Une Approche Ontologique d'Intégration de Sources de Données dans un Environnement de Pair à Pair

Amina Azzaz[1], Mimoun Malki[1], Ladjel Bellatreche[2], Youcef Benmimoun[3]

[1] EEDIS, Université Djillali Liabès, Sidi Bel Abess, Algérie
(azzaz.ami, mymalki)@gmail.com

[2] LISI/ENSMA - Université de Poitiers, France
bellatre@ensma.fr

[2] LSTE, Université Mustapha Stambouli, Mascara, Algérie
benyou@yahoo.fr

Résumé. Les systèmes pair à pair (P2P) sont des systèmes à grande échelle, auto-organisés et répartis. Ils permettent la gestion des ressources de manière totalement décentralisée. Cependant, l'intégration sémantique des données structurées, hétérogènes et distribuées à travers ces systèmes s'avère un problème complexe. L'objectif de ce travail consiste à proposer une approche dirigée par la sélection pour la reformulation des requêtes dans les systèmes d'intégration P2P, en introduisant, d'une part, l'information sur les mappings les plus pertinents et d'autre part, l'information provenant des requêtes passées. Une particularité de cette approche est qu'elle réalise un bon compromis entre l'efficacité et la qualité de la réponse.

Mots-clés: Intégration des données, ontologie, Pair à pair, correspondance sémantique, Reformulation de la requête.

1 Introduction

L'environnement pair à pair se présente actuellement comme une solution viable pour permettre le passage à l'échelle de l'Internet. Chaque pair se comporte à la fois comme client et serveur, et fournit une partie de l'ensemble des informations de l'environnement distribué sans s'appuyer sur une administration centrale. En effet, le paradigme p2p est mis en œuvre dans de nombreux domaines d'applications couvrant notamment le calcul distribué tel que *Seti@home* [1], les systèmes de stockage persistant à grande échelle comme *OceanStore* [2] et les systèmes de partage de fichier tel que *Napster* [3] ou *Kaaza* [4]. Les réseaux Pair-à-Pair, ont permis de mettre en place des moyens simples de partage de données tout en se limitant, cependant, à la recherche par mots clés.

Par ailleurs, l'intégration de sources de données de structures et de sémantique complexes, est devenue un domaine de recherche très important à cause de l'explosion du nombre de sources et leurs hétérogénéités[1]. L'objectif est de donner l'impression d'utiliser un système homogène et centralisé. Deux approches principales pour la conception des systèmes d'intégration ont été définies en se fondant sur la localisation des données gérées par le système : lorsque les données des sources sont stockées dans le système d'intégration on parle *d'approche matérialisée* ou *entrepôt de données* [5], à l'inverse, lorsque les données intégrées ne le sont pas, on parle *d'approche virtuelle* ou *système de médiation* [6]. Bien que ces systèmes soient efficaces pour des applications comportant peu de sources de données, ils sont peu adaptés au nouveau contexte d'intégration soulevé par le web car ils reposent sur un schéma global unique.

Récemment, les systèmes Pair à Pair de gestion de données (PDMS[2]) ont vu le jour. Ils combinent la technologie Pair à Pair et celle des bases de données distribuées et s'appuient sur une description sémantique des sources de données [12], on peut citer l'approche *PIAZZA* [7], *SomeWhere* [8] et *PeerDB* [9]. Bien que ce couplage entre les techniques d'intégration de données et les systèmes p2p est fructueux, il est indispensable de lever certains défis: tels que ceux qui sont dus à l'hétérogénéité[3] et à la nature décentralisée et dynamique du p2p. La problématique de ce type de système peut être décrite comme suit: étant donnés un ensemble **P** de pairs liés physiquement, contenant des sources de données $\mathbf{S_i}$ autonomes et hétérogènes. On espère pouvoir interroger les données de ces pairs comme si elles constituaient une seule source en se basant uniquement sur un réseau **M** des correspondances sémantiques[4]. Un mapping sémantique définit l'équivalence conceptuelle entre des attributs définis dans deux schémas[5] de pairs différents.

Il s'agit de systèmes de médiation sans schéma global (Cf. la Figure.1) où chaque pair dispose de son propre schéma local et de schéma de correspondance vers d'autres pairs. Il n'y a pas ici, à proprement parler, de processus de routage de requêtes, puisque les pairs vers lesquels propage une requête sont définis par les schémas de correspondance. Par contre, il faut ici : (i) *de nouvelles méthodes de découverte automatique des correspondances* (alignement des ontologies), parmi les travaux récents dans ce sens [13, 14], et (ii) *des algorithmes efficaces de réécriture et d'optimisation des requêtes en fonction de cet ensemble de mappings*. Ce dernier challenge est indispensable pour rendre un PDMS fonctionnel, autrement dit, pour permettre le traitement des requêtes à large échelle tenant compte l'efficacité et la qualité de la réponse.

---

[1] Plusieurs types de conflits dus à l'hétérogénéité peuvent être considérés dans l'établissement des correspondances entre les sources lors de l'intégration des données. De nombreuses taxonomies de conflits ont été proposées [18], on peut simplement les considéré en deux types : syntaxique et sémantique.

[2] Communément appelé PDMS : Peer Data Management System.

[3] Le rôle des ontologies est central dans le développement des systèmes intégrant des sources hétérogènes.

[4] Dans cette article nous utilisons aussi le terme mappings pour désigner les correspondances sémantiques.

[5] Schémas relationnels, DTDs, Ontologies... Dans la suite de cet article, nous utilisons les termes ontologie et schéma de manière interchangeable.

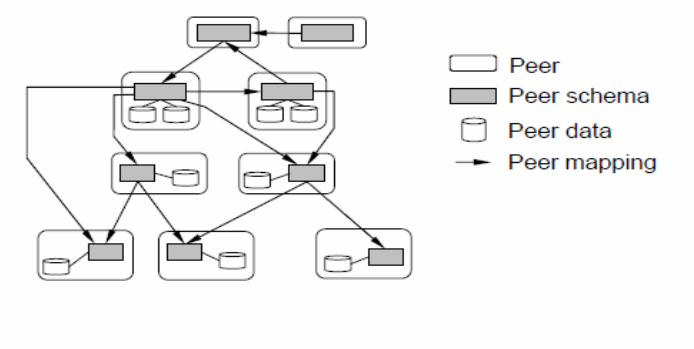

Fig. 1. Architecture d'un système d'intégration p2p.

Notre contribution est consacrée au deuxième problème soulevé de par sa complexité, nous proposerons une approche dirigée par un algorithme de sélection afin d'éviter la redondance des chemins empruntés par une requête et de ne la faire propager que vers les pairs pertinents. En supposant que les pairs partagent la même ontologie (conceptualisation du domaine pour la perspective de réconciliation [17]), chaque pair est décrit par une ontologie locale (source de données à base ontologique [16]) et maintenir une expertise (un module supplémentaire qui représente sa vision au monde) et également faire intervenir leur comportement passé (historique des requêtes).

Dans le reste du document, nous discuterons des travaux connexes afin de positionner notre contribution (Section 2), présenterons notre approche et les algorithmes associés (Section 3). Aussi, nous définirons les critères d'évaluation et présenterons notre simulateur (section 4). En conclusion, nous étalerons des perspectives et orientations pour les travaux futurs (Section 5).

2 Travaux connexes

De nombreux travaux basés sur des correspondances sémantiques entre les pairs ont été développés. Différents procédés sont utilisés pour générer ces correspondances de manière plus ou moins automatique. Nous passons en revue quelques travaux typiques suivis d'un tableau comparatif.

***Piazza* [7]** Permet aussi bien l'échange de données relationnelles, XML que RDF. Il est basé sur une architecture Pair à Pair pure. En présence de différents schémas et de différentes représentations, les pairs intéressés par l'échange de données définissent des correspondances sémantiques entre eux, deux à deux ou entre petits groupes de pairs. Chaque pair exprime ses requêtes sur son propre schéma. Les requêtes sont dans ce cas évaluées globalement sur un réseau de pairs sémantiquement liés par les correspondances. Piazza combine et généralise les formalismes LAV (Local As View) et GAV (Global-As-View) proposés dans la médiation de schémas dans les systèmes

d'intégration de données et les étend aux documents XML. Le langage d'expression des correspondances pour les données relationnelles est PPL (Peer Programming Language) tandis que celui utilisé pour les documents XML est basé sur XQuery. La réécriture des requêtes est basée sur un pattern matching entre les expressions XQuery et les correspondances sémantiques et elle est faite de manière centralisée. L'approche Piazza présente cependant des insuffisances liées à la difficulté de décrire les correspondances, de les construire mais aussi à la maintenance de ces dernières. A noter que les reformulations des requêtes sont faite par un nœud central.

***SomeWhere* [8]** Dans SomeWhere, aucun utilisateur n'impose aux autres sa propre ontologie, car le système permet de créer des mises en correspondance entre différentes ontologies. Un pair se connectant un réseau construit les mappings entre sa propre ontologie et les ontologies des pairs servant de point d'entrée dans le réseau. Pour traiter une requête, l'utilisateur doit choisir le pair par lequel sa requête sera initiée dans le réseau. Le routage des requêtes est guidé vers les pairs dont les mappings sont pertinents. Cette pertinence est établie selon un algorithme distribué de raisonnement en logique des propositions (FOL).

***GLUE* [11]** Le système GLUE utilise une approche par machine learning pour classifier les concepts d'une ontologie afin de les mettre de manière semi automatique en correspondance avec les concepts définis dans les ontologies distantes. Cependant, bien que tout repose sur la phase d'apprentissage, cette approche suppose qu'un nombre important d'utilisateurs collabore pour définir les mappings sémantiques entre les ontologies.

***PeerDB* [9]** Permet le partage de données relationnelles distribuées sans partage de schéma. Il combine les propriétés des systèmes multi-agents avec celles des systèmes Pair-à-Pair. Chaque pair fournit une base de données relationnelle décrite grâce à des méta-données (mots-clés). La reformulation des requêtes est faite par des agents grâce à une mise en correspondance des méta-données associées aux schémas. L'approche PeerDB présente la faiblesse d'autoriser des correspondances entre mots-clés pouvant aboutir à de fausses reformulations.

Table 1. Comparaison entre quelque PDMS

	Piazza	**SomeWhere**	**Glue**	**PeerDB**
Découverte de mappings	Statique	Statique	Semi - automatique	dynamique
Principe	PPL	FOL	Machine learning	Agent mobile
Reformulation de requêtes	faite par un noeud central	totalement décentralisée	Collaboration d'un nombre important d'utilisateur	basé sur annotation des mots clés
Passage à l'échelle	√ (Jusqu'à 80 pairs)	√√ (Mille pairs)	√√	√√

3 Approche de reformulation efficace dirigée par la sélection

Considérant le processus du traitement des requêtes dans les systèmes d'intégration p2p suivant :

1. Chaque utilisateur interroge le réseau via un pair de son choix ;
2. Les requêtes complexes sont décomposées au niveau du pair;
3. Transmission des requêtes atomiques aux autres pairs concernés ;
4. Chaque pair traite la requête atomique reçue ;
5. Les réponses sont exprimées en terme de langage de mappings utilisé ;
6. Les réponses d'une requête complexe doivent être recombinées.

Ce processus est itératif. L'étape 3 est fondamentale, raison pour laquelle l'effort de notre approche s'y focalise. Ainsi, on peut formaliser le problème de reformulation des requêtes comme suit : *soit* $\boldsymbol{N\ (P_i, O_i, M)}$ *un système d'intégration P2P (avec* $\boldsymbol{P_{i=1...n}}$ *collection de pairs,* $\boldsymbol{O_i}$ *ontologies locales,* $\boldsymbol{M}$ *ensemble de mappings),* $\boldsymbol{Q}$ *une requête posé à un pair (en termes de* $\boldsymbol{O_i}$*). L'objectif est de Calculer les reformulations de Q en fonction de M (les réécritures maximales* $\boldsymbol{Q_e}$ *de* $\boldsymbol{Q}$*).*

Une approche classique pour atteindre cet objectif est d'effectuer toutes les reformulations possibles c'est-à-dire propager Q du nœud initiateur à ses voisins (avec lesquels il dispose de correspondances sémantiques) et ceci récursivement jusqu'à ce qu'une borne de terminaison soit atteinte (généralement un TTL)[6]. C'est le principe de Gossiping utilisé dans les systèmes de partage de fichiers (comme Napster ou Gnutella). Mais il est clair que ce principe est inadapté dans le cas de sources de données de structures et de sémantiques complexes vis-à-vis le coût des reformulations et la complexité élevée de l'algorithme (cas de cycle). Une autre approche basée sur un stockage centralisé de tous les schémas et les mappings offrant une vision globale du réseau et permettant de guider le processus de reformulation a prouvé son avantage dans le système Piazza par la mis en place d'un mécanisme efficace pour trouver toutes les réponses certaines (Rule-Goal Tree Expansion). Contrairement au Gossiping, la complexité est polynomiale. Néanmoins, le besoin d'un stockage centralisé est un obstacle pour le passage à l'échelle de cette approche. Dans SomeWhere, un algorithme distribué basés sur un langage de description d'ontologie avec un encodage logique (FOL) est implémenté, prouvant sa robustesse, complétude et terminaison, c'est un algorithme anytime, malgré cela, il a une faiblesse due à l'échange d'un nombre important de messages. Nous proposons une approche de reformulation efficace basée sur une sélection à priori des mappings pertinents en minimisant le nombre de messages échangés entre les pairs et en évitant les cycles.

3.1 Architecture et Principe de fonctionnement

L'architecture de notre approche est illustrée par l'exemple de la figure.2 ; pour simplifier la vue, notre système est composé de quatre pairs $\boldsymbol{P= \{P_1, P_2, P_3, P_4\}}$ partageant une seule relation respectivement:

[6]Time To Live de 4 à 7 selon le niveau de complétude de résultat que l'on souhaite.

Publication *(id, Titre, Nom.Auteur, Année)*
Book *(code, Title, Author, Year, Prise)*
Livre *(Idl, Titre, Auteur, Maison.edition, Prix)*
Chercheur *(Id, Nom, Mail, Grade, Projet)*

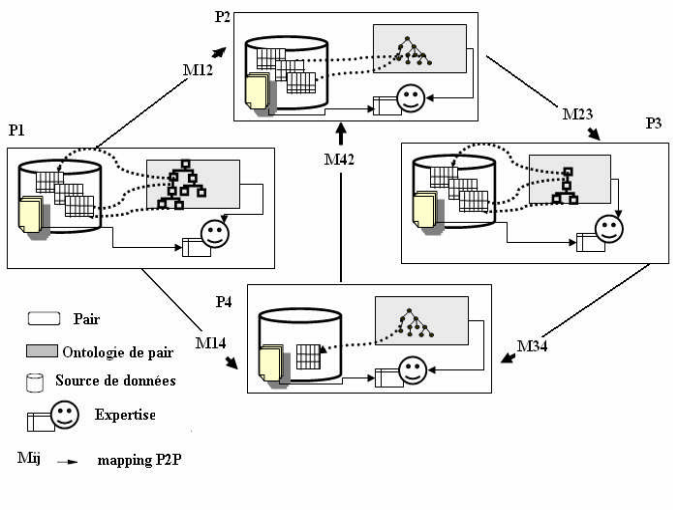

Fig. 2. Architecture de l'Approche Proposée : Exemple.

Nous considérons que les pairs n'échangent que les sources de données à base ontologique[7]. En effet chaque pair pouvant stocker à la fois le contenu usuel d'une base de données et l'ontologie qui décrit la sémantique de ces données. Le principe de notre approche consiste à l'établissement d'un plan de reformulation à priori en intégrant deux idées : d'une part, avoir une information sur les mappings les plus pertinents (pour cela un degré de similarité[8] est associé à chaque mappings) et d'autre part, utiliser l'information provenant des requêtes passées (pour cela un historique des requêtes est envisagé). Ces deux types d'informations sont exploités par un module supplémentaire au niveau de chaque pair que nous appelons ici **une Expertise**, notant qu'une conceptualisation partagée (ontologie de domaine) est toujours recommandée (ontologie pour la perspective de réconciliation [17]).

3.2 Algorithmes de l'approche proposée

Le fonctionnement de notre approche est donné par l'algorithme de *reformulation-efficace* et l'algorithme récursif de *sélection (P, Q)*.

---

[7] Le concept Base de données à base ontologique est développé dans le cadre du projet OntoDB au sein du LISI.

[8] Le domaine de l'identification de la similarité a été considéré comme un sujet de recherche fortement recommandé dans les domaines du Web sémantique, de l'intelligence artificielle et de la littérature linguistique. Dans [18], un état de l'art sur les mesures de similarité ainsi qu'une proposition d'une nouvelle mesure (pour laquelle on a opté) sont présentés.

A- Algorithme de reformulation-efficace ;

Entrées : N <P, O, M> (un système d'intégration P2P); Q (une requête posé au pair P_i) ; E (Expertise) ;

Sorties : l'évaluation de Q globalement suivant un Path de reformulation ;

Début

1. *Prétraitement de la requête ; /* Décomposition des requêtes complexes et translater en terme de l'ontologie global */*
2. *Consulter l'expertise du pair ;*
3. ***Si** (existe-déjà (Q) = vrai) **Alors** Path (Q) ← E.Path;*
4. ***Sinon** Path ← Sélection (Pi, Q) ;*
5. ***Fin Si***
6. *Traitement de la requête ; / * suivant les mappings du Path choisi, reformuler en Q', Q'',… et répondre à chaque Requête localement */*
7. **Si** *(qualité(R) = vrai) Alors E.Path ← Path ; /* Mise à jour de l'expertise */*
8. ***Fin Si***

Fin

Comme il est décrit dans l'algorithme de reformulation efficace, notre approche fait appel à un algorithme de sélection qui assure le choix d'un plan de reformulation au préalable (Path). Ceci fait que le pair émetteur de la requête forme une vision plus au mois globale à propos du déroulement de sa requête.

B- Algorithme de Sélection (Pi, Q);

Entrées : N <P, S, M>; Q; E;

Sortie : Path ;

Début

1. *Q est posé à un pair Pi ;*
2. ***Pour tous** les mappings Mi,j de Pi*
3. Calculer *eff (Mi,j) ; /*efficacité d'un mappings est donné par un degré De similarité calculé entre [0,1] */*
4. ***Si** (eff (Mi,j) > seuil et Max) **Alors***
5. *Marquer (Mi, j) ;*
6. ***Fin Si***
7. *Sélection (Pj, Q') ; / * reformuler la requête et passer à une nouvelle itération */*
8. ***Fin Pour***
9. *Path←les mapping marqués;*

Fin

3.3 Exemple

Soit la requête ***Q : donner les titres des publications de l'année 2009***, requête posée au pair P_1 (exemple précédent). Le déroulement de notre algorithme de reformulation est comme suit: Q est posé en terme de O_1 (l'ontologie locale de P_1). Au niveau de l'expertise de P_1, une reformulation de Q en terme de O_G est effectuée en détectant le sub-concept (publication), supposant que Path $_{\text{requêtes passées}}$ ← vide.

Toujours au niveau de E et en collaboration avec les expertises des autres pairs, on calcule les degrés de similarité par rapport au concept en question (publication) à savoir : eff (Mi, j). On abouti ainsi aux résultats suivants (Table *2.).*

Table 2. eff (M_{ij})/Publication

eff (M_{12})	*eff (M_{14})*	*eff (M_{23})*	*eff (M_{34})*	*eff (M_{42})*
0,8	*0,4*	*0,6*	*0,3*	*0,3*

Nous choisissons parmi les mappings supérieur à un seuil, celle dont la valeur d'efficacité est la plus élevée. En suite nous passons à une deuxième itération (Q←Q' reformuler Q en fonction de M choisi dans l'étape 4 et $P_1 \leftarrow P_2$). Le path résultant (formé des mappings sélectionnés) M_{12}, M_{34} avec un seuil donné (0,5), donc la réponse à la requête en question est fournie par P_1, P_2 et P_3. Cet exemple montre l'efficacité de notre algorithme en évitant le cycle (M_{42}) et le pair non pertinent (P_4).

4 Implémentation

La démonstration de l'approche se fait par le développement d'un simulateur de système d'intégration p2p en se basant sur PeerSim [15]: un outil[9] open source écrit en Java et qui présente l'avantage d'être spécialisé pour l'étude des systèmes p2p. De plus, il dispose d'une architecture ouverte et modulaire qui permet de l'adapter et le spécialiser. Notre simulateur doit avoir en entrée un jeu de données qui décrivent les pairs du système, la liste des sources de données ainsi que les requêtes qui seront lancées dans le réseau. Il effectue ensuite des simulations sur la base de ces données pour produire en sortie la liste des requêtes envoyées avec leurs chemins de propagation, la liste des réponses à chaque requête, les sources pertinentes, etc. Pour cela, notre Simulateur réalise les fonctions suivantes :

- Construire le réseau (nombre de pairs et leurs caractéristiques).
- Initialiser le réseau (sources de données, ensemble des mappings, ...).
- Maintenir l'expertise (liste de voisinages actifs, paths des requêtes passées).
- Implémenter les approches classiques (Gossiping, Nœud Central) et notre nouvelle approche (approche à base de sélection) pour effectuer une comparaison suivant des critères d'évaluation.

La figure suivante représente le diagramme de séquences du scénario Simulation. L'utilisateur lance la simulation à travers l'Interface Utilisateur. Ensuite la classe configuration s'occupe du chargement du contenu du fichier de configuration. Le simulateur initialise le réseau, et charge les différentes composantes du système, à

[9] C'est un projet Java libre (licence GPL) de l'université de Bologne.

savoir : les observers, dynamics et les protocoles avant d'entrer dans les boucles de la simulation.

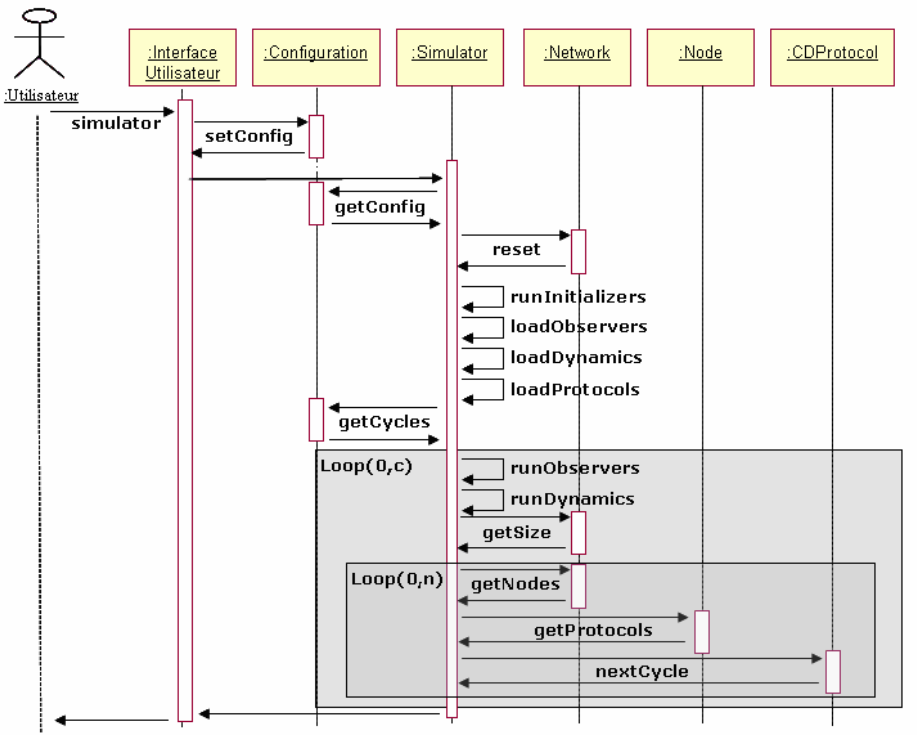

Fig. 3. Diagramme de séquence du scénario Simulation.

4.1 Critères d'évaluation

Afin d'évaluer les performances de notre approche, nous considérons les critères suivants :

- ***L'efficacité du traitement :*** signifie la terminaison du processus de la reformulation dans un temps tolérable. Nous mesurons l'efficacité en fonction du temps de réponse moyen.
- ***La qualité de la réponse (IQ) :*** la réponse rendue doit être correcte (soundness) et doit inclure toutes les réponses qui existent (completeness). Nous jugeons IQ par rapport à une réponse modèle (pattern).

5 Conclusion et perspectives

L'intégration de sources de données est devenue un domaine de recherche très important du faite de l'explosion du nombre de sources et leurs hétérogénéités. Cependant, le passage à l'échelle et à un comportement dynamique est fonctionnellement problématique pour les systèmes d'intégration centralisés. D'où l'apparition d'une nouvelle classe d'outils pour l'intégration de données tirant profit des principes de fonctionnement des systèmes P2P. Beaucoup de contraintes s'imposent sur l'intégration de données dans les réseaux P2P. Il s'agit de systèmes de médiation sans schéma global où chaque pair dispose de son propre schéma local (ontologie locale) et de schéma de correspondance vers d'autres pairs. Il faut ici des algorithmes efficaces de réécriture et d'optimisation. Ce challenge est indispensable pour rendre un PDMS fonctionnel.

Dans cet article, nous avons présenté une approche de reformulation efficace de requête dans les systèmes d'intégration p2p. L'originalité de cette approche est quelle passe par une étape de sélection à priori, ce qui permettra de garantir l'autonomie des pairs mais aussi, de minimiser le nombre de messages échangés, aussi que les reformulations non pertinentes. Cette approche évitera aussi la redondance, tout en assurant une qualité supérieure de la réponse (soundness, completeness). Ce travail présente l'avantage de tirer partie de l'historique des requêtes et vise l'élaboration d'un plan d'exécution à priori qui offre au pair une vision globale sur le déroulement de sa requête. La démonstration de l'approche se fait par le développement d'un simulateur de système d'intégration p2p.

Une amélioration peut être atteinte en enrichissant l'expertise (de nouvelles mesures de sélection, des algorithmes d'apprentissage de type neuronal, d'ajouter la connaissance des centres d'intérêt des utilisateurs), et/ou en introduisant la notion des agents (agent intelligent). Le rajout d'un mécanisme de retour dans l'algorithme s'avère indispensable afin de pouvoir établir plusieurs plans d'exécution (Paths) candidats permettant la tolérance aux pannes. Cependant, notre travail ne maintient pas l'étape de la reformulation proprement dite qui reste la tâche la plus difficile mais également dont l'impacte est prometteuse, donc une perspective à long terme est d'étudier cette étape en bas niveau (encodage en logique descriptive).

Références

[1] SETI@home. Home Page, «*http://setiathome.ssl.berkely.edu/*».
[2] KubiatowiczJ. , Bindel D., Chen Y., Czerwinski S., Eaton P., Geels D., Gummadi R., Rhea S.,Weatherspoon H., Wells C., Zhao B. «OceanStore: architecture for global-scale Perststent storage ». ACM SIG ARCH (2000).
[3] NAPSTER, *« www.napster.com »*.
[4] KAZAA, *« www.kazaa.com »*.
[5] J. Widom. *«Research problems in data warehousing»*. In Conference on Information and Knowledge Management, (1995).
[6] Wiederhold G. Mediators in the architecture of future information systems. (1992).

[7] Halevy A., Ives Z. G., Mork P., Tatarinov I., *«Piazza: Data Management Infrastructure for Semantic Web Applications»*, Proceedings of the twelfth international conference on World Wide Web Budapest, Pages 556-567, (2003).
[8] P. Adjiman, P. Chatalic, F. Goasdoué, M-C. Rousset, and L. Simon. «Somewhere in the semantic web». International workshop on principles and practice of semantic web reasoning, (2005).
[9] W. S. Ng, B. C. Ooi, K-L. Tan, and A. Zhou. «Peerdb: A p2p-based system for distributed data sharing», (2003).
[10] Parent, C. et Spaccapietra, S. *«Intégration de bases de données : Panorama des problèmes et des approches»*. Ing. Des Syst. D'Info., 4(3). (1996).
[11] A. Doan, P. Domingos, and A. Halevy. «Learning to match the schemas of data sources: A multistrategy approach. » Mach. Learn., (2003).
[12] M. Lenzerini. *«Principles of p2p data integration»* In Zohra Bellahsene and Peter McBrien, editors, DIWeb, (2004).
[13] François-Elie Calvier, Chantal Reynaud. Découverte de correspondances entre ontologies distribuées LRI, Univ. Paris-Sud & INRIA Futurs, Orsay Cedex, France, (2007).
[14] Lionel Médini, C. Ferreira da Silva, Nicolas Lumineau, Patrick Hoffmann, Parisa Ghodous. *«Découverte de correspondances sémantiques par inférences dans un environnement P2P»*. DECOR Passage à l'échelle des techniques de découverte de correspondances, Namur, Belgique janvier (2007).
[15] PeerSim, *« http://sourceforge.net/projects/peersim»*.
[16] Ladjel Bellatreche, Guy Pierra, Dung Nguyen Xuan, Dehainsala Hondjack. *«Intégration de sources de données autonomes par articulation à priori d'ontologies»*. In proceeding of the XXIIème Congrès INFORSID, Biarritz, France, 25-28 May (2004).
[17] Fatiha SAIS. « Intégration Sémantique de Données guidée par une Ontologie ». Thèse de Doctorat de l'Université Paris-Sud. Décembre (2007).
[18] Thabet Slimani, Boutheina Ben Yaghlane Khaled Mellouli. *«Une extension de mesure de similarité entre les concepts d'une ontologie»*. SETIT2007 4rth International Conference: Sciences of Electronic, Technologies of Information and Telecommunications–TUNISIA (2007).