Accès personnalisé à des sources de données multiples

Dimitre Kostadinov, Mokrane Bouzeghoub

Laboratoire PRiSM, Université de Versailles
45, avenue des Etats-Unis, 78035 Versailles
{Prénom.Nom@prism.uvsq.fr}

Abstract : Les systèmes de médiation actuels permettent un accès transparent à un ensemble de sources de données hétérogènes. Ceci est fait en proposant aux utilisateurs un schéma global qui intègre l'ensemble des schémas sources par le biais d'un ensemble de requêtes de médiation. Généralement lorsque deux utilisateurs soumettent la même requête, ils reçoivent les mêmes réponses bien que leurs besoins et même leurs intentions soient différents. Le but de la personnalisation est de faciliter l'expression du besoin de l'utilisateur et de lui permettre d'obtenir des informations pertinentes lors de ses accès à un système d'information. Dans certaines approches elle se traduit par l'enrichissement de la requête utilisateur par un ensemble de prédicats contenus dans son profil. Dans un contexte de médiation, la personnalisation doit tenir compte non seulement du profil des utilisateurs mais aussi du contenu sémantique des sources décrit par les requêtes de médiation. Ce rapport décrit et évalue à travers un exemple deux approches de personnalisation qui peuvent être explorées dans ce contexte.

Mots clés : personnalisation, accès adaptatif, reformulation de requête

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. Cette limite a plusieurs conséquences pour l'utilisateur:

(i) les mêmes réponses sont fournies aux mêmes requêtes quels que soient les utilisateurs qui les ont émises: les systèmes se contentant de délivrer tous les objets satisfaisant strictement les critères de la requête;

(ii) la taille des réponses est souvent volumineuse et génère une surcharge informationnelle qui déroute ou décourage l'utilisateur dans son exploration ou sa navigation;

(iii) la pertinence des réponses se trouve souvent réduite dans la mesure où elles ne sont pas adaptées au contexte de l'utilisateur : les objets délivrés ne sont pas forcément en adéquation avec le lieu d'émission de la requête ou le terminal utilisé pour la requête;

(iv) la qualité de l'information délivrée est ignorée; ce qui laisse souvent l'utilisateur perplexe quant à son utilité et rend difficile la prise de décision à base de cette information.

Ces conséquences 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, qui désigne les objets auxquels il veut appliquer un certain traitement. Les seules variations admises sont celles liées aux mises à jour qui peuvent altérer les résultats d'une requête selon le moment de son exécution; variations admises car le monde réel de l'utilisateur est évolutif et la base de données est vue comme une succession d'états représentant ce réel.

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.

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

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 a toujours un administrateur qui maintient 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 (SRI). Les utilisateurs ne connaissent pas forcément les sources de données qu'ils interrogent, leur description leurs sont inaccessibles et ils ne savent même pas si l'information qu'ils recherchent existe ou non. En conséquence, leurs requêtes ne traduisent plus un besoin précis mais une intention qui doit être affinée en fonction des sources de données disponibles dans le système d'intégration 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 [Kieß 02], enrichissement de requêtes à l'aide de préférences définies dans un profil utilisateur [KoIo 04], sélection des sources de données en fonction de leur qualité [Naum 98]. Les deux 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 (centre d'intérêt, préférences, contexte d'exécution de la requête) et des profils des sources de données (méta données décrivant leurs contenus et leurs facteurs de qualité). En effet, les bénéfices de l'accès personnalisé à l'information sont plus visibles dans un contexte distribué où la multiplicité des sources de données conduit à des résultats volumineux, souvent non pertinents et redondants. C'est le cas des systèmes de médiation qui délivrent l'ensemble des résultats possibles collectés à partir des sources qui leur sont connectées, sans évaluation de leur pertinence par rapport aux préférences de l'utilisateur. Le problème peut être encore pire dans le cas des systèmes P2P où la requête de l'utilisateur est disséminée sur le réseau pour acquérir un maximum d'informations sans tenir compte d'autres critères que ceux exprimés dans la requête elle-même.

Ce rapport décrit le principe d'une approche de personnalisation de l'information dans un contexte de médiation à grande échelle où les sources de données sont évolutives et sujettes à des déconnexions temporaires ou permanentes. Dans ce contexte, l'évaluation d'une requête se fait en tenant compte, d'une part, du profil utilisateur qui enrichira son expression, et, d'autre part, des sources disponibles et de leur qualité au moment de l'évaluation de la requête. Nous présenterons et évaluerons en particulier deux approches de personnalisation:

- une approche *enrichissement-réécriture* (E-R) qui enrichit d'abord la requête de l'utilisateur à l'aide du profil de ce dernier avant de la réécrire sur les sources de données;
- une approche *réécriture-enrichissement* (R-E) qui effectue d'abord la réécriture de la requête utilisateur et introduit ensuite des enrichissements sur les réécritures résultantes.

Nous montrerons les différences entre les deux approches et expliciterons la pertinence de chacune sur un test modeste mais assez significatif pour montrer l'intérêt de la personnalisation dans un système de médiation. Pour ce faire, nous nous appuierons sur un algorithme de réécriture dans une approche LAV (Local As View) proposé par [LeOR 96] et sur un algorithme d'enrichissement de requêtes à l'aide d'un profil proposé par [KoIo 04]. Nous montrerons également comment le choix de ces deux algorithmes de base impacte les deux approches de personnalisation.

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 analyse et compare ces approches et montre les résultats que nous avons obtenus sur un échantillon de requêtes. La section 6 conclut le rapport 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 qui le relient aux sources de données participantes, appelés aussi requêtes de médiation. Pour la simplicité de l'exemple, nous nous limiterons à des sources de données relationnelles et nous nous plaçons dans un contexte LAV (Local As View). Dans la suite on utilisera S_v pour désigner le schéma virtuel et S_m pour l'ensemble de requêtes de médiation $\{S_1, \ldots, S_n\}$.

Exemple 1 : Schéma virtuel (S_v)

```
VOYAGE(idV,prix,lieu_depart, lieu_arrivee,
    nbre_jours, date_depart, heure,
    type_sejour, type_formule, idT, idH)
TRANSPORT(idT, moyen, type_trajet, comfort)
HOTEL(idH, nbre_etoiles, nom, region,
    ville, restaurant)
```

Notre exemple de système d'intégration de données traite des voyages, des moyens de transport et des hôtels qu'un voyageur peut réserver pour des séjours professionnels ou d'agréments. 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 suivantes :

- HOTELSDUMONDE : source contenant des hôtels,
- TRANSPORTAERIEN : source proposant des vols vers différentes destinations,
- SNCF : sources des trains régionaux ou internationaux de la SNCF,
- VOYAGERPARTOUT : source proposant des déplacements avec différents moyens de transport,
- PROMOVACANCES : compagnie proposant des voyages promotionnels au départ de Paris,
- LYONVACANCES : compagnie proposant des voyages au départ de Lyon.

Chaque source est décrite par une requête de médiation exprimée à la Datalog. L'exemple 2 montre la définition de la source TRANSPORTAERIEN. La partie gauche de la requête correspond au schéma de la source et la partie droite contient un atome pour chaque relation virtuelle invoquée (TRANSPORT et VOYAGE). Les requêtes de médiation des autres sources peuvent être trouvées en annexe.

Exemple 2 : Définition de la source TRANSPORTAERIEN

```
S2:TRANSPORTAERIEN(idT, lieu_depart,
   lieu_arrivee, date_depart, heure,
   moyen,type_trajet,comfort):-
   TRANSPORT(idT, 'avion', type_trajet,
      comfort),
   VOYAGE(idV, prix, lieu_depart,
      lieu_arrivee, nbre_jours,
      date_depart, heure, type_sejour,
      type_formule, idT, idH).
```

Les contraintes sur le contenu des sources sont exprimées en remplaçant un attribut par sa valeur. Par exemple, pour la définition de la source TRANSPORTAERIEN, l'attribut *moyen* est remplacé par la valeur 'avion' dans l'atome qui correspond à la relation virtuelle TRANSPORT.

Les requêtes de médiation ainsi définies vont servir à la réécriture des requêtes utilisateurs, avant ou après leur enrichissement par les profils utilisateurs.

Les sections suivantes rappellent brièvement les principes de la réécriture et de l'enrichissement à travers des algorithmes que nous avons choisis pour notre expérimentation.

3. Rappels des principes de réécriture et d'enrichissement des requêtes

Cette section rappelle les principes des 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.

3.1. Principe de réécriture des requêtes

La réécriture d'une requête dans une approche Local As View consiste à déterminer les sources contributives pour l'exécution de la requête utilisateur [BaHM 04] [LeOR 96] et à utiliser leurs définitions pour reformuler cette requête. Il existe deux classes principales d'algorithmes de réécriture: les algorithmes de règles inversées et les algorithmes à base de tas (Bucket-based) [BaHM 04]. Pour notre étude,

nous utilisons un algorithme de la dernière classe. Il fonctionne en deux phases [CaLL 01] :

- Création d'un tas (bucket) pour chaque atome *g* de la requête Q qui contient les vues (requêtes de médiation) contributives pour cet atome. Ce sont les vues à partir desquelles on peut obtenir des tuples de l'atome *g*.
- Construire des réécritures candidates et ne garder que les réécritures qui sont incluses dans la requête. Chaque réécriture candidate est une requête conjonctive obtenue en prenant une vue de chaque tas. 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.

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 3 : Requête initiale

```
Q1 = SELECT idV, prix, V.lieu_depart,
   T.moyen, T.comfort
FROM VOYAGE V, TRANSPORT T
WHERE V.idT = T.idT
       AND V.lieu_arrivee='Madrid'
       AND V.nbre_jours=4 ;
```

Soit $\mathcal{R}$ l'algorithme de réécriture de Q_1 sur les définitions de sources $S_m = \{S_1, \ldots, S_6\}$ ($\mathcal{R}(Q_1 / S_m)$), la première phase de réécriture de Q_1 correspond à la construction des 2 tas de requêtes de médiations contributives, $\{S_2, S_3, S_4\}$ et $\{S_5, S_6\}$ respectivement pour les relations virtuelles TRANSPORT et VOYAGE. La deuxième phase de $\mathcal{R}$ consiste à combiner les requêtes de médiation de chaque tas ; ce qui implique une réécriture RW_1 représentée par l'union de 6 requêtes conjonctives $w_1^1, \ldots, w_1^6$ correspondant aux 6 possibilités de choix de sources contributives :

$$RW_1 = w_1^1 / \{S_2, S_5\} \cup w_1^2 / \{S_2, S_6\} \cup w_1^3 / \{S_3, S_5\} \cup w_1^4 / \{S_3, S_6\} \cup w_1^5 / \{S_4, S_5\} \cup w_1^6 / \{S_4, S_6\}$$

où $w_1^i / \{S_j, S_k\}$ correspond à la i-ème réécriture de la requête Q_1 qui est faite avec les requêtes de médiation S_j et S_k. Par exemple $w_1^1 / \{S_2, S_5\}$ correspond à la première réécriture candidate de Q_1 qui est faite avec les requêtes de médiation S_2 et S_5 qui définissent respectivement les sources TRANSPORTAERIEN et PROMOVACANCES (Exemple 4).

Exemple 4 : Exemple de réécriture candidate de Q_1

$w_1^1 / \{S_2, S_5\} =$

```
w1^1 (idV,prix,lieu_depart,moyen,comfort):-
TRANSPORTAERIEN(idT, 'Paris', 'Madrid',
   date_depart, heure, 'avion',
   type_trajet, comfort),
PROMOVACANCES(idV, prix, 'Paris',
   'Madrid', 4, date_depart, heure,
   type_sejour, type_formule, 'avion',
   nom, nbre_etoiles, restaurant).
```

Chacune des 6 requêtes conjonctives qui composent la réécriture de Q_1, délivre des informations dont la sémantique est différente en fonction des prédicats de sélection qu'elle vérifie. Etant donné qu'un des objectifs de la personnalisation de l'information est de fournir à l'utilisateur des données pertinentes tout en réduisant leur volume, il peut être intéressant de considérer chaque requête conjonctive comme une réécriture candidate afin de mieux cibler les besoins des utilisateurs. Dans la suite, nous allons utiliser le terme de réécriture candidate pour désigner une requête conjonctive qui est faite en utilisant une requête de médiation contributive par but de la requête utilisateur. L'ensemble des réécritures candidates d'une requête Q_i sera noté W_i.

3.2. Principe d'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 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 Ioannidis [KoIo 04], qui nous servira de support dans ce rapport. 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.

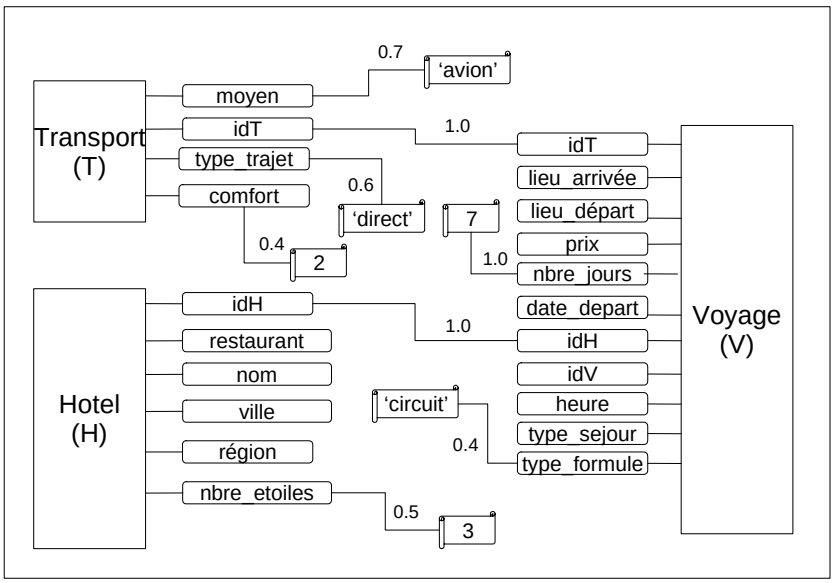

Figure 1 : Représentation graphique d'un profil utilisateur [KoIo 04]

Dans l'exemple 5 ci-dessous, le profil décrit un utilisateur qui aime voyager en avion pour des séjours de plus de 7 jours, préfère les vols directs dont le niveau de confort est supérieur à 3, 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.

Le profil de l'utilisateur peut également être présenté sous forme d'un graphe où les nœuds représentent soit des relations, soit des attributs, soit des valeurs, et les arcs représentent soit des sélections (entre un noeud d'attribut et un noeuds de valeur) soit des jointures (entre deux noeuds d'attribut). Les poids relatifs des prédicats sont représentés sur les arcs correspondants (Figure 1).

Exemple_5 : Profil utilisateur

```
P1 =
{  VOYAGE.idT=TRANSPORT.idT       1.0 (a)
   VOYAGE.idH=HOTEL.idH           1.0 (b)
   VOYAGE.nbre_jours>7            1.0 (c)
   TRANSPORT.moyen='avion'        0.7 (d)
   TRANSPORT.type_trajet='direct' 0.6 (e)
   HOTEL.nbre_etoiles>3           0.5 (f)
   VOYAGE.type_formule<>'circuit' 0.4 (g)
   TRANSPORT.comfort>2            0.4 (h)}
```

L'enrichissement d'une requête à l'aide d'un profil se fait en deux étapes : (i) recherche des prédicats 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 « lieu_arrivee='Venise' » est contradictoire avec le prédicat « lieu_arrivee='Madrid' » dans Q_1 puisque ces deux prédicats ne peuvent pas être satisfaits en même temps.

Selon la méthode de [KoIo 04], la recherche des prédicats pertinents consiste à chercher les chemins, dans le graphe représentant le profil utilisateur, qui partent des nœuds attributs d'une relation qui apparaît dans la requête et qui vont jusqu'aux nœuds valeurs. En prenant la requête utilisateur Q_1 de l'exemple 3, un tel chemin est celui représenté par les prédicats (b) et (f). Dans cet exemple, (f) est le prédicat pertinent et (b) permet de le lier à la requête Q_1. Pour ajouter (f) 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.

L'intégration des prédicats du profil à la requête est guidée par trois paramètres :

- Top K : nombre de prédicats du profil devant être pris en compte. La notion de Top K peut être exprimée de différentes manières : les K prédicats de plus grand poids, les prédicats dont le poids est supérieur à un seuil donné etc. Dans notre exemple, nous considérons les K prédicats de plus grand poids. On remarque que seuls les prédicats de sélection non contradictoires avec la requête sont pris en compte. Des prédicats de jointure sont ajoutés au cas où il faut ajouter une nouvelle relation à la requête.
- M : nombre de prédicats parmi les Top K

qui doivent *obligatoirement* être satisfaits ; ça correspond aux M prédicats de plus grand poids parmi les top K.

- L : nombre minimal de prédicats parmi les Top K-M restants que chaque tuple du résultat doit satisfaire.

Les grandes lignes du processus d'enrichissement de la requête utilisateur peuvent être résumées ainsi :

1. choisir les Top K prédicats,
2. ajouter les M prédicats de plus grand poids comme une conjonction aux prédicats de la requête utilisateur,
3. calcul de tous les ensembles possibles de L prédicats parmi les K-M restants dont la conjonction n'est pas contradictoire. On appellera L-combinaison un ensemble de L prédicats parmi les K-M.
4. ajouter la disjonction des conjonctions de L prédicats à la requête utilisateur.

Soit $\mathcal{E}$ l'algorithme d'enrichissement d'une requête Q_i, qui est exprimée sur un schéma S_v, avec le profil utilisateur P_j ($\mathcal{E}(Q_i / [P_j,S_v])$). Pour illustrer ce processus d'enrichissement des requêtes, fixons les paramètres K-L-M sur le profil P_1 de l'exemple 5 et considérons la requête Q_1 de l'exemple 3. Pour K=5, M=2 et L=2, 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 6).

Exemple 6 : Requête initiale enrichie avec les prédicats obligatoires

```
Q'1 =
SELECT idV, prix, V.lieu_depart, T.moyen,
   T.comfort
FROM VOYAGE V, TRANSPORT T
WHERE V.idT = T.idT
       AND V.lieu_arrivee='Madrid'
       AND V.nbre_jours=4
       AND T.moyen='avion'
       AND T.type_trajet='direct' ;
```

Finalement la disjonction des conjonctions de 2 prédicats parmi les prédicats non obligatoires (f, g et h) est ajoutée à la requête; ce qui donne la requête enrichie Q_{1+} (Exemple 7). On remarque que le prédicat (f) 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 7 : Requête initiale enrichie

```
Q1+ =
SELECT idV, prix, V.lieu_depart, T.moyen,
   T.comfort
FROM VOYAGE V, TRANSPORT T, HOTEL H
WHERE V.idT = T.idT AND V.idH = H.idH
       AND V.lieu_arrivee='Madrid'
       AND V.nbre_jours=4
       AND T.moyen='avion'
       AND T.type_trajet='direct'
       AND ((H.nbre_etoiles>3 AND
             V.type_formule<>'circuit')
         OR (H.nbre_etoiles>3 AND
             T.comfort>2)
         OR (V.type_formule<>'circuit'and
             T.comfort>2) );
```

Les algorithmes d'enrichissement et de réécriture de la requête utilisateur peuvent être composés dans le but de personnaliser l'accès à l'information dans un système multi-sources. La section suivante présente l'approche de médiation personnalisée avec ses différentes variantes.

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:

(i) Un schéma virtuel décrivant les besoins métiers des applications qui utiliseront ce médiateur ;

(ii) Un ensemble de liens sémantiques reliant ce schéma métier aux sources de données (requêtes de médiation) ;

(iii) Un module de réécriture de requêtes qui reformule les requêtes utilisateur exprimées sur le schéma virtuel en requêtes exprimées sur les sources de données ;

(iv) 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.

Un mécanisme de vue supplémentaire peut être ajouté pour restreindre, le cas échéant, la vue de chaque utilisateur sur une partie du schéma virtuel.

Nous adoptons cette architecture en la plaçant dans un contexte de médiation à grande échelle où l'autonomie des sources de données est

totale. Ce qui signifie que ces sources sont indépendantes les unes des autres et sont définies indépendamment des systèmes de médiation auxquels elles participent : elles peuvent évoluer dans leur définition et leur qualité sans être contraintes par ces systèmes de médiation et leur disponibilité maintenue ou rompue selon des règles ou des événements propres à chaque source. La prise en compte de ces contraintes se traduit par un test de validité des liens sémantiques reliant le schéma virtuel à ses sources à chaque évaluation d'une requête utilisateur.

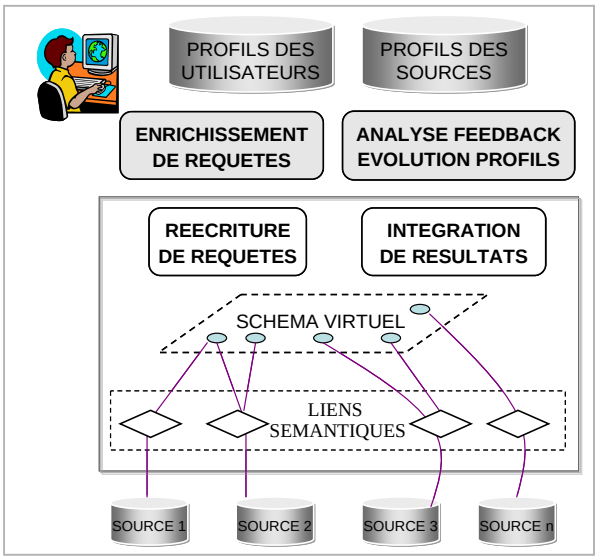

Figure 2 : Système de médiation

L'introduction de la personnalisation de l'accès dans une telle architecture impose les hypothèses complémentaires suivantes :

- chaque utilisateur est décrit par un ou plusieurs profils définissant ses centres d'intérêt et ses préférences ; ces profils sont exprimés sur le schéma de médiation qui joue le rôle de profil communautaire à un ensemble d'utilisateurs;
- chaque requête utilisateur est évaluée relativement à un profil ; de ce fait la requête ne traduit plus qu'une expression approchée du besoin de l'utilisateur;
- le module de réécriture du médiateur doit être capable de fournir des réécritures partielles si certains liens sémantiques ne sont plus valides en raison d'une évolution ou d'une déconnexion d'une ou plusieurs sources de données;
- le médiateur doit inclure un nouveau module qui analyse le 'feedback', explicite ou implicite, des utilisateurs et met à jour leurs profils ou le profil communautaire (le schéma virtuel et ses liens sémantiques avec les sources).

La figure 2 décrit l'architecture d'un médiateur personnalisable. Les composants gris sont les compléments aux composants classiques d'une architecture de médiation représentés par les boites blanches. La base de profils des sources décrit les méta données caractérisant chaque source de données (structure de données, contraintes d'intégrité, facteurs de qualité, événements d'évolution, ...). Le profil d'un utilisateur est décrit par plusieurs dimensions [BoKo 05] 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.

Dans ce contexte architectural, l'évaluation d'une requête utilisateur se fait selon le cycle de vie de la figure 3. Chaque requête utilisateur est reformulée et enrichie en exploitant, d'une part, le profil de l'utilisateur et, d'autre part, les profils des sources de données. Les sous-requêtes obtenues sont exécutées sur les sources et leurs résultats intégrés au niveau du médiateur. La personnalisation intervient à chacune des étapes de ce cycle de vie, y compris dans la présentation des résultats.

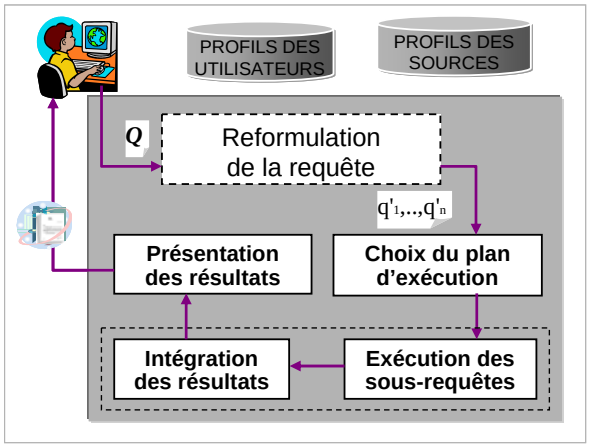

Figure 3 : Cycle de vie d'une requête personnalisée

La reformulation de requêtes peut se faire selon deux approches:

- Une approche enrichissement-réécriture (Figure 4) qui exploite d'abord le profil utilisateur pour enrichir sa requête avant de considérer les réécritures sur les sources de données. Dans ce cas, l'enrichissement tient compte uniquement du profil utilisateur et du schéma virtuel ;

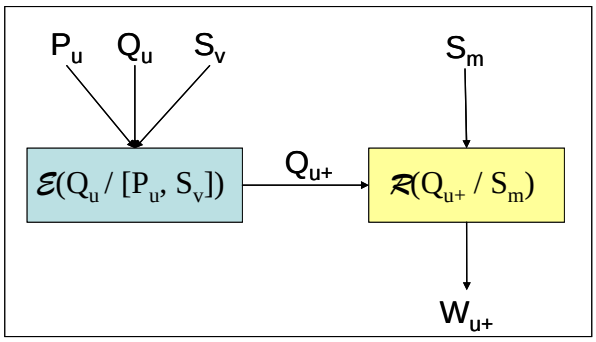

Figure 4 : Approche Enrichissement-Réécriture

- Une approche réécriture-enrichissement (Figure 5) qui identifie d'abord les sources pertinentes par la réécriture avant d'enrichir chacune des réécritures par le profil utilisateur. Dans ce cas, l'enrichissement tient compte du profil utilisateur et des méta données de chaque source.

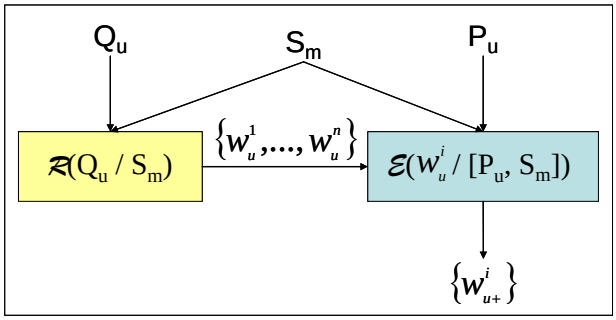

Figure 5 : Approche Réécriture-Enrichissement

Nous nous limitons dans ce rapport à l'étude de ces deux approches. L'effet de la personnalisation sur la définition du plan d'exécution, sur l'exécution effective des requêtes et sur la présentation des résultats dépasse le cadre de ce rapport.

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 récriture 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 de la requête initiale qui combine les deux techniques pour produire des requêtes exécutables sur les sources de données et qui intègre l'aspect adaptatif.

L'enrichissement et la réécriture ne sont pas complètement indépendants. Les deux algorithmes ajoutent des contraintes à la requête (prédicats du profil pour l'enrichissement et prédicats des requêtes qui définissent le contenu des sources utilisées pour la réécriture) et leur comportement dépend des prédicats de la requête qu'ils reformulent ; ce qui rend les résultats obtenus dépendant de l'ordre d'application des deux algorithmes. Cette section présente les deux approches de composition possibles.

4.1. Approche enrichissement – réécriture

Le premier scénario consiste à enrichir d'abord la requête utilisateur à l'aide de son profil, sans tenir compte des sources et ensuite à rechercher les réécritures possibles de la requête enrichie. L'objectif visé par cette approche est de prendre en compte les préférences les plus importantes pour l'utilisateur. En effet, étant donné que l'enrichissement dans ce cas est fait sur la requête initiale, on y ajoute les Top K prédicats non contradictoires avec elle. On obtient ainsi l'enrichissement le plus pertinent en fonction du profil utilisateur et de la requête initiale. On peut formaliser cette approche comme suit :

$$\mathcal{R}(\mathcal{E}(Q_u / [P_u,S_v]) / S_m)$$

$\mathcal{E}(Q_u / [P_u,S_v])$ délivre l'enrichissement Q_{u+} de la requête Q_u par rapport au profil utilisateur P_u et au schéma virtuel S_v. $\mathcal{R}(Q_{u+} / S_m)$ est la réécriture de la requête enrichie Q_{u+} par rapport à l'ensemble S_m des définitions LAV des sources de données.

Nous avons vu que l'algorithme d'enrichissement peut ajouter de nouvelles relations virtuelles à la requête initiale, ce qui se traduit par la construction de nouveaux tas de sources contributives pendant la phase de réécriture (un nouveau tas pour chaque nouvelle relation virtuelle). Comme la recherche des réécritures candidates est un problème combinatoire, chaque nouveau tas qui contient plus d'une source contributive multiplie le nombre de réécritures candidates à explorer par la cardinalité de ce tas. Cependant, le nombre de réécritures candidates obtenues par l'approche E-R n'est pas toujours supérieur à celui du scénario R-E. Le fait d'ajouter des prédicats supplémentaires à la requête utilisateur sans prendre en compte les possibilités de réponse des sources permet d'augmenter la pertinence des résultats obtenus, mais peut éliminer des sources contributives pour sa réécriture. Prenons par exemple le profil P_1 et la requête utilisateur Q_1 de l'annexe, également utilisés en section 3. La requête enrichie Q_{1+} contient une nouvelle relation virtuelle (HOTEL) dont la réécriture ne possède qu'une source contributive (HOTELSDUMONDE) ; ce qui fait que le nombre de combinaisons de sources possibles n'est pas influencé par les relations ajoutées. Par contre, la source SNCF doit

être écartée des sources contributives de TRANSPORT parce que Q_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 au lieu des six de la requête Q_1:

$$W_{1+} = \{ w^1_{1+} / \{S_1, S_2, S_5\},\ w^2_{1+} / \{S_1, S_2, S_6\},$$
$$w^3_{1+} / \{S_1, S_4, S_5\},\ w^4_{1+} / \{S_1, S_4, S_6\} \}$$

Dans certains cas il est possible que la requête enrichie ne puisse pas être réécrite parce qu'il n'y a pas de sources contributives pour une des relations virtuelles. Si, par exemple on remplace dans P_1 le prédicat (c) par « VOYAGE.lieu_depart='Toulouse' », après enrichissement, on obtient une requête qui contient ce prédicat et qui ne peut pas être réécrite parce qu'il est contradictoire avec les définitions des deux sources de voyages. Une solution possible à ce problème est de prendre en compte les contraintes sur le contenu des sources avant de choisir les prédicats pour la phase d'enrichissement. Dans ce cas on inverse l'ordre de l'application des deux algorithmes ce qui revient à enrichir les réécritures possibles de la requête initiale. La section suivante décrit cette approche.

4.2. Approche réécriture – enrichissement

L'objectif de cette approche est de s'assurer que chaque prédicat qui est ajouté à la requête de l'utilisateur pendant la phase d'enrichissement est exécutable i.e. il y a potentiellement des résultats qui le satisfont. L'idée principale 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 et des profils des sources correspondantes (limités à leurs schémas et leurs définitions LAV dans notre cas). Cette approche peut être formalisée par l'expression suivante.

$$\mathcal{E}(\mathcal{R}(Q_u / S_m) / [P_u, S_m])$$

$\mathcal{R}(Q_u / S_m)$ délivre l'ensemble des réécritures W_u de la requête Q_u par rapport à l'ensemble S_m des définitions LAV des sources de données. $\mathcal{E}(W_u / [P_u, S_m])$ délivre les enrichissements de l'ensemble W_u de réécritures de Q_u avec le profil P_u.

Pour pouvoir enrichir les réécritures candidates, nous faisons l'hypothèse que le nom de chaque attribut est unique et le même dans tous les schémas. Cette hypothèse simplificatrice permet de ne pas aborder les problèmes linguistiques inhérents à un système d'intégration de sources de données hétérogènes.

Un premier constat que nous pouvons faire est que les prédicats qui sont utilisés pour l'enrichissement d'une réécriture candidate peuvent être moins pertinents que ceux utilisés pour enrichir la requête initiale. Si un ou plusieurs prédicats du profil, parmi les Top K qui ne sont pas contradictoires avec la requête utilisateur, sont en contradiction avec les prédicats des sources, ils seront remplacés par les prédicats qui les suivent dans le profil (à condition d'avoir plus de prédicats dans P_u). Les nouveaux prédicats ont un poids au plus égal à celui du prédicat le moins « intéressant » parmi le Top K initial. Autrement dit, on choisit des prédicats dont la position dans le profil utilisateur est plus basse par rapport au dernier prédicat qui aurait été choisi avant la réécriture. D'un autre coté, si au lieu d'être contradictoires avec la réécriture, les prédicats sont remplacé parce qu'ils sont satisfaits par la définition des sources utilisées, on va obtenir au final des requêtes plus pertinentes parce qu'elles vérifient plus de prédicats du profil utilisateur.

Le risque que crée la prise en compte des contraintes des sources avant l'enrichissement est que le profil ne puisse pas être pris en compte. Si la définition des sources est trop riche (i.e. contient beaucoup de prédicats) et que par conséquent, après réécriture, tous les prédicats du profil sont soit contradictoires soit satisfaits par les sources utilisées, alors l'enrichissement n'apporte aucune information supplémentaire. Le résultats obtenu est le même que si on applique uniquement la réécriture. Or l'objectif de l'accès personnalisé est de cibler mieux les besoins de l'utilisateur.

Un problème supplémentaire vient du fait que les sources utilisées pour les différentes réécritures candidates n'ont pas les mêmes contraintes sur leur contenu et de ce fait il se peut qu'elles ne soient pas enrichies avec les mêmes prédicats. Par conséquent, les requêtes reformulées finales n'auront pas la même pertinence. Prenons par exemple les réécritures candidates de Q_1. Celle qui est faite avec les requêtes de médiation des sources TRANSPORTAERIEN et PROMOVACANCES ($w^1_1 / \{S_2, S_5\}$) possède une contrainte sur le moyen de transport (« moyen='avion' ») ce qui

fait que le prédicat (d) ne peut pas être utilisé pour son enrichissement tandis qu'il fait partie des top K prédicats qui peuvent enrichir la réécriture $w_1^5 / \{S_4, S_5\}$ qui est faite avec les définitions de VOYAGERPARTOUT et PROMOVACANCES. L'évaluation du degré de pertinence des requêtes obtenues dépasse le cadre du rapport et fait partie de nos objectifs.

Bien que cette approche permette d'obtenir des enrichissements exécutables, elle a un inconvénient majeur qui vient du fait que la réécriture de la requête initiale fixe le schéma des relations qu'elle interroge. 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. Par exemple un prédicat exprimé sur l'attribut *region* ne peut être utilisé pour l'enrichissement d'aucune réécriture candidate w_1^i de Q_1.

Nous venons de voir qu'il y a deux scénarios de reformulation de la requête utilisateur. La section suivant discute les résultats que chacune d'elles permet d'obtenir.

5. Analyse et comparaison des deux approches

Le résultat de chacune des deux approches dépend des dépendances (contradiction, inclusion etc.) qui existent entre les prédicats de la requête initiale, ceux du profil utilisateur et ceux des définitions des sources. Cette section discute les avantages et les inconvénients respectifs des deux approches en fonction des différents cas de figure qui peuvent se présenter. Cette analyse est faite sur la base d'un test qui est décrit dans la section suivante.

5.1. Tests réalisés

Pour valider l'analyse des deux approches de reformulation de la requête utilisateur, nous avons réalisé un test sur un système dont le schéma virtuel est celui de l'exemple 1 et les définitions des sources sont celles de l'annexe. Nous avons créé de façon aléatoire 10 000 tuples de voyages répartis uniformément dans les deux sources PROMOVACANCES et LYONVACANCES. Nous avons utilisé un échantillon de 4 profils utilisateur et de 10 requêtes données en annexe.

Pour l'étape d'enrichissement des requêtes dans les deux approches nous avons utilisé les mêmes valeurs pour les paramètres K, M et L. A chaque fois les Top 5 prédicats sont pris en compte en considérant que les 2 premiers sont obligatoires et que chaque tuple résultat doit satisfaire au minimum 2 prédicats optionnels (K=5, M=2 et L=2).

Les quantités de résultats obtenus par les deux approches pour chaque profil et chaque requête sont présentées dans le tableau 1. Les résultats de ce test ont démontré qu'aucune des deux approches n'est meilleure que l'autre dans l'absolue. Dans certains cas le scénario E-R est plus réducteur que R-E (par exemple P_3 et Q_5), dans d'autres c'est l'inverse qui se produit (ex. P_4 et Q_2) et parfois les deux approches sont équivalentes (P_1 et Q_4). Leur comportement dépend des prédicats du profil utilisateur, de la requête initiale et des définitions des sources de données. La section suivante présente les conclusions de l'analyse des résultats obtenus par ce test

Requête	Profils	Nombre de résultats avec profils											
		P_1			P_2			P_3			P_4		
	Nombre de résultats sans profils	E-R	E-R ∩ R-E	R-E	E-R	E-R ∩ R-E	R-E	E-R	E-R ∩ R-E	R-E	E-R	E-R ∩ R-E	R-E
Q_1	152	5	0	0	0	0	0	0	0	0	7	1	27
Q_2	10000	451	80	80	425	78	155	312	291	391	804	18	41
Q_3	2121	116	20	20	48	11	49	276	124	277	0	0	0
Q_4	1529	74	74	74	15	4	12	72	29	29	91	1	7
Q_5	1702	44	19	19	55	55	103	137	57	260	38	1	2
Q_6	573	18	12	20	14	14	16	87	87	87	19	1	9
Q_7	1256	39	0	0	11	11	48	153	153	270	0	0	6
Q_8	3418	156	0	0	67	67	67	104	104	104	94	3	14
Q_9	2565	0	0	0	124	73	151	183	183	391	435	120	250
Q_{10}	12	0	0	0	0	0	0	4	4	4	0	0	5

Tableau 1 : Récapitulatif du nombre de résultats des tests

5.2. Discussion sur les résultats des deux approches

De façon générale, l'approche E-R permet de prendre en compte un plus grand nombre de préférences du profil utilisateur. Lorsque l'enrichissement de la requête initiale est fait en premier, tous les prédicats du profil sont potentiellement utilisables i.e. sont exprimés sur des attributs qui sont présents dans le schéma sur lequel est exprimée la requête. 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 R-E 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. Pire encore, supposons que pour chaque réécriture candidate w_u^i de la requête utilisateur Q_u et chaque prédicat p_j du profil utilisateur P_u, une des trois conditions suivantes soit vérifiée :

- p_j est satisfait par la définition des sources utilisées pour la réécriture w_u^i
- p_j est contradictoire avec la définition des sources de w_u^i
- p_j est exprimé sur un attribut qui n'apparaît pas dans le schéma des sources de w_u^i

Dans ce cas, aucun prédicat du profil ne peut être utilisé pour l'enrichissement des réécritures candidates. Par conséquent l'approche R-E ne personnalise pas les résultats contrairement au scénario E-R qui permet de prendre en compte les prédicats qui vérifient une des deux dernières conditions.

Dans l'approche E-R, ajouter à la requête utilisateur des prédicats contradictoires avec les définitions des sources, permet de réduire le nombre de sources contributives à la réécriture de celle-ci. L'inconvénient de cette propriété est le risque d'obtenir une requête qui ne peut pas être réécrite (si on élimine l'ensemble des sources contributives pour un but). Par contre si le nombre de sources disponibles est trop important et si les prédicats de la requête ne sont pas très restrictifs, la réduction du nombre de sources à prendre en compte peut être bénéfique, constituant ainsi un filtre supplémentaire qui réduit la quantité des résultats.

Nous avons vu précédemment que, de façon générale, l'approche E-R permet de prendre en compte les préférences les plus importantes du profil utilisateur. Cependant si les conditions suivantes sont vérifiées, l'approche R-E génère des requêtes qui vérifient d'avantage de prédicats du profil :

- tous les Top K prédicats choisis pour l'enrichissement de la requête initiale Q_u sont exprimés sur des attributs présents dans le schéma des sources qui peuvent être utilisées pour sa réécriture,
- aucun prédicat du Top K initial n'est contradictoire avec la définition des sources des réécritures candidates,
- au moins un prédicat du Top K initial est satisfait par les définitions des sources qui sont choisies pour les réécritures candidates,
- parmi les prédicats du profil en dehors du Top K initial, il y a certains qui peuvent être utilisés pour l'enrichissement des réécritures W_u de Q_u (ne sont pas contradictoires avec les définitions des sources et sont exprimés sur des attributs présents dans leurs schémas).

Dans cette situation, l'enrichissement des réécritures candidates obtenu par l'approche R-E, est fait avec les prédicats du Top K initial qui ne sont pas satisfaits par les définitions des sources choisies, complétés avec les meilleures préférences du profil parmi celles qui restent et qui peuvent être utilisées. Il est important de remarquer que les requêtes obtenues après enrichissement prennent en compte l'ensemble des prédicats du Top K initial parce que les prédicats remplacés sont satisfaits par la définition des sources. Les résultats de l'approche R-E vérifient un plus grand nombre de préférences par rapport à ceux du scénario E-R et on peut considérer qu'ils sont plus pertinents pour l'utilisateur. Un cas extrême est que l'ensemble des prédicats du Top K initial soient satisfaits par la définition de l'ensemble des réécritures candidates ce qui implique que l'approche E-R n'apporte aucune personnalisation des résultats par rapport à une exécution classique de la requête.

Bien que les objectifs et le fonctionnement des deux approches soient différents, il se peut qu'elles produisent les mêmes résultats. Ceci est possible si l'ensemble des Top K prédicats choisis pour l'enrichissement est le même dans les deux approches et si les réécritures candidates sont faites avec les mêmes combinaisons de requêtes

de médiation. Une telle situation se produit si aucun prédicat du Top K initial ne vérifie une des trois conditions citées plus haut. Dans ce cas le choix des sources contributives pour la réécriture ne dépend que des prédicats de la requête initiale et les préférences du Top K initial peuvent être utilisés pour la phase d'enrichissement quelque soit l'approche de reformulation (E-R ou R-E).

Un deuxième cas d'égalité des deux approches apparaît si : (i) les prédicats obligatoires de l'enrichissement ainsi que les combinaisons des requêtes de médiation de la réécriture sont les mêmes dans les deux approches et (ii) il n'y a pas d'autres prédicats du profil qui peuvent remplacer ceux qui sont contradictoires ou satisfaits par les définitions des sources dans le scénario R-E. L'hypothèse de l'égalité des prédicats obligatoires garantie que les résultats des deux approches vérifient les mêmes prédicats. Dans le cas contraire, si un prédicat obligatoire est satisfait par les requêtes de médiation d'une réécriture candidate, il sera remplacé par un des prédicats optionnels ce qui fait que les résultats obtenus par la requête enrichie vont vérifier au minimum M+L+1 prédicats au lieu des M+L initialement.

En résumé, l'approche E-R est orienté vers la satisfaction des préférences les plus importantes pour l'utilisateur et permet de prendre en compte plus de prédicats de son profil, excepté dans les cas particuliers cités dans cette section. Alors que l'approche R-E permet de garantir l'exécutabilité des requêtes obtenues après reformulation.

6. Conclusion

Nous avons présenté deux approches de reformulation de la requête utilisateur dans un système multi-sources et nous avons discuté leurs avantages et les risques de leur application. Cette analyse a été validée par une expérience menée sur un échantillon de profils et de requêtes.

Une première perspective de travail soulevée par cette étude est l'élaboration d'une approche de reformulation de la requête utilisateur qui possède les avantages des deux scénarios (E-R et R-E) et qui gomme leurs inconvénients. Cette reformulation constitue le premier composant d'un système d'information multi-sources à accès personnalisé. La conception d'un tel système passe par la satisfaction des hypothèses que nous avons faites sur l'introduction de la personnalisation dans une architecture de médiation de la section 4. Chacune de ces hypothèses constitue un problème ouvert et fait partie de nos objectifs de recherche dans le contexte du projet APMD[1].

Références :

[BaHM 04] Q. Bai, J. Hong, M. F. McTear, Improving View Selection in Query Rewriting Using Domain Semantics, In Proceedings of the 6th International Conference on Enterprise Information Systems, Porto, Portugal, 2004

[BoKo 05] M. Bouzeghoub, D. Kostadinov, 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'Infomations et Applications (CORIA), Grenoble, France, 2005

[CaLL 01] D. Calvanese, D. Lembo, and M. Lenerini, Survey on methods for query rewriting and query answering using views, Technical report, University of Rome, Roma, Italy, April 2001.

[Kieß 02] W. Kießling, Foundations of Preferences in Database Systems, In Proceedings of the 28th Conference on Very Large Data Bases, Hong Kong, China, 2002

[KoIo 04] G. Koutrika, Y. E. Ioannidis, Personalization of Queries in Database Systems, In Proceedings of the 20th International Conference on Data Engineering, Boston, Massachusetts, USA, April, 2004

[LeOR 96] A. Y. Levy, A. Rajaraman, J. J. Ordille, Querying Heterogeneous Information Sources Using Source Descriptions, In Proceedings of the 22nd Very Large Data Bases Conference, Bombay, India, 1996.

[Naum 98] F. Naumann, J.C. Freytag, M. Spiliopoulou, Quality Driven Source Selection Using Data Envelope Analysis. In Proceedings of the MIT Conference on Information Quality (IQ'98), Cambridge, USA, 1998

NB : Cf annexe pages suivantes

[1] http://apmd.prism.uvsq.fr/

Annexe

Schéma virtuel :

```
VOYAGE(idV, prix, lieu_depart, lieu_arrivee, nbre_jours, date_depart, heure, type_sejour,
    type_formule, idT, idH)

TRANSPORT(idT, moyen, type_trajet, comfort)

HOTEL(idH, nbre_etoiles, nom, region, ville, restaurant)
```

Profils utilisateurs :

Profil P1 :

{VOYAGE.idT=TRANSPORT.idT	1.0
VOYAGE.idH=HOTEL.idH	1.0
VOYAGE.nbre_jours>7	1.0
TRANSPORT.moyen='avion'	0.7
TRANSPORT.type_trajet='direct'	0.6
HOTEL.nbre_etoiles>3	0.5
VOYAGE.type_formule<>'circuit'	0.4
TRANSPORT.comfort>2	0.4 }

Profil P2 :

{VOYAGE.idT=TRANSPORT.idT	1.0
VOYAGE.lieu_depart='Paris'	1.0
VOYAGE.lieu_arrivee='Madrid'	0.9
VOYAGE.lieu_arrivee='Barcelone'	0.7
TRANSPORT.moyen='avion'	0.7
VOYAGE.type_formule='week end'	0.6
VOYAGE.date_depart='05/05/2006'	0.6
TRANSPORT.moyen='train'	0.4
VOYAGE.heure='10H10'	0.4
TRANSPORT.type_trajet='direct'	0.2
VOYAGE.prix<1000	0.2 }

Profil P3 :

{VOYAGE.idT=TRANSPORT.idT	1.0
VOYAGE.lieu_arrivee='Barcelone'	1.0
VOYAGE.type_formule='week end'	1.0
VOYAGE.lieu_depart='Lyon'	0.8
VOYAGE.prix<800	0.8
TRANSPORT.moyen='train'	0.7
TRANSPORT.moyen='avion'	0.3
VOYAGE.lieu_depart='Paris'	0.2
TRANSPORT.comfort>2	0.1 }

Profil P4 :

{VOYAGE.idT=TRANSPORT.idT	1.0
VOYAGE.idH=HOTEL.idH	1.0
HOTEL.nbre_etoiles=2	0.9
VOYAGE.lieu_depart='Paris'	0.9
HOTEL.region='banlieu'	0.8
VOYAGE.nbre_jours=2	0.7
TRANSPORT.moyen='train'	0.6
TRANSPORT.moyen='car'	0.5
TRANSPORT.comfort=3	0.4
VOYAGE.type_formule='sejour'	0.3
VOYAGE.type_formule='thalasso'	0.3
VOYAGE.prix<800	0.2
HOTEL.restaurant='oui'	0.1 }

Définitions des sources :

```
S1:HotelsDuMonde(idH, nbre_etoiles, nom, region, ville, restaurant) :-
    HOTEL(idH, nbre_etoiles, nom, region, ville, restaurant).

S2:TransportAerien(idT, lieu_depart, lieu_arrivee, date_depart, heure, moyen, type_trajet,
   comfort) :-
   TRANSPORT(idT, 'avion', type_trajet, comfort),
   VOYAGE(idV,prix,lieu_depart, lieu_arrivee, nbre_jours, date_depar, heure, type_sejour,
      type_formule, idT, idH).

S3:sncf(idT,lieu_depart,lieu_arrivee, date_depart, heure, moyen, type_trajet, comfort) :-
   TRANSPORT(idT, 'train', type_trajet, comfort),
   VOYAGE(idV, prix, lieu_depart, lieu_arrivee, nbre_jours, date_depart, heure,
      type_sejour, type_formule, idT, idH).

S4:VoyagerPartout(idT, lieu_depart, lieu_arrivee, date_depart, heure, moyen, type_trajet,
   comfort) :-
   TRANSPORT(idT, moyen, type_trajet, comfort),
   VOYAGE(idV, prix, lieu_depart, lieu_arrivee, nbre_jours, date_depart, heure,
      type_sejour, type_formule, idT, idH).

S5:PromoVacances(idV, prix, lieu_depart, lieu_arrivee, nbre_jours, date_depart, heure,
   type_sejour, type_formule, moyen, nom, nbre_etoiles, restaurant) :-
   TRANSPORT(idT, moyen, type_trajet, comfort),
   VOYAGE(idV, prix, 'Paris', lieu_arrivee, nbre_jours, date_depart, heure, type_sejour,
      type_formule, idT, idH),
   HOTEL(idH, nbre_etoiles, nom, region, lieu_arrivee, restaurant).

S6:LyonVacances(idV, prix, lieu_depart, lieu_arrivee, nbre_jours, date_depart, heure,
   type_sejour, type_formule, moyen, nom,nbre_etoiles,restaurant) :-
   TRANSPORT(idT, moyen, type_trajet, comfort),
   VOYAGE(idV, prix,'Lyon', lieu_arrivee, nbre_jours, date_depart, heure, type_sejour,
      type_formule, idT,idH),
   HOTEL(idH, nbre_etoiles, nom, region, lieu_arrivee, restaurant).
```

Echantillon de requêtes initiales pour le test :

```
Q1 : SELECT idV, prix, V.lieu_depart, V.lieu_arrivee, T.moyen, T.comfort
     FROM VOYAGE V, TRANSPORT T
     WHERE V.idT = T.idT and V.lieu_arrivee='Madrid' and V.nbre_jours=4;

Q2 : SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, V.date_depart
     FROM VOYAGE V;

Q3 : SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, T.moyen, H.nom, T.comfort,
      H.region
     FROM VOYAGE V, TRANSPORT T, HOTEL H
     WHERE V.idT = T.idT and V.nom=H.nom and V.lieu_arrivee='Venise' and
      V.nbre_jours<13;

Q4 : SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, T.moyen, T.type_trajet
     FROM VOYAGE V, TRANSPORT T
     WHERE V.idT=T.idT and V.type_sejour='pension complete' and T.moyen ='train';

Q5 : SELECT idV, prix, V.lieu_depart, V.lieu_arrivee, V.date_depart, V.type_sejour
     FROM VOYAGE V
     WHERE V.lieu_arrivee='Rome' and V.prix<700;

Q6 : SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, T.moyen, T.type_trajet
     FROM VOYAGE V, TRANSPORT T
     WHERE V.idT = T.idT and V.lieu_depart='Paris' and V.lieu_arrivee='Madrid' and
      V.type_sejour='hotel et trajet';

Q7 : SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, V.date_depart
     FROM VOYAGE V
     WHERE V.lieu_depart='Paris' and V.lieu_arrivee='Venise';

Q8 : SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, V.date_depart, H.region
     FROM VOYAGE V, HOTEL H
     WHERE V.idH = H.idH and V.lieu_depart='Lyon' and H.region='centre ville';

Q9:  SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, V.date_depart, H.region
     FROM VOYAGE V, HOTEL H
     WHERE V.idH = H.idH and V.type_formule='week end';

Q10: SELECT idV, V.prix, V.lieu_depart, V.lieu_arrivee, V.date_depart, H.region
     FROM VOYAGE V, TRANSPORT T, HOTEL H
     WHERE V.idT = T.idT and V.idH = H.idH and V.lieu_depart='Paris' and
      T.type_trajet = 'direct' and T.comfort>2 and V.lieu_arrivee='Barcelone' and
      T.moyen='car';
```