur le profil  $P_1$  de l'exemple 4. Le premier pas de l'algorithme va sélectionner les 5 prédicats de plus grand poids de  $P_1$  qui ne sont pas contradictoires avec ceux de la requête; ce qui exclut le prédicat (c). Sur les prédicats sélectionnés  $\{d, e, f, g, h\}$ , les deux premiers (d et e) sont obligatoires et sont ajoutés à la requête  $Q_1$  qui devient  $Q'_1$  (Exemple 5).

**Exemple 5:** Requête initiale enrichie avec les prédicats obligatoires

```
Q'_1 = SELECT idV, prix, V.lieuDep, T.moyen, T.confort
FROM VOYAGE V, TRANSPORT T
WHERE V.idT = T.idT AND V.lieuArr='Madrid' AND V.nbJours=4
AND V.prix<1000 AND T.moyen='avion';
```

Finalement, la disjonction des conjonctions de deux prédicats parmi les prédicats non obligatoires (f, g et h) est ajoutée à la requête; ce qui donne la requête enrichie  $Q_{1+}$  (Exemple 6). On remarque que le prédicat (g) est exprimé sur la relation HOTEL qui n'est pas présente dans la requête initiale. Cette relation est ajoutée à la requête utilisateur ainsi que le prédicat de jointure (b) qui permet de la lier aux autres relations.

---

Exemple 6: Requête initiale enrichie

```
Q1+ = SELECT idV, prix, V.lieuDep, T.moyen, T.confort
FROM VOYAGE V, TRANSPORT T, HOTEL H
WHERE V.idT = T.idT AND V.idH = H.idH AND V.lieuArr='Madrid'
AND V.nbJours=4 AND V.prix<1000 AND T.moyen='avion'
AND ((H.nbEtoiles>3 AND V.Formule<>'circuit')
OR (H.nbEtoiles>3 AND T.typeTrajet='direct')
OR (V.Formule<>'circuit' AND T.typeTrajet='direct'));
```

---

#### 4. Approche de médiation personnalisée

Un système de médiation est un système d'intégration de données, qui offre un accès transparent à des sources de données distribuées et hétérogènes. Il est généralement défini par quatre composants: un schéma virtuel; un ensemble de liens sémantiques reliant ce schéma aux sources de données (appelés aussi requêtes de médiation) ; un module de réécriture de requêtes utilisateur ; un module d'intégration de données qui réalise les opérations multi sources (jointures, union, agrégat) à partir des résultats partiels calculés par les systèmes sources.

L'introduction de la personnalisation de l'accès dans une telle architecture nous conduit à l'enrichir. Tout d'abord, nous considérons que chaque utilisateur est décrit par un ou plusieurs profils définissant ses centres d'intérêts. D'autre part, l'évaluation d'une requête utilisateur doit se faire relativement à un profil. La requête ne traduit alors plus qu'une expression approchée du besoin de l'utilisateur.

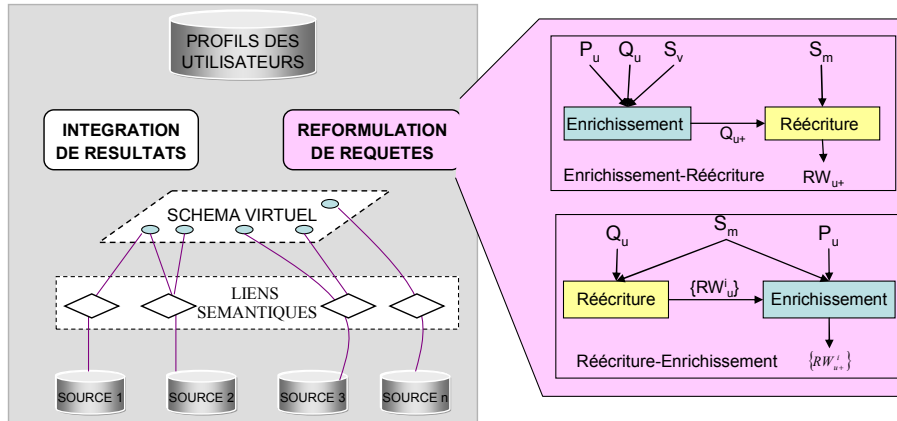
Dans notre contexte, le principal problème est la surcharge informationnelle de l'utilisateur : la quantité de résultats issus de l'évaluation de la requête est souvent très importante et peu pertinente. L'objectif de la personnalisation sera donc de réduire le nombre de résultats en ciblant de façon plus précise, à l'aide de son profil, les besoins de l'utilisateur.

La figure 2 décrit l'architecture d'un médiateur personnalisable. Le profil d'un utilisateur est décrit par plusieurs dimensions (Bouzeghoub et al., 2005) dont le centre d'intérêt de l'utilisateur, le contexte d'émission de la requête, le niveau de qualité désiré, l'historique des interactions ainsi que diverses préférences sur ces dimensions.

Chaque requête utilisateur est réécrite et enrichie en exploitant le profil de l'utilisateur et la définition des sources de données. Les sous-requêtes obtenues sont exécutées sur les sources et leurs résultats intégrés au niveau du médiateur (exécution des opérations multi-sources).

L'enrichissement et la réécriture sont des processus de reformulation de la requête utilisateur qui ont des objectifs différents. L'enrichissement permet de prendre en compte les préférences de l'utilisateur et de mieux cibler ses besoins tandis que la réécriture est faite dans le but d'accéder aux sources de données réelles. L'accès personnalisé à des sources de données multiples nécessite une étape de reformulation qui combine les deux techniques pour produire des requêtes exécutables sur les sources de données et qui intègre le profil utilisateur.





**Figure 2.** Architecture de médiation et zoom sur la reformulation d'une requête

Dans notre contexte, l'enrichissement et la réécriture ne sont pas indépendants. En effet, ces deux algorithmes peuvent ajouter des prédicats à la requête (prédicats du profil pour l'enrichissement et prédicats des liens sémantiques pour la réécriture). Or, comme leur comportement dépend des prédicats de la requête qu'ils reformulent, le résultat final dépend de l'ordre d'application des deux algorithmes.

La remarque précédente nous conduit à proposer et à analyser deux approches pour la reformulation d'une requête : la première où la réécriture s'applique sur le résultat de l'enrichissement, la seconde où l'enrichissement s'applique sur le résultat de la réécriture.

La problématique étudiée dans cet article étant l'étude et l'évaluation de la composition des algorithmes de réécriture et d'enrichissement, nous faisons deux hypothèses : (i) parmi les métadonnées décrivant les sources, seules leurs définitions sont utilisées et, (ii) les problèmes liés à l'hétérogénéité sont supposés résolus, i.e. les noms de relations ou d'attributs identiques correspondent à des concepts identiques et les noms différents correspondent à des concepts différents.

#### 4.1 Approche enrichissement – réécriture $\mathcal{R}(\mathcal{E})$

Cette approche consiste à enrichir d'abord la requête utilisateur à l'aide de son profil, sans tenir compte des sources de données et à rechercher ensuite les réécritures possibles de la requête enrichie.

Comme le montre l'exemple 6, la requête enrichie peut contenir une disjonction de prédicats. Or, les algorithmes de réécriture traitent des requêtes conjonctives. Afin d'obtenir des requêtes conjonctives, la requête enrichie peut être décomposée en plusieurs requêtes conjonctives, chacune contenant un terme de la disjonction. Chaque requête conjonctive ainsi obtenue est réécrite séparément.

Prenons par exemple le profil  $P_1$  de l'exemple 4, la requête utilisateur  $Q_1$  de l'exemple 2 et les valeurs des paramètres de l'algorithme d'enrichissement de la section 3.2 ( $K=5$ ,  $M=2$ ,  $L=2$ ).

La requête enrichie qu'on obtient est la requête  $Q_{1+}$  de l'exemple 6 qui peut être décomposée en 3 requêtes conjonctives (Exemple 7).

Exemple 7 : Décompositions de la requête enrichie  $Q_{1+}$  :

```

 Q_{1+}^1 = SELECT idV, prix, V.lieuDep, T.moyen, T.confort
 FROM VOYAGE V, TRANSPORT T, HOTEL H
 WHERE V.idT = T.idT AND V.idH = H.idH AND V.lieuArr='Madrid'
 AND V.nbJours=4 AND V.prix<1000 AND T.moyen='avion'
 AND H.nbEtoiles>3 AND V.Formule<>'circuit' ;

 Q_{1+}^2 = SELECT idV, prix, V.lieuDep, T.moyen, T.confort
 FROM VOYAGE V, TRANSPORT T, HOTEL H
 WHERE V.idT = T.idT AND V.idH = H.idH AND V.lieuArr='Madrid'
 AND V.nbJours=4 AND V.prix<1000 AND T.moyen='avion'
 AND H.nbEtoiles>3 AND T.typeTrajet='direct' ;

 Q_{1+}^3 = SELECT idV, prix, V.lieuDep, T.moyen, T.confort
 FROM VOYAGE V, TRANSPORT T, HOTEL H
 WHERE V.idT = T.idT AND V.idH = H.idH AND V.lieuArr='Madrid'
 AND V.nbJours=4 AND V.prix<1000 AND T.moyen='avion'
 AND V.Formule<>'circuit' AND T.typeTrajet='direct' ;

```

Prenons par exemple la requête  $Q_{1+}^1$ . Comparée à la requête initiale  $Q_1$ , elle contient une nouvelle relation virtuelle (HOTEL) pour la réécriture de laquelle il n'y a qu'une source contributive (HOTELS DUMONDE). La source SNCF doit être écartée des sources contributives de TRANSPORT parce que  $Q_{1+}^1$  contient le prédicat «T.moyen='avion'» qui n'est pas compatible avec la définition de SNCF (qui ne contient que des transports en train). On obtient ainsi quatre réécritures candidates pour  $Q_{1+}^1$  au lieu des six de la requête initiale  $Q_1$ . Les requêtes  $Q_{1+}^2$  et  $Q_{1+}^3$  se réécrivent de la même manière.

L'avantage de l'approche  $\mathcal{RE}$  est qu'elle permet d'enrichir d'abord la requête initiale avec tous les prédicats du profil utilisateur qui ne sont pas conflictuels avec celle-ci. Par contre, étant donné que certains prédicats peuvent être contradictoires avec les définitions des sources, il y a un risque d'éliminer toutes les réécritures candidates. Par exemple une requête enrichie avec le prédicat «VOYAGE.lieuDep = 'Toulouse'» ne peut pas être réécrite parce que ce prédicat est conflictuel avec les définitions des deux sources de voyages. Le second inconvénient de cette approche vient du fait que certains prédicats du profil utilisateur peuvent être satisfaits par les définitions des sources de toutes les réécritures candidates. En d'autres termes, dans ce cas, le prédicat du profil n'apporte rien. Eviter les prédicats déjà satisfaits par les sources permet donc de prendre en compte davantage de prédicats du profil. Par exemple le prédicat «VOYAGE.prix<1000» est satisfait par les définitions des deux sources de voyage et par conséquent toutes les réécritures candidates de  $Q_1$  le satisfont aussi.

#### 4.2 Approche réécriture – enrichissement $\mathcal{ER}$

L'idée principale de l'approche réécriture-enrichissement est d'effectuer la réécriture de la requête en premier, ce qui permet d'obtenir plusieurs sous-requêtes (réécritures) qui contiennent les prédicats de sélection des sources qu'elles interrogent en plus des prédicats de la requête utilisateur. Ensuite, chacune des réécritures est enrichie à l'aide du profil utilisateur.

Prenons par exemple les réécritures candidates de la requête initiale  $Q_1$  (exemple 2), le profil utilisateur  $P_1$  (exemple 4) et les valeurs  $K=5$ ,  $M=2$  et  $L=2$  pour l'algorithme d'enrichissement. Les ensembles de 5 prédicats de  $P_1$  qui peuvent être choisis pour enrichir les réécritures candidates ne sont pas les mêmes. Le prédicat (c) est contradictoire avec toutes les réécritures candidates parce qu'il est conflictuel avec la requête initiale et par conséquent n'est pas choisi pour l'enrichissement. De même le prédicat (d) est satisfait par la définition des sources de toutes les réécritures candidates et peut être écarté. Le prédicat (g) est exprimé sur l'attribut « nbEtoiles » qui n'est pas présent dans le schéma des sources choisies pour la réécriture et ne peut pas être utilisé pour l'enrichissement des réécritures candidates. Sur les 6 prédicats restants, (e) est contradictoire avec la définition de la source SNCF ( $S_3$ ) et est satisfait par celle de la source TRANSPORTAERIEN ( $S_2$ ) ce qui fait qu'il n'est pas choisi pour enrichir les réécritures candidates basées sur ces deux sources. Ce qui donne l'ensemble de 5 prédicats  $\{e, f, h, i, j\}$  qui servira à enrichir les deux réécritures candidates, basées sur  $S_4$  et l'ensemble  $\{f, h, i, j, k\}$  pour les réécritures candidates basées sur  $S_2$  et  $S_3$ .

L'avantage de l'approche  $\mathcal{E}(\mathcal{R})$  est qu'elle permet de considérer dans la phase d'enrichissement uniquement des prédicats du profil utilisateur qui ne sont pas contradictoires ou déjà satisfaits par la définition des sources. Elle a un inconvénient majeur qui vient du fait que la réécriture de la requête initiale fixe le schéma des relations qu'elle interroge et par conséquent il est impossible d'utiliser les prédicats du profil utilisateur qui sont exprimés sur des attributs non présents dans les schémas des sources choisies lors de la réécriture (par exemple le prédicat (g) de  $P_1$ ). Nous avons vu que l'algorithme d'enrichissement peut ajouter de nouvelles relations à la requête initiale s'il y a des prédicats qui sont exprimés sur leurs attributs. Lorsque la réécriture est faite en premier, ceci n'est plus possible et seuls les prédicats exprimés sur les attributs, présents dans les schémas des sources utilisées pour la réécriture de la requête initiale, peuvent être ajoutés.

## 5. Evaluation des deux approches

Dans la section précédente, nous avons présenté les deux approches de reformulation que nous proposons et discuté de façon informelle les effets de l'ordre d'application des opérations d'enrichissement et de réécriture. Cette section introduit deux métriques pour mesurer de façon plus formelle les différences entre les deux approches.

### 5.1 Définition des métriques

Nous souhaitons mesurer l'apport de chaque approche en considérant la proportion de prédicats du profil utilisée et l'utilité réelle des prédicats pris en compte. Le premier point nous conduit à définir une notion de *couverture* et le second une notion de *précision*.

Ces deux métriques sont habituellement utilisées pour mesurer les résultats restitués par l'exécution des requêtes et les comparer par rapport à un référentiel idéal constitué des résultats attendus par les utilisateurs. Dans notre cas, nous nous plaçons en amont de l'exécution et proposons des mesures qui tiennent compte uniquement de la sémantique des requêtes (expressions logiques des requêtes). Ces deux mesures sont évaluées par rapport au profil de

l'utilisateur (prédicats) et non pas par rapport aux résultats de l'évaluation de la requête reformulée (tuples).

### 5.1.1. Couverture des prédicats du profil utilisateur

Etant donnée une approche de reformulation de requête, la *couverture* a pour objectif de mesurer la proportion de prédicats du profil utilisables par cette approche pour l'enrichissement de la requête. Un prédicat est utilisable par une approche de reformulation s'il peut être utilisé dans la phase d'enrichissement de celle-ci.

Les poids des prédicats du profil utilisateur sont plus une estimation des préférences de l'utilisateur faite au moment de la construction du profil qu'une préférence exacte de l'utilisateur. Pour gommer les éventuelles imperfections des valeurs des poids des prédicats du profil, nous proposons de partitionner le profil en un ensemble de groupes tel que les prédicats de chaque groupe aient relativement la même importance pour l'utilisateur et chaque prédicat n'appartienne qu'à un seul groupe.

Soit  $G$  le partitionnement d'un profil utilisateur  $P_u$  en  $N$  groupes  $\{G_1, \dots, G_N\}$ , i.e.  $G_1 \cup \dots \cup G_N = P_u$  et  $\forall i, j \in 1..N, i \neq j, G_i \cap G_j = \emptyset$ . Les groupes de prédicats n'ont pas la même importance pour l'utilisateur. L'importance d'un groupe  $G_i$ , notée  $I_i$ , est fonction des poids des prédicats le composant (il est plus important de satisfaire les prédicats de plus grand poids) ainsi que du nombre de ces prédicats (il est plus intéressant de couvrir 5 prédicats sur 10 plutôt que 1 sur 2). Soient  $I_i^n$  et  $I_i^p$  les importances relatives d'un groupe de prédicats  $G_i$ , calculées respectivement par rapport au nombre et aux poids des prédicats de  $G_i$ . Dans notre évaluation, nous avons choisi comme fonction de calcul des  $I_i^n$  le pourcentage de prédicats dans chaque groupe par rapport au nombre total de prédicats dans le profil utilisateur et comme fonction de calcul des  $I_i^p$  la moyenne des poids  $w_j$  des prédicats de chaque groupe (en normalisant l'ensemble des moyennes obtenues) :

$$I_i^n = \frac{|G_i|}{|P_u|} \text{ et } I_i^p = \frac{AVG(G_i)}{\sum_{j=1}^N AVG(G_j)} \text{ avec } AVG(G_i) = \frac{\sum_{(p_j, w_j) \in G_i} w_j}{|G_i|}$$

En utilisant ces deux paramètres, on peut définir le poids  $I_i$  d'un groupe comme étant la somme pondérée de  $I_i^n$  et  $I_i^p$  :  $I_i = \frac{\alpha I_i^p + \beta I_i^n}{\alpha + \beta}$  où  $\alpha$  et  $\beta$  sont des constantes exprimant

l'importance relative de  $I_i^p$  et  $I_i^n$ . Par exemple si le poids des prédicats dans chaque groupe est deux fois plus important que leur nombre, alors  $\alpha=2$  et  $\beta=1$ .

Notons par  $C(P_1, P_2)$ , où  $P_1$  et  $P_2$  sont deux ensembles de prédicats, la couverture des prédicats de  $P_1$  par les prédicats de  $P_2$  :  $C(P_1, P_2) = \frac{|P_1 \cap P_2|}{|P_1|}$ .

L'ensemble de *prédicats utilisables* du profil  $P_u$  par une approche de reformulation  $F$  pour enrichir une requête  $Q_u$ , noté  $PU(P_u, Q_u, F)$ , contient les prédicats du profil utilisateur qui sont

exprimés sur des attributs présents dans les schémas des relations de la requête et qui ne sont pas contradictoires avec les prédicats déjà présents dans la requête.

**Définition 1 :** La *couverture*, notée CP, d'un profil utilisateur  $P_u$  représenté par la partition G, par l'ensemble de prédicats PU qu'une approche de reformulation  $F$  peut prendre en compte est égale à la somme pondérée des couvertures des différents groupes de G par PU :

$$CP(P_u, Q_u, F) = \sum_{i=1}^N I_i C(G_i, PU(P_u, Q_u, F))$$

### 5.1.2. Précision des prédicats utilisés pour l'enrichissement

Etant donnée une approche de reformulation  $F$ , la *précision* de cette approche sert à évaluer la proportion de prédicats réellement utiles pour l'enrichissement d'une requête utilisateur. Un prédicat  $p$  d'un profil utilisateur  $P_u$  n'est pas utile à l'enrichissement d'une requête  $Q_u$  si : (i)  $p$  est satisfait par l'ensemble des définitions des sources qui sont choisies pour réécrire  $Q_u$  ou (ii)  $p$  ne peut être satisfait par aucune source (c.-à-d.  $p$  est contradictoire avec la définition d'au moins une source de chaque réécriture de  $Q_u$ ).

Soit  $PPU(P_u, Q_u, F)$  l'ensemble de prédicats du profil utilisateur  $P_u$  *potentiellement utiles* à l'enrichissement d'une requête  $Q_u$  par une approche de reformulation  $F$ . Autrement dit, PPU représente les prédicats utilisables PU auxquels sont enlevés les prédicats inutiles que l'approche  $F$  est capable de détecter. Soit  $PRU(P_u, Q_u)$  l'ensemble de prédicats de  $P_u$  *réellement utiles* pour l'enrichissement de  $Q_u$ . PRU représente l'ensemble idéal de prédicat d'un profil  $P_u$  pouvant enrichir une requête  $Q_u$ , c'est-à-dire  $p \in PRU(P_u, Q_u)$  ssi (i)  $p$  peut être utilisé pour enrichir  $Q_u$  et (ii) l'ajout de  $p$  à  $Q_u$  n'empêche pas de réécrire  $Q_u$  et (iii)  $p$  n'est pas déjà satisfait par les définitions des sources choisies pour au moins une réécriture candidate de l'enrichissement de  $Q_u$  ou  $p$  est conflictuel avec la définition d'au moins une source contributive à la réécriture de l'enrichissement de  $Q_u$ . La première et la deuxième condition permettent d'identifier les prédicats qui peuvent être utilisés pour enrichir la requête  $Q_u$  tout en donnant un résultat non nul, tandis que la dernière condition assure que l'ajout de ce prédicat influe sur le résultat final de la requête. Un prédicat peut influencer sur le résultat de la requête soit en ajoutant une restriction supplémentaire aux résultats provenant d'une réécriture de  $Q_u$ , soit en éliminant des réécritures candidates.

**Définition 2 :** La *précision*, notée PR, des prédicats du profil utilisateur  $P_u$  utilisés par une approche de reformulation  $F$  pour l'enrichissement d'une requête  $Q_u$  est égale au pourcentage de prédicats réellement utiles parmi l'ensemble de prédicats jugés utiles par cette approche :

$$PR(P_u, Q_u, F) = \frac{|PRU(P_u, Q_u) \cap PPU(P_u, Q_u, F)|}{|PPU(P_u, Q_u, F)|}$$

La *couverture* et la *précision*, ainsi définies, permettent d'évaluer les approches de reformulation de requêtes et de montrer de façon plus objective leurs caractéristiques, même si ces dernières peuvent être appréhendées intuitivement. La section suivante décrit et discute les

résultats que nous avons obtenus en évaluant les deux approches de reformulation de la section 4 ( $\mathcal{E}(\mathcal{R})$ ,  $\mathcal{R}(\mathcal{E})$ ) et conclut sur leurs avantages et inconvénients respectifs.

## 5.2 Tests réalisés

Afin de montrer les limites des deux scénarios de reformulation de requêtes  $\mathcal{R}(\mathcal{E})$  et  $\mathcal{E}(\mathcal{R})$ , nous les avons appliqués sur un jeu de test réalisé en utilisant le schéma virtuel et les définitions des sources de l'exemple 1, et un échantillon de 4 profils et 10 requêtes en faisant varier le nombre de prédicats et de relations de médiation présentes dans les différentes requêtes. Plus de détails sur les tests réalisés peuvent être trouvés dans (Kostadinov et al., 2006). Les valeurs de couverture et de précision obtenues par les deux approches de reformulation  $\mathcal{E}(\mathcal{R})$  et  $\mathcal{R}(\mathcal{E})$  sont présentées dans le tableau 1.

Les résultats confirment l'analyse que nous avons faite dans la section précédente et montrent que la couverture obtenue par l'approche  $\mathcal{E}(\mathcal{R})$  est au plus égale à celle de l'approche  $\mathcal{R}(\mathcal{E})$  et que dans la plupart des cas, l'approche  $\mathcal{R}(\mathcal{E})$  possède une meilleure couverture que l'approche  $\mathcal{E}(\mathcal{R})$ . Ceci est dû principalement au fait que lorsque l'enrichissement de la requête initiale est fait en premier, tous les prédicats non contradictoires avec ceux de la requête peuvent être utilisés. Si un prédicat porte sur un attribut qui appartient à une relation non présente dans la requête, cette relation y est ajoutée à l'aide des prédicats de jointure. Cette extension de la portée de la requête est impossible dans l'approche  $\mathcal{E}(\mathcal{R})$  où l'enrichissement est fait sur un schéma fixe qui est celui des relations sources choisies. Ceci implique que les préférences exprimées sur des attributs non présents dans ce schéma ne peuvent pas être utilisées.

En comparant les deux approches selon la précision des prédicats du profil qu'elles utilisent, nous pouvons remarquer que la précision de l'approche  $\mathcal{E}(\mathcal{R})$  est toujours de 100%. En effet, dans ce cas l'enrichissement est fait sur les réécritures candidates de la requête initiale qui contiennent les prédicats des définitions des sources. On dispose donc de toutes les informations nécessaires pour l'identification des prédicats utiles (prédicats de la requête initiale, prédicats du profil utilisateur et prédicats des définitions des sources). En revanche, l'approche  $\mathcal{R}(\mathcal{E})$  ne permet pas de prendre en compte les définitions des sources de données dans le choix des prédicats du profil pour l'enrichissement de la requête utilisateur. Il y a deux types de prédicats inutiles du profil utilisateur non détectés par l'approche  $\mathcal{R}(\mathcal{E})$  : (i) les prédicats qui sont satisfaits par l'ensemble des réécritures candidates et (ii) les prédicats qui sont conflictuels avec la définition d'au moins une source de chaque réécriture candidate de la requête utilisateur.

| Profils<br>Requêtes | P1                         |                            | P2                         |                            | P3                         |                            | P4                         |                            |
|---------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
|                     | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ |
| Q1                  | 86.7                       | 62.3                       | 91.25                      | 91.25                      | 86.2                       | 86.2                       | 91.2                       | 74.4                       |
| Q2                  | 100                        | 51.3                       | 100                        | 80.5                       | 100                        | 78.8                       | 100                        | 68.2                       |
| Q3                  | 100                        | 89                         | 78.25                      | 78.25                      | 86.2                       | 86.2                       | 100                        | 100                        |
| Q4                  | 89                         | 64.7                       | 91.25                      | 91.25                      | 92.7                       | 92.7                       | 91.2                       | 74.4                       |
| Q5                  | 100                        | 42.3                       | 78.25                      | 58.75                      | 86.2                       | 65                         | 100                        | 68.2                       |
| Q6                  | 89                         | 75.7                       | 91.25                      | 91.25                      | 74                         | 74                         | 100                        | 83.2                       |
| Q7                  | 89                         | 42.3                       | 78.25                      | 58.75                      | 74                         | 52.8                       | 100                        | 68.2                       |
| Q8                  | 89                         | 55.7                       | 87                         | 67.5                       | 92.7                       | 71.5                       | 83.2                       | 68.2                       |
| Q9                  | 91                         | 46.7                       | 100                        | 80.5                       | 100                        | 78.8                       | 86.8                       | 71.85                      |
| Q10                 | 78                         | 78                         | 71.75                      | 71.75                      | 68.3                       | 68.3                       | 91.2                       | 91.2                       |

| Profils<br>Requêtes | P1                         |                            | P2                         |                            | P3                         |                            | P4                         |                            |
|---------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
|                     | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ | $\mathcal{R}(\mathcal{E})$ | $\mathcal{E}(\mathcal{R})$ |
| Q1                  | 87.5%                      | 100%                       | 88%                        | 100%                       | 57%                        | 100%                       | 91%                        | 100%                       |
| Q2                  | 88.9%                      | 100%                       | 82%                        | 100%                       | 67%                        | 100%                       | 92%                        | 100%                       |
| Q3                  | 86%                        | 100%                       | 75%                        | 100%                       | 57%                        | 100%                       | 91%                        | 100%                       |
| Q4                  | 86%                        | 100%                       | 75%                        | 100%                       | 50%                        | 100%                       | 90%                        | 100%                       |
| Q5                  | 88.9%                      | 100%                       | 88%                        | 100%                       | 71%                        | 100%                       | 92%                        | 100%                       |
| Q6                  | 100%                       | 100%                       | 86%                        | 100%                       | 80%                        | 100%                       | 100%                       | 100%                       |
| Q7                  | 100%                       | 100%                       | 88%                        | 100%                       | 83%                        | 100%                       | 100%                       | 100%                       |
| Q8                  | 100%                       | 100%                       | 90%                        | 100%                       | 86%                        | 100%                       | 100%                       | 100%                       |
| Q9                  | 86%                        | 100%                       | 80%                        | 100%                       | 63%                        | 100%                       | 90%                        | 100%                       |
| Q10                 | 100%                       | 100%                       | 75%                        | 100%                       | 50%                        | 100%                       | 100%                       | 100%                       |

**Tableau 1.** Couverture et précision des prédicats du profil

En conclusion, l'approche  $\mathcal{R}(\mathcal{E})$  permet de couvrir un plus grand nombre de prédicats du profil utilisateur. Cette approche prend en compte l'ensemble des prédicats utiles du profil utilisateur, mais peut utiliser des prédicats inutiles pour l'enrichissement. Son point critique est l'utilisation de prédicats qui ne peuvent pas être réécrits. A la différence de l'approche  $\mathcal{R}(\mathcal{E})$ , l'approche  $\mathcal{E}(\mathcal{R})$  n'utilise que des prédicats utiles, mais ne permet pas de les couvrir dans leur totalité. Cette approche manque tout prédicat exprimé sur des attributs non présents dans le schéma des sources choisies pour la réécriture de la requête initiale et ne permet pas de prendre en compte dans la phase d'enrichissement les « prédicats filtres » qui permettent d'éliminer un certain nombre de sources contributives pour la réécriture de la requête initiale.

Les deux scénarios de reformulation de requêtes ne sont pas contradictoires, mais complémentaires. Chacun d'entre eux permet d'atteindre des objectifs différents. Si le but est de satisfaire autant de préférences que possible et si tous les prédicats du profil peuvent être réécrits, alors l'approche  $\mathcal{R}(\mathcal{E})$  est préférable à  $\mathcal{E}(\mathcal{R})$ . Si au contraire, on veut un processus de reformulation performant en limitant le nombre de prédicats à considérer qui n'utilise que des prédicats utiles, alors l'approche  $\mathcal{E}(\mathcal{R})$  permet d'obtenir de meilleurs résultats.

Dans les deux cas, seule l'exécution effective des requêtes déterminera si l'intention obtenue par l'enrichissement correspond bien à l'intention de l'utilisateur. Cette étape fait l'objet de tests en cours.

## 6. Conclusion

Nous avons proposé deux approches de reformulation de la requête utilisateur dans un système multi-sources et nous avons discuté les avantages et les inconvénients de leur application. Cette analyse a été validée par une expérience menée sur un échantillon de profils et de requêtes. La validation par l'exécution effective des requêtes obtenues est en cours.

Nous sommes également en train d'élaborer une approche de reformulation mixte qui possède les avantages des deux approches ( $\mathcal{R}(\mathcal{E})$  et  $\mathcal{E}(\mathcal{R})$ ) et qui gomme leurs inconvénients.

L'architecture présentée dans cet article représente une base pour la personnalisation dans un système de médiation. Différentes extensions peuvent être envisagées. Par exemple, la prise en compte de profils de sources intégrant des mesures de qualité permettrait d'affiner le processus de reformulation. Un autre exemple pourrait être de tenir compte du *feedback* de l'utilisateur. Chacune de ces extensions constitue un problème ouvert et fait partie de nos objectifs de recherche dans le contexte du projet APMD<sup>1</sup>.

## 7. Bibliographie

- Bouzeghoub M., Kostadinov D., « Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils », *Dans les actes de la seconde Conférence en Recherche d'Informations et Applications (CORIA)*, Grenoble, France, 2005
- Calvanese D., Lembo D., Lenerini M., « Survey on methods for query rewriting and query answering using views », Technical report, University of Rome, Roma, Italy, April 2001.
- Duschka O., Genesereth M., « Answering Recursive Queries Using Views », *In Proceedings of the 16<sup>th</sup> ACM SIGACT-SIGMOD-SIGART Conference on Principles of Database Systems, PODS*, Tucson, AZ, 1997
- Kießling W., « Foundations of Preferences in Database Systems », *In Proceedings of the 28th Conference on Very Large Data Bases*, Hong Kong, China, 2002
- Kostadinov D., Bouzeghoub M., Lopes S., « Accès personnalisé à des sources de données multiples », Rapport interne, Laboratoire PRISM, Université de Versailles, France, 2006
- Koutrika G., Ioannidis Y. E., « Personalization of Queries in Database Systems », *In Proceedings of the 20th International Conference on Data Engineering*, Boston, Massachusetts, USA, April, 2004
- Koutrika G., Ioannidis Y. E., « Personalized Queries under a Generalized Preference Model », *In Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*, Tokyo, Japan, April 5-8, 2005
- Lacroix M., Lavency P., « Preference: Putting More Knowledge into Queries », *Proceeding of the 13th Very Large Data Bases Conference (VLDB)*, Brighton, 1987
- Levy A. Y., Rajaraman A., Ordille J. J., « Querying Heterogeneous Information Sources Using Source Descriptions », *In Proceedings of the 22nd Very Large Data Bases Conference*, Bombay, India, 1996.
- Naumann F., Freytag J.C., Spiliopoulou M., « Quality Driven Source Selection Using Data Envelope Analysis », *In Proceedings of the MIT Conference on Information Quality (IQ'98)*, Cambridge, USA, 1998.
- Rocacher D., Liétard L., « Préférences et quantités dans le cadre de l'interrogation flexible : sur la prise en compte d'expressions quantifiées », *Dans les actes des 22<sup>èmes</sup> Journées Bases de Données Avancées (BDA)*, Lille, France, 2006.

---

1. <http://apmd.prism.uvsq.fr/>