SenPeer : un système pair-à-pair de médiation de données

David C. Faye, Gilles Nachouki, Patrick Valduriez

SenPeer

Un système Pair-à-Pair de médiation de données

David Faye[*-**] — Gilles Nachouki[**] — Patrick Valduriez[***]

[*] Laboratoire d'Analyse Numérique et d'Informatique (LANI)
Université Gaston Berger de Saint-Louis
BP 234 Saint-Louis (SENEGAL)
David.Faye@univ-nantes.fr

[**] Laboratoire d'Informatique de Nantes Atlantique (LINA)
Université de Nantes
2 rue de la Houssinière , BP-92208, 44322 Nantes cedex 03 (FRANCE)
Gilles.Nachouki@univ-nantes.fr

[***] Equipe ATLAS (INRIA et LINA)
Patrick.Valduriez@inria.fr

RÉSUMÉ. Dans cet article nous présentons SenPeer, un nouveau système Pair-à-Pair de gestion de données distribuées permettant le partage décentralisé et flexible de données relatives à la mise en valeur du fleuve sénégal. SenPeer est un système de type Super-Pair reposant sur une organisation des pairs en domaines sémantiques et dans lequel les pairs peuvent publier des bases de données relationnelles ou objets ou des documents XML. Chaque pair exporte ses données dans un formalisme pivot commun qui a une structure de graphe enrichi sémantiquement avec des mots-clés destinés à guider la découverte des correspondances entre les éléments des schémas. Ces correspondances sont découvertes grâce à un ensemble de mesures de similarités floues. De plus elles permettent l'établissement d'un réseau sémantique à coté de la topologie physique, pouvant servir de support à un routage intelligent des requêtes.

ABSTRACT. In this article we present SenPeer, a new Peer-to-Peer data management system allowing data sharing among experts working on the development of the senegal river in a decentralized and flexible fashion. SenPeer has a Super-peer network topology based on an organization of peers in semantic domains and in which peers can contribute XML documents, relational or object databases. Each peer exports its data in a common formalism which has a graph struture semantically enriched with a set of keywords in order to guide mappings discovery. Mappings discovery relies on a set of fuzzy similarity measures. Moreover they allow the establishment of a semantic topology that is independent of the underlying network toplogy which is the basis for intelligent query routing.

MOTS-CLÉS : Médiation de données, similarité sémantique, Systèmes Pair-à-Pair

KEYWORDS : Data mediation, semantic similarity, Peer-to-Peer systems

1. Introduction

La société de l'information demande un accès complet et efficace à l'information disponible, information qui est souvent hétérogène et distribuée. Les solutions techniques proposées dans un premier temps, le Web et 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. D'autre part, l'intégration de données et les Entrepôts de Données ont été les deux principales approches proposées pour la réconciliation de données. Cependant ces deux technologies sont lourdes car imposant un schéma central médiateur qui en plus d'être un frein à l'évolution de schéma, complique aussi le partage de données.

Les systèmes Pair-à-Pair classiques tels que Napster[24] ou Kaaza[18] se sont illustrés par une description des nœuds par clés et une localisation des données par un routage de ces mêmes clés dans le réseau. Récemment, les systèmes Pair-à-Pair de gestion de données communément appelés PDMS (*Peer Data Management System*) [12] 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 pour aider dans la formulation et le routage des requêtes à travers le réseau mais aussi l'intégration des résultats. Cependant la plupart des PDMS ne permettent pas le partage de données décrites par des modèles de données différents[29][26][11][25][16]. Dans certains cas, leur mise en place est lourde car nécessitant un certain nombre de correspondances sémantiques[9][3], ou parfois même faisant appel à l'intervention humaine[2][26], ce qui n'est pas envisageable dans un environnement distribué mettant en jeu un nombre important de pairs.

Dans cet article, nous présentons SenPeer, un système Pair-à-Pair de partage de données utiles en matière de gestion d'aménagement et de valorisation de la vallée du fleuve Sénégal. SenPeer présente une architecture hybride de type Super-Pair s'appuyant sur le regroupement des pairs en domaines sémantiques. Chaque pair publie des données décrites avec les modèles de données relationnel, objet ou XML et dispose de son propre langage d'interrogation. Dans le but d'une médiation flexible, chaque pair exporte une couche de ses données dans un modèle pivot qui a une structure de graphe enrichi sémantiquement avec des mots-clés issus des schémas et destinés à guider la découverte des correspondances sémantiques. Ces correspondances sémantiques permettent ainsi l'établissement d'un réseau sémantique à coté du réseau physique existant, autorisant ainsi le routage intelligent des requêtes et leur réécriture sur d'autres schémas.

Le reste de l'article est organisé comme suit. Dans la partie suivante nous posons le problème en l'illustrant sur un exemple lié à la mise en valeur de la vallée du fleuve Sénégal. La partie 3 présente l'architecture du système SenPeer. La partie 4 décrit les processus de médiation de données avec les différentes mesures de similarités et les algorithmes liés. Dans la partie 5 nous passons en revue quelques travaux connexes et nous terminons finalement par une conclusion et quelques directions pour des travaux futurs.

2. Le problème de la médiation de données du fleuve sénégal

Le fleuve sénégal délimite une frontière naturelle entre le Mali, la Mauritanie et le Sénégal. Toute la région du fleuve constitue une zone agricole, d'élevage et de pêche partagée par ces trois pays frontaliers. La mise en valeur de la vallée du fleuve fait intervenir depuis des années des experts de divers organismes (OMVS - Organisation pour la Mise en valeur de la Vallée du Fleuve Sénégal, ISRA - Institut Sénégalais de Recherche Agronomique, SAED - Société d'Aménagement et d'Exploitation des terres du Delta, OMS - Organisation Mondiale de la Santé, etc.) de différents domaines de compétence (hydraulique, activités agricoles, recherche agronomique, santé, etc.) mais aussi localisés dans ces différents pays. Tous ces experts mènent des travaux qui aboutissent généralement à la production et à l'exploitation de gros volumes de données. La gestion et l'exploitation des données sont loin d'être satisfaisantes à cause de leur distribution, hétérogénéité, volume et appartenance[6]. En effet le système actuel de partage de données est basé sur un serveur central, localisé à Dakar, coordonnant toutes les sources de données de la sous-région et dont la défaillance entraine l'effondrement de tout le réseau de partage de données. De plus il ne permet que l'échange de bases de données relationnelles.

La politique de mise en valeur passe par une exploitation rationnelle des données existantes. Il est donc capital de fournir aux producteurs et consommateurs de données de puissants moyens d'intégration de gestion, de diffusion et de recherche des données existantes. Cela va ainsi permettre aux décideurs de tirer le maximum d'informations pertinentes à partir de ces données. Nous pensons qu'une plate-forme logicielle de type

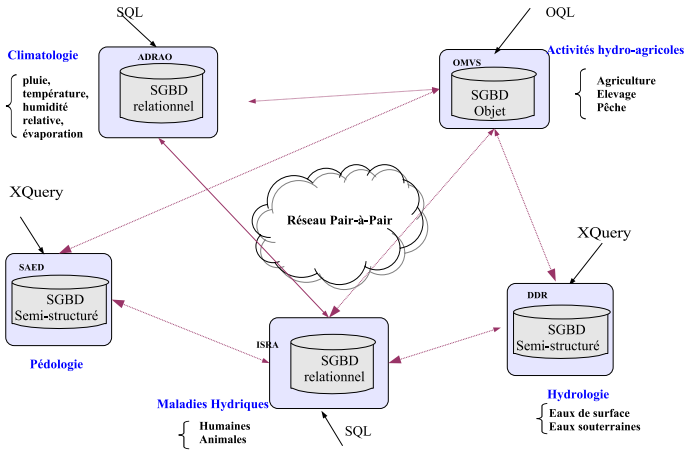

Figure 1. *Partage de données du fleuve en présence de plusieurs modèle de données.*

PDMS semble bien adaptée à ce scénario. Dans un réseau de partage de données comme celui de la figure 1 chaque expert ou pair devra pouvoir, de façon autonome :

– fournir des données de nature diverse en vue de les partager avec les autres experts quelque soit son pays ou son domaine de compétence et les lier à celles existantes ;

– effectuer la recherche de données pertinentes sur celles proposées par les autres experts. Il peut choisir d'étendre ses données avec les nouvelles données trouvées.

Les données relatives à un domaine étant réparties entre différents acteurs, il y a un besoin de répondre à des requêtes complexes du type :

– (R1) *Quel est le taux de bilharziose quand le débit des eaux est de 640* m^3/s *?*

– (R2) *Quel est le rendement des cultures sous pluie quand la température moyenne saisonnière est de 30*$^\circ$ *?*

– (R3) *Quelles sont les caractéristiques des sols sur lesquels est cultivé le riz ?*

Ces requêtes nécessitent une médiation sémantique tenant compte des liens complexes pouvant exister entre les différentes données des experts. Par exemple, le traitement de la requête (R1) va impliquer les spécialistes des maladies hydriques humaines et ceux de l'hydraulique mais aussi tenir en compte le fait que les données d'un même domaine ont été mises en place par différents acteurs. Le problème est d'autant plus difficile que ces derniers sont autorisés à fournir des données basées sur des modèles de données différents (relationnel, objet ou XML). Dans ce cas, une requête sur le PDMS sera posée sur le schéma d'un pair spécifique avec son propre langage d'interrogation SQL, OQL, XQuery.

Dans ce contexte, il faut relever les défis suivants :

– **Médiation de données** : La maintenance d'un schéma global qui est un frein à l'extensibilité du réseau vu sa dynamicité. Les sources étant hétérogènes aussi, il faut trouver un moyen pour réconcilier les différentes représentations de façon dynamique.

– **Réécriture des requêtes** : Les langages d'interrogation n'étant pas les mêmes, il faut trouver un mécanisme permettant, étant donné une requête formulée avec un langage donné sur le schéma d'un pair, de reformuler cette dernière sur le schéma d'autres pairs avec le langage d'interrogation de ces pairs.

– **Passage à l'échelle** : Le nombre important de pairs affecte le temps de traitement des requêtes et l'accessibilité des sources. Il faut trouver dans ce cas des algorithmes de routage efficaces qui tiendront compte de la diversité des modèles de données.

– **Dynamicité du système** : Aussi bien la topologie du réseau que les données et schémas des pairs peuvent changer subitement. La topologie changeante a des incidences sur le routage des requêtes. La mise à jour subite des données et schémas affecte le routage des requêtes, mais aussi la médiation entre schémas.

Dans cette partie nous avons introduit un exemple nécessitant une plateforme logicielle de type PDMS. Nous avons aussi énuméré les défis à relever vu l'hétérogénéité des sources et leur volatilité.

3. Architecture du système SenPeer

3.1. Notre contexte

Nous nous plaçons dans le cadre des applications de partage de données pour lesquelles ces dernières peuvent être organisées par thèmes grâce à une taxonomie. Chaque pair est libre d'exporter des données de son choix à condition qu'elles soient décrites par un schéma. Les thèmes ne sont pas forcément disjoints et chacun correspond à un domaine particulier. Pour le cas du fleuve sénégal, les domaines identifiés et intéressants pour la mise en valeur de la vallée peuvent être représentés en partie dans la figure 2. Par exemple, les thèmes *hydrologie* et *maladies hydriques* sont non disjoints car les maladies hydriques sont liées au débit et au niveau de l'eau notamment dans les environs des barrages. Donc les experts de ces deux domaines doivent disposer des données liées à l'eau.

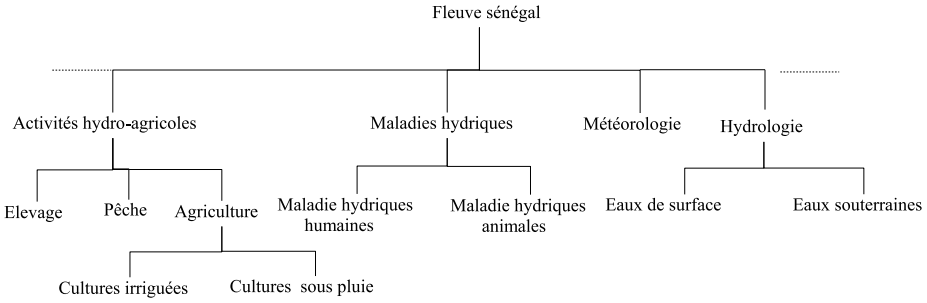

Figure 2. *Partie de la taxonomie organisant les données de la vallée du fleuve sénégal.*

3.2. Topologie du système SenPeer

Le système proposé repose sur une architecture de type Super-Pair s'appuyant sur un regroupement des pairs en domaines sémantiques (Figure 3). Cette architecture combine l'approche centralisée et celle non structurée prenant ainsi les avantages de la recherche centralisée et de l'autonomie, de la répartition des charges et de la robustesse pour une recherche distribuée. L' architecture Super-Pair permet de profiter de l'hétérogénéité des pairs en assignant plus de responsabilités aux pairs capables de les assumer. Par conséquent, certains pairs, appelés super-pairs, qui ont un pouvoir de calcul additionnel et une plus grande bande passante effectuent des tâches administratives. Ils sont chargés de la gestion d'un ensemble de pairs, permettant entre autre de diminuer les efforts de compilation de requêtes mais aussi d'empêcher la diffusion des requêtes dans le réseau.

Dans chaque domaine se trouve un super-pair auquel est rattaché un réseau sémantique expliquant les données du domaine. Ce réseau sémantique est purement conceptuel, les données nécessaires pour répondre aux requêtes étant stockées dans les pairs du domaine. Les pairs d'un même domaine sont directement rattachés au super-pair correspondant, mais peuvent décrire leurs données avec des vocabulaires différents. Dès lors, il s'avère nécessaire d'établir une médiation sémantique entre les pairs d'un domaine et leur super-pair, mais aussi entre les super-pairs puisque les domaines ne sont pas disjoints.

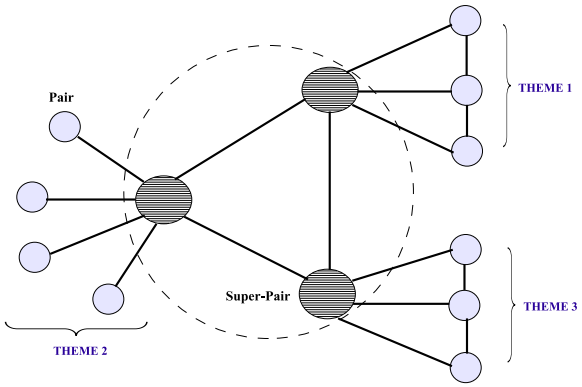

Figure 3. *Réseau Super-pair sémantique organisé par thèmes.*

3.3. Architecture d'un pair

Chaque pair dispose essentiellement des composantes suivantes (figure 4) :

Source de données : Chaque pair dispose d'un système pemettant de gérer ses données qui peuvent être : bases de données objet ou relationnelle ou document XML et d'un langage d'interrogation (OQL, SQL ou XQuery) en rapport avec son modèle de données lui autorisant à fonctionner seul indépendamment des autres pairs.

Interface graphique utilisateur : L'interface utilisateur permet à un pair d'importer des données, mais aussi de formuler plus facilement des requêtes locales sur ses données ou globales dans l'ensemble du réseau.

Réseau sémantique local : Les données publiées par le pair sont abstraites sous forme d'un réseau sémantique ($sGraph$) avec une annotation des nœuds par un ensemble de mots-clés issus des schémas et ajoutés par les concepteurs de ces derniers. Ce modèle interne a pour but de venir à bout de l'hétérogénéité syntaxique des pairs pour faciliter la découverte des correspondances sémantiques entre domaines.

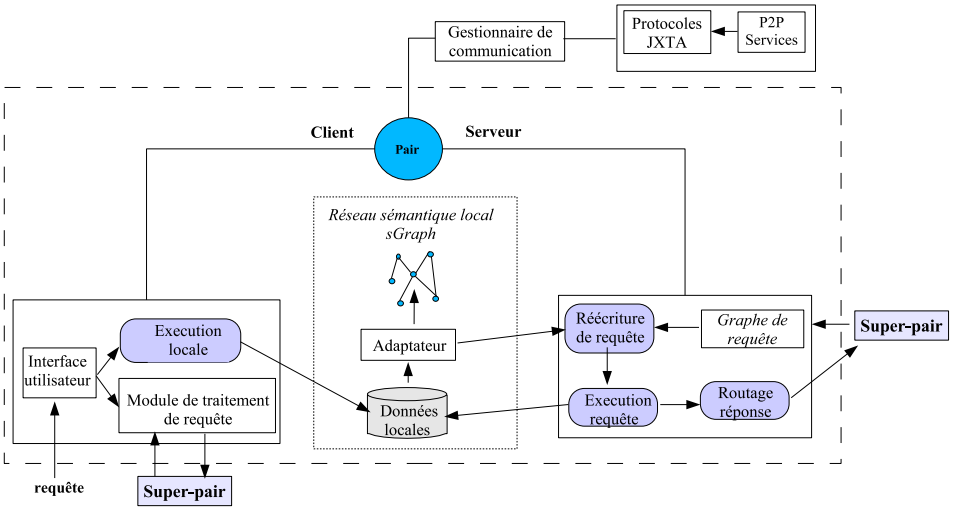

Figure 4. *Architecture d'un pair*

Adaptateur : Il a en charge l'expression du schéma du pair en modèle interne mais aussi la conversion des graphes des requêtes arrivant vers le pair en langage de requête conforme avec le modèle de données du pair.

Gestionnaire de communication : La communication entre les pairs du système est assurée par le projet *Open Source* JXTA de Sun[17]. JXTA définit un réseau générique permettant de construire une variété de réseaux Pair-à-Pair tout en étant indépendant de la plateforme, des langages de programmation (C ou Java) des systèmes (Microsoft Windows, Unix), des définitions de services (RMI, WSDL) et des protocoles réseaux (TCP/IP ou Bluetooth).

3.4. Architecture d'un super-pair

Chaque super-pair contient les composantes suivantes (figure 5) :

Réseau sémantique du domaine : C'est un graphe (appelé $sGraph$) reflétant la structuration des données du domaine dont le super-pair a la responsabilité.

Gestionnaire de correspondances : Il est chargé de prendre les modèles internes des pairs du domaine et ceux des autres super-pairs afin de générer les matrices de correspondance établissant les liens sémantiques entre les éléments de ces réseaux sémantiques et celui du super-pair.

Matrices de correspondance : Elles stockent les correspondances trouvées par le gestionnaire de correspondances et sont maintenues par ce dernier. Elles sont de deux

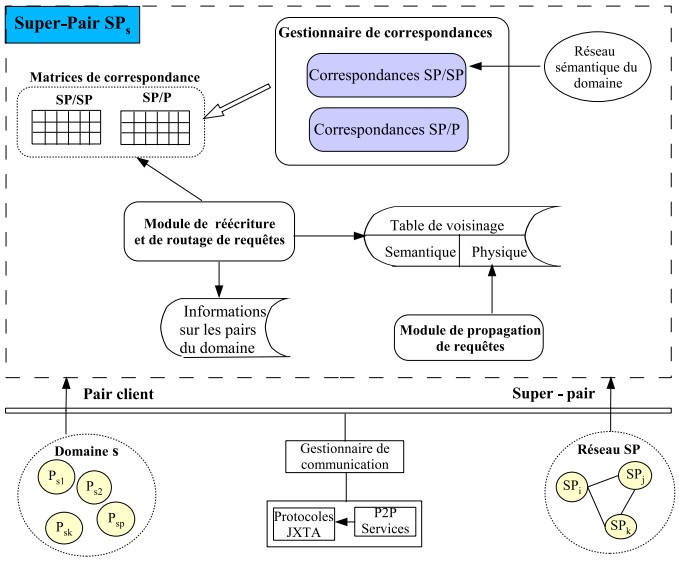

Figure 5. *Architecture d'un super-pair*

types : Super-pair/Super-pair (SP/SP) contenant les correspondances entres les Super-pairs responsables de deux domaines donnés et Super-pair/Pair (SP/P) contenant les correspondances entre un super-pair et les pairs de son domaine.

Module de récriture et de routage de requêtes : Cette composante est chargée de réecrire les requêtes et de les router vers les pairs et super-pairs concernés en accord avec les matrices de correspondance. Etant donné une requête arrivant au super-pair, il génère les sous-requêtes sous forme de graphe pour les Adaptateurs des pairs, puis la fait suivre aux super-pairs liés.

Module de propagation de requête : Il permet de propager les requêtes reçues par le super-pair aux super-pairs voisins en s'appuyant sur la table de voisinage physique.

Module de communication : Comme pour le pair, la communication est assurée par JXTA de Sun[17].

4. Médiation sémantique de données dans SenPeer

4.1. Niveaux de médiation

La médiation a pour but principal de faciliter le partage de données entre pairs. Elle

passe par l'établissement d'un ensemble de correspondances sémantiques entre les schémas des données, correspondances qui seront ensuite utilisées pour la réécriture des requêtes. Nous distinguons deux niveaux de médiation dans SenPeer :

– médiation entre les pairs d'un domaine car les schémas de ces derniers ont été conçus par des experts différents ;

– médiation entre super-pairs, puique les domaines ne sont pas disjoints.

Par ailleurs, la diversité des données publiées en termes de modèles de données constitue une difficulté supplémentaire pour la génération des correspondances. En effet aucun des modèles de données considérés n'est assez expressif pour représenter les autres. De la même façon que P. Berstein[4], nous pensons qu'une représentation pivot interne indépendante des modèles des sources de données des pairs est nécessaire pour faciliter la découverte des correspondances sémantiques. Les schémas des sources sont représentés dans un formalisme interne sous forme de réseau sémantique que nous appelons $sGraph$ (*semantic Graph*) tout en préservant les relations structurelles entre les objets spécifiques dans les langages des schémas. De plus cette structure est enrichie sémantiquement grâce à un ensemble de mots-clés ajoutées à la conception des schémas. La représentation du schéma de la source en $sGraph$ se fait grâce à un adaptateur lié à son modèle de données.

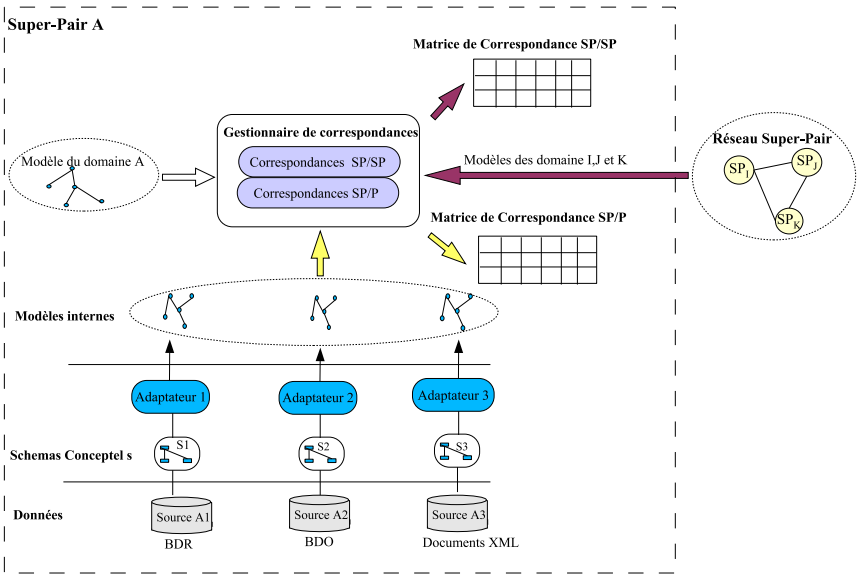

Figure 6. *Une architecture de médiation à deux niveaux.*

4.2. Correspondances sémantiques intra-domaine

Un pair sollicitant le partage de données envoie une méta-requête MQ, décrivant son domaine par un ensemble de mots-clés, au réseau de super-pairs dans le but de trouver le super-pair responsable de son domaine. Lors de la connection d'un pair à un super-pair les informations concernant l'*id* du pair, sa bande bassande, son adresse IP, etc. sont enregistrées dans une table localisée au niveau du super-pair et contenant les informations sur les pairs du domaine. Ensuite le pair est indexé par le super-pair en établissant des correspondances sémantiques (déduites grâce à des mesures de similarité) entre le modèle interne du pair et le modèle du super-pair exprimé lui aussi sous forme de réseau sémantique. Les correspondances trouvées sont ajoutées dans une matrice de correspondance SP/P indiquant les liens existant entre le super-pair et les pairs déjà indexés. Cette matrice de correspondance, stockée au niveau du super-pair, est utilisée plus tard pour la réécriture des requêtes concernant le domaine. Quand un pair quitte le réseau, toutes les références le concernant dans la matrice de correspondance sont supprimées.
Cette approche de génération de correspondances sémantiques est flexible et automatique car ne nécessitant pas l'intervention humaine. De plus elle est guidée par des mots-clés qui constituent un enrichissement sémantique des schémas.

4.3. Correspondances sémantiques inter-domaines

A l'arrivée d'un nouveau super-pair, ce dernier envoie aussi une méta-requête décrivant ses centres d'intêret au réseau de super-pairs dans le but de trouver les super-pairs dont les domaines recoupent le sien. De la même façon les correspondances entre super-pairs sont stockées dans une matrice de correspondance Super-pair/Super-pair (SP/SP). Chaque super-pair abrite un tel type de matrice. A la connection d'un super-pair, ce dernier envoie à ses voisins son modèle interne qui est en quelque sorte son expertise. Ces derniers peuvent alors calculer l'affinité entre leurs domaines et celui du super-pair initial. En fonction de leurs affinités, ils peuvent accepter les correspondances trouvées en les stockant dans leurs matrices de correspondance et propager la requête à leurs voisins. En cas d'acceptation du partage, ils envoient les correspondances trouvées au super-pair initial qui les ajoute à son tour dans sa matrice de correspondance. Il s'en suit l'établissement d'un réseau super-pair sémantique à coté du réseau physique déduit à partir de la matrice de correspondance et qui va être utile pour le routage des requêtes hors du domaine.

Nous présentons dans la partie suivante le modèle pivot interne qui traite de l'hétérogénéité des modèles de données.

4.4. Structure du modèle interne

Le modèle interne est un formalisme permettant de supporter la diversité des modèles de données et de faciliter le processus de découverte de correspondances sémantiques.

Soit $S = \{s_1, \ldots, s_q\}$ le schéma d'un pair dont les élements s_i peuvent être des tables, des colonnes, des éléments ou des attributs XML. Le réseau sémantique correspondant est un graphe orienté étiqueté et acyclique $sGraph =< G(N, E), S, SR, \mu, \delta >$ avec :

$N = \{n_1, \ldots, n_p\}$ ensemble des nœuds correspondants aux éléments figurant dans les schémas. Formellement l'étiquette d'un nœud $n_i \in N$ est définie par la fonction μ qui fait correspondre à n_i un élément $s_i \in S$:

$$\begin{array}{lccl} \mu: & N & \rightarrow & S \\ & n_i & \rightarrow & s_i = \mu(n_i) \end{array}$$

$E = \{e_1, \ldots, e_k\}$ ensemble d'arcs. L'étiquette d'un arc $e_i = (n_1, n_2) \in E$ est donnée par la fonction δ qui fait correspondre e_i à une chaîne r_i appartenant à un ensemble SR de relation sémantiques comme suit :

$$\begin{array}{lccl} \delta: & E \subseteq N \times N & \rightarrow & SR \\ & e_i & \rightarrow & r_i = \delta(e_i) \end{array}$$

Les relations sémantiques dans l'ensemble SR sont définies dans le Tableau 1.

Relation	Description
$isA(n_1, n_2)$	n_1 est une spécialisation n_2.
$partOf(n_1, n_2)$	n_2 agrège n_1. Elle permet de regrouper des élements. Par exemple une clé concaténée agrège les colonnes d'une table.
$contains(n_1, n_2)$	n_1 est un conteneur pour n_2. Par exemple une base de données contient des tables qui contiennent elles mêmes des colonnes.
$typeOf(n_1, n_2)$	n_1 est de type n_2.
$associates(n_1, n_2)$	n_1 est associé à n_2, mais rien n'est dit sur la sémantique de cette relation. Par exemple une table est associée à une autre.
$dataType(n_1, n_2)$	n_1 a pour type de données n_2.
$sourceType(n_1, n_2)$	n_1 a pour modèle de données n_2.
$hasId(n_1, n_2)$	n_1 a pour identificateur n_2.
$references(n_1, n_2)$	n_1 fait référence à n_2. Par exemple une colonne clé étrangère fait référence à une colonne clé primaire.

Tableau 1. *Description des relations sémantiques entre deux nœuds d'un $sGraph$.*

De plus, à chaque nœud n_1 est associé un ensemble de mots-clés $syn(n_1)$ (essentiellement des synonymes issus des schémas et ajoutés par leurs concepteurs) et destinés à guider le processus de découverte des correspondances sémantiques. Ces mots-clés permettent donc de fournir plus d'information linguistique sur l'élément correspondant.

Certains nœuds comme les nœuds représentant des types ne disposent pas de mots-clés associés puisque leur similarité est plus facile à calculer et est prédéfinie. Notons aussi que le nom du nœud fait partie de l'ensemble de ses synonymes. En revenant sur l'exemple de la vallée du fleuve sénégal, la figure 7 constitue une partie d'un modèle interne exprimé dans le formalisme $sGraph$ et représentant une source de données sur les types de cultures pratiquées sur les sols de la vallée. Par souci de lisibilité nous avons volontairement omis les mots-clés associés à certains nœuds.

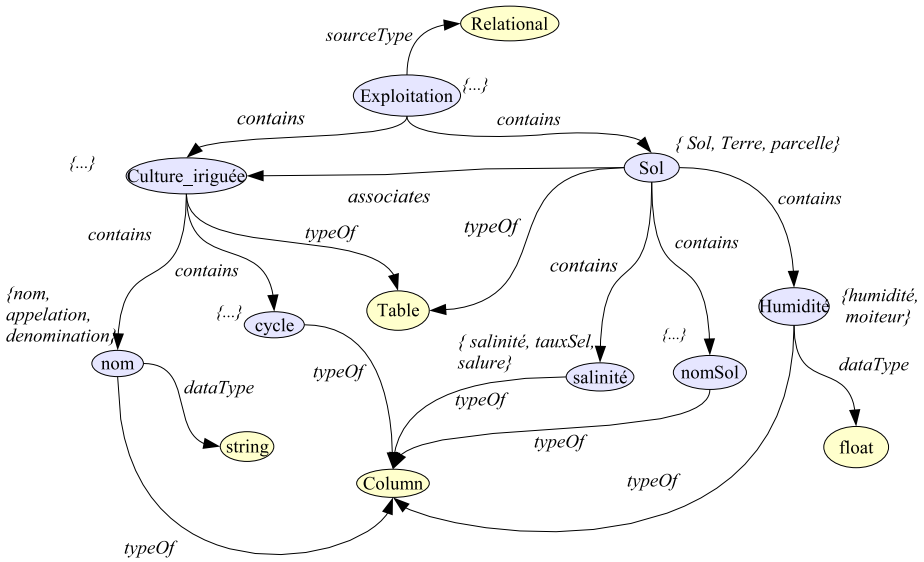

Figure 7. *$sGraph$ partiel sur les cultures pratiquées sur les sols du bassin du fleuve.*

La structure d'un $sGraph$ quelconque peut être décrite formellement, dans un formalisme orienté objet avec la syntaxe BNF (figure 8).

$< SNode >$: := class **{**
$name$: <*word*>
$synonym$: {}|{<*syn-set*>}
$typeOf$: <*typeNode*>
$sourceType$: <*sourceTypeNode*>
$dataType$: <*dataTypeNode*>
isA : <*isA*>
$contains$: <*contains*>
$partOf$: <*partOf*>
$associates$: <*associates*>
$hasId$: <*hasId*>
$references$: <*references*> **}**

<*syn-set*> : := <*word*> | <*word*>,<*synset*>
<*isA*> : := {} | {$< SNode >$}
<*typeNode*> : :={} | $Table$ | $Column$ | $Element$ | $Attribute$ | $Class$ | PK | FK | ID | $IDREF$
<*dataTypeNode*> : :={} | $Integer$ | $Float$ | $String$
<*sourceTypeNode*> : :={} | $RelationalDB$ | $ObjectDB$ | $XMLDocument$
<*contains*> : :={} | {$< SNode >$}
<*partOf*> : :={} | {$< SNode >$}
<*associates*> : := {} | {$< SNode >$}
<*hasId*> : := {} | {$< SNode >$}
<*references*> : := {} | {$< SNode >$}

Figure 8. *Définition de la classe représentant un nœud du modèle interne* $sGraph$.

La partie suivante introduit les mesures de similarité permettant d'établir les correspondances entre les éléments de deux modèles représentés avec le formalisme $sGraph$.

4.5. Mesures de similarités entre nœuds de modèles quelconques

4.5.1. Similarité globale

Nous utilisons un processus de découverte de correspondances basé sur le modèle pivot. La génération des correspondances entre éléments passe par la définition d'une mesure de similarité entre ces derniers. Cette mesure de similarité est composite et dépend de l'affinité linguistique et structurelle des éléments figurant dans les schémas[28][8][5]. La similarité entre deux nœuds appartenant à deux $sGraph$ différents est fonction de :

– leurs descriptions sémantiques avec des mots-clés et de leurs types ;

– leurs relations sémantiques avec les autres concepts (voisinage).

La fonction de similarité $Sim(n_1, n_2)$ de deux nœuds n_1 et n_2 appartenant à deux $sGraph$ SG_1 et SG_2 est une somme pondérée des similarités linguistiques $S_l(n_1, n_2)$ et de voisinage $S_v(n_1, n_2)$ de ces deux nœuds, soit :

$$Sim(n_1, n_2) = \lambda_l.S_l(n_1, n_2) + \lambda_v.S_v(n_1, n_2) \quad \text{avec} \quad \lambda_l, \lambda_v \geq 0 \text{ et } \lambda_l + \lambda_v = 1 \quad (1)$$

Les coefficients λ_l et λ_v sont les poids associés respectivement aux synonymes et aux relations de voisinage des concepts.

4.5.2. Similarité linguistique

La similarité linguistique $S_l(n_1, n_2)$ de deux éléments n_1 et n_2 permet d'évaluer l'affinité linguistique entre les deux ensembles de synonymes des deux éléments. Elle constitue une somme pondérée de la similarité des deux ensembles de synonymes $S_s(n_1, n_2)$ et de leur similarité de type $S_t(n_1, n_2)$:

$$S_l(n_1, n_2) = \omega_s.S_s(n_1, n_2) + \omega_t.S_t(n_1, n_2) \quad \text{avec} \quad \omega_s, \omega_t \geq 0 \text{ et } \omega_s + \omega_t = 1 \quad (2)$$

Nous considérons que les similarités de types sont déjà connues et définies dans une table et qu'elles sont comprises dans l'intervalle $[0, 1]$.

La similarité globale entre les deux ensembles de synonymes dépend de la comparaison lexicale entre les différents mots de ces deux ensembles. La similarité lexicale entre deux mots p et q est basée sur la mesure de similarité SM lexicale proposée par [22], elle même basée sur la distance ed de Levenstein[19].

$$SM(p, q) = max\Big(0, \frac{min(|p|, |q|) - ed(p, q)}{min(|p|, |q|)}\Big) \in [0, 1] \quad (3)$$

Cette fonction calcule la similarité des chaînes p et q en tenant compte du nombre d'actions atomiques (ajout, suppression de caractère) nécessaires pour transformer l'une des chaînes de caractères en l'autre chaîne. Elle est fonction du rapport entre le nombre de ces opérations d'édition et de la longueur de la plus courte des deux chaînes p et q à comparer.

Prenons par exemple les deux entrées lexicales $cultureirrigue$ et $culture\_irriguee$. Puisque $ed(cultureirrigue, culture\_irriguee) = 1$ on a alors

$SM(cultureirrigue, culture\_irriguee) = \frac{13}{14}$

La similarité $S_s(n_1, n_2)$ entre les ensembles de synomymes est basée sur la mesure proposée par Tversky[30]. Pour comparer deux ensembles d'éléments A et B, Tversky se base sur les intuitions suivantes :

– La similarité entre A et B dépend de ce qu'ils ont en commun. Plus ils ont de choses en commun, plus leur similarité sera élevée.

– La similarité entre A et B dépend de leurs différences. Plus ils ont de différences plus leur similarité sera faible.

La mesure de Tvsersky est donnée par la formule ci-dessous dans laquelle $A \cap B$ représente l'intersection des ensembles A et B et $A \setminus B$ leur différence, $|A|$ le cardinal de A et α une valeur indiquant l'importance des caractéristiques communes et non communes.

$$S(a,b) = \frac{|A \cap B|}{|A \cap B| + \alpha|A \setminus B| + (1-\alpha)|B \setminus A|} \quad \text{avec} \quad 0 \leq \alpha \leq 1 \tag{4}$$

Cette mesure est pratique, mais telle qu'elle, elle n'est pas tout à fait adaptée à notre cas puisqu'elle ne prend en compte qu'un appariement exact des termes. Par exemple les termes $Culture\_irrigue$ et $CultureIrrigue$ sont considérés comme différents alors qu'ils sont, à notre avis, équivalents à une transformation près. Nous nous proposons de l'étendre en faisant un appariement flou des termes présents dans les ensembles de synonymes, plutôt qu'un appariement exact de ces termes. Ainsi, étant donné A et B deux ensembles d'entrées lexicales, nous définissons l'intersection floue $A \cap_f B$ et la différence floue $A -_f B$ de ces deux ensembles comme suit :

$$\begin{aligned} A \cap_f B &= \{a \in A / \max_{b \in B} SM(a,b) > \epsilon_{acc}\} \\ A -_f B &= \{a \in A / \max_{b \in B} SM(a,b) \leq \epsilon_{acc}\} \end{aligned} \tag{5}$$

ϵ_{acc} est le seuil au dessus duquel la similarité calculée est considérée comme acceptable. La similarité $S_s(n_1, n_2)$ entre deux ensembles de synonymes $syn(n_1)$ et $syn(n_2)$ de deux nœuds n_1 et n_2 s'exprime alors comme suit :

$$S_s(n_1,n_2) = \frac{|syn(n_1) \cap_f syn(n_2)|}{|syn(n_1) \cap_f syn(n_2)| + \alpha|syn(n_1) -_f syn(n_2)| + (1-\alpha)|syn(n_2) -_f syn(n_1)|} \tag{6}$$

4.5.3. Similarité de voisinage sémantique

La similarité globale de deux nœuds dépend aussi de leur contexte d'apparition dans les deux $sGraph$, donc de leur voisinage sémantique. Elle se base sur l'intuition selon laquelle des nœuds *similaires* son liés à des nœuds *similaires*.

Soit $SG =< G(N,E), S, SR, \mu, \delta >$ un $sGraph$ et $n_1 \in SG$, le voisinage sémantique $V(n_1)$ de n_1 est constitué de l'ensemble des nœuds ayant un lien sémantique direct avec n_1, soit : $V(n_1) = \{n_2 \in InternalNode(SG) / \exists\, r \in SR \wedge r(n_1, n_2)\}$.

$InternalNode(SG)$ est constitué de l'ensemble des nœuds du $sGraph$ SG sauf les nœuds représentant des types.

La similarité de voisinage sémantique compare les nœuds dans les voisinages en se basant sur leur similarité linguistique. Cette similarité concerne les nœuds internes des graphes. Elle dépend de la taille et de l'intersection floue de leurs voisinages. En faisant un appariement flou des voisinages, nous obtenons la similarité suivante :

$$S_v(n_1,n_2) = \frac{|\{x \in V(n_1) / \exists y \in V(n_2) \wedge Sim(x,y) > \epsilon_{acc}\}|}{|V(n_1) \cup V(n_2)|} \tag{7}$$

Notons que la similarité de voisinage n'est calculée qu'entre les nœuds internes des $sGraph$ car celle entre les nœuds externes (qui sont les nœuds représentant les types) est déjà prédéfinie dans une table.

4.5.4. Génération des matrices de correspondance

A chaque domaine sont associés deux types de matrices de correspondance localisées au niveau du super-pair.

– matrice de correspondance **Super-pair/Pair (SP/P)** : Pour chaque pair du domaine le super-pair maintient une matrice de correspondance de ce type contenant les correspondances entre le vocabulaire du pair et celui du domaine.

– matrice de correspondance **Super-pair/Super-pair (SP/SP)** : Pour chaque Super-pair dans son voisinage sémantique le Super-pair dispose d'une matrice indiquant les correspondances entre domaines.

Les correspondances entre les modèles internes des pairs et le modèle d'un super-pair sont déduites en évaluant les mesures de similarité entre les éléments de ces modèles internes et ceux du modèle interne du super-pair. La matrice de correspondance SP/P pour un super-pair donné contient en colonne les nœuds du $sGraph$ du domaine et en ligne ceux des $sGraph$ des pairs.

	ng_1	ng_2	$\ldots$	ng_n
nl_1				$\cong$
nl_2	$\cong$			
$\ldots$				
nl_m			$\cong$	

Tableau 2. *Matrice de correspondance Super-pair/pair*

Un élément $MatSPP(n_g, n_l)$ de cette matrice indique la façon dont les nœuds correspondants sont liés. Nos travaux actuels s'orientent vers les correspondances sémantiques exactes et directes (équivalence $\cong$). Dans ce cas, une correspondance sémantique est un triplet $< n_g, n_l, \cong >$ avec n_g et n_l des éléments respectifs des deux $sGraph$ concernés. Dans le cas où il n'y a pas de correspondance la relation entre n_g et n_l est évaluée à $null$.

L'algorithme ci dessous décrit le principe de construction de la matrice de correspondance SP/P. Pour chaque couple de nœuds des $sGraph$, les mesures de similarités sont évaluées et la correspondance initialisée par défaut à $null$ (1-8). Ensuite (9-16) pour chaque nœuds n_g du $sGraph$ $sGSP$ du super-pair, on cherche l'élément $nlmax$ qui lui est le plus similaire dans le $sGraph$ local sGP. Si cette similarité est supérieure au seuil admis ϵ_{acc}, alors on a trouvé une équivalence $n_g \cong nlmax$.

Algorithme 1 Génération de la Matrice de correspondance Super-pair /pair

Entree : $sGSP$ et sGP $sGraph$ du super-pair et du pair.
Sortie : $MatSPP$: Matrice de correspondance des deux $sGraph$.
Pour tout $n_g \in NoeudInterne(sGSP)$ **Faire**
  Pour tout $n_l \in NoeudInterne(sGP)$ **Faire**
    $x \leftarrow \omega_s \times sim_s(n_g, n_l) + \omega_t \times sim_t(n_g, n_l)$
    $y \leftarrow sim_v(n_g, n_l)$
    $sim(n_g, n_l) \leftarrow \alpha \times x + (1 - \alpha) \times y$
    $add(MatSPP, n_g, n_l, null)$
  Fin Pour
Fin Pour
Pour tout $n_l \in NoeudInterne(sGP)$ **Faire**
  $ngmax \leftarrow \arg \max_{n_g \in sGSP} sim(n_l, n_g)$
  Si $sim(n_l, ngmax) > \epsilon_{acc}$ **Alors**
    $add(MatSPP, ngmax, n_g, \cong)$
  Fin Si
Fin Pour

Algorithme 2 Génération de la Matrice de correspondance Super-Pair/Super-Pair

Entrée : $sGSP1$ et $sGSP2$ $sGraph$ respectif de $SP1$ et de $SP2$.
Sortie : $MatSPSP$: Matrice de correspondance des deux $sGraph$
Pour tout $n_g^1 \in NoeudInterne(sGSP1)$ **Faire**
  Pour tout $n_g^2 \in NoeudInterne(sGSP2)$ **Faire**
    $x \leftarrow \omega_s \times sim_s(n_g^1, n_g^2) + \omega_t sim_t \times sim_t(n_g^1, n_g^2)$
    $y \leftarrow sim_v(n_g^1, n_g^2)$
    $sim(n_g^1, n_g^2) \leftarrow \alpha \times x + (1 - \alpha) \times y$
    $add(MatSPP, n_g^1, n_g^2, null)$
    Si $sim(n_g^1, n_g^2) > \epsilon_{acc}$ **Alors**
      $add(MatSPP, n_g^1, n_g^2, \cong)$
    Fin Si
  Fin Pour
Fin Pour

5. Illustration

Dans cette partie nous illustrons le processus de découverte des correspondances sémantiques, mais aussi comment ces dernières sont utilisées pour reformuler une requête d'un pair vers un autre. Considérons le scénario de la figure 9 dans lequel le système de partage de données du fleuve concerne les domaines *Agriculture* et *Pédologie*. La connaissance de chaque domaine est décrite par un $sGraph$ dans lequel nous avons omis, pour des raisons de lisibilité, les étiquettes des arcs et les nœuds externes représentant les types. La connaissance du domaine *Agriculture* concerne les sols et les cultures pratiquées sur ces sols tandis que celle du domaine *Pédologie* concerne les sols. De plus le domaine

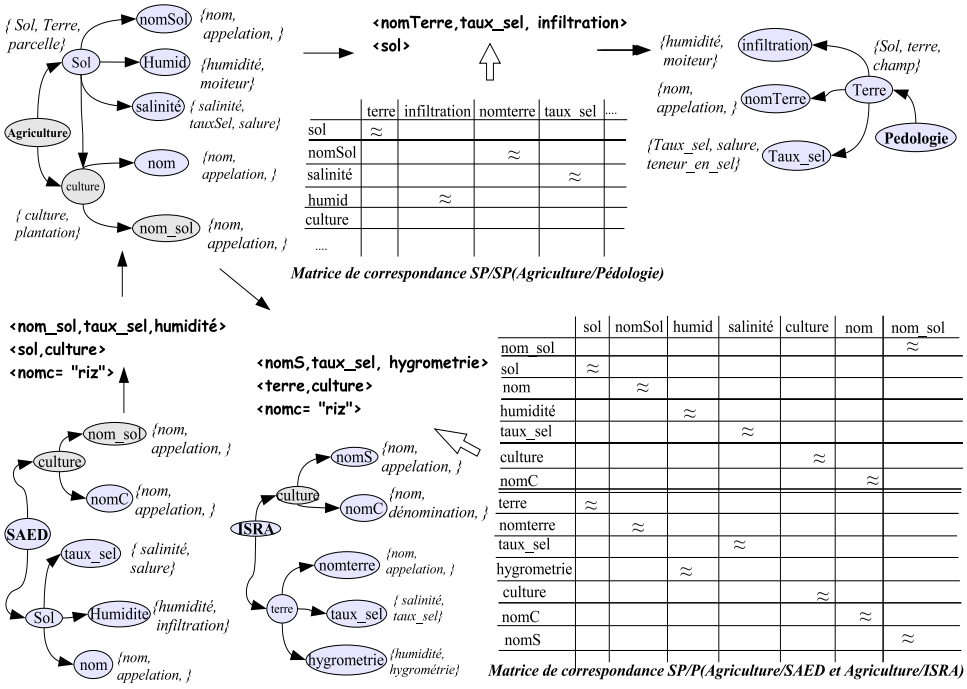

Figure 9. *Médiation de données des domaines Agriculture et Pédologie. ISRA(Institut Sénégalais de Recherche Agronomique), SAED (Société d'Aménagement et d'Exploitation du Delta).*

Agriculture contient deux pairs, SAED (Société d'Aménagement et d'Exploitation des terres du Delta) et ISRA (Institut Sénégalais de Recherche Agronomique) publiant aussi leurs données sous forme de $sGraph$. Nous n'avons pas matérialisé les pairs du domaine *Pédologie* par souci de simplicité. Les deux domaines ne sont pas disjoints puisque aussi bien les experts de l'agriculture que ceux de la pédologie manipulent des données sur les sols. Les correspondances entre ces domaines sont alors stockées dans la matrice de correspondance SP/SP (Agriculture/Pédologie). Par ailleurs les pairs SAED et ISRA décrivent des données sur les sols et les cultures mais avec des vocabulaires différents de celui de leur domaine. Les correspondances déduites sont stockées dans la matrice de correspondance SP/P (Agriculture/SAED et Agriculture/ISRA).

Toutes ces correspondances sont déduites grâce aux mesures de similarité présentées dans la partie précédente. Par exemple, pour établir la correspondance $< taux\_sel, salinite, \cong >$ de la matrice de correspondance SP/SP (Agriculture/Pédologie), nous calculons la similarité $Sim(taux\_sel, salinite)$ entre les nœuds $taux\_sel$ et $salinite$ (pour $\lambda_l = \lambda_v = \omega_s = \omega_t = 0.5$) comme suit :

$Sim(taux\_sel, salinite) = 0.5 \times S_l(taux\_sel, salinite) + 0.5 \times S_v(taux\_sel, salinite)$
avec $S_l(taux\_sel, salinite) = 0.5 \times S_s(taux\_sel, salinite) + 0.5 \times S_t(taux\_sel, salinite)$.

En considérant que $taux\_sel$ et $salinite$ sont de même type, la similarité de type est alors maximale et vaut 1. En prenant par exemple $\alpha = 0.5$ dans l'équation (6), la similarité de synonymes s'exprime comme suit :

$$S_s(taux\_sel, salinite) = \frac{|\{tauxsel, salure\}|}{|\{tauxsel, salure\}| + 0.5|\{teneur\_en\_sel\}| + 0.5|\{salinite\}|} = \frac{2}{3}.$$

Par conséquent $S_l(taux\_sel, salinite) = 0.5 \times \frac{2}{3} + 0.5 \times 1 = 0.83$

La similarité de voisinage $Sim_v(taux\_sel, salinite)$ s'exprime ainsi :

$$S_v(taux\_sel, salinite) = \frac{|\{x \in V(taux\_sel) / \exists y \in V(salinite) \wedge Sim(x,y) > \epsilon_{acc}\}|}{|V(taux\_sel) \cup V(salinite)|}$$

On a $V(salinite) = \{sol\}$, $V(taux\_sel) = \{terre\}$, $syn(sol) = \{sol, terre, parcelle\}$ et $syn(terre) = \{sol, terre, champ\}$. On montre que $S_s(sol, terre) = 0.5$ et $S_t(sol, terre) = 1$. On trouve alors : $S_l(sol, terre) = 0.5 \times 0.5 + O.5 \times 1 = 0.75 > \epsilon_{acc} = 0.6$. Par substitution $S_v = \frac{1}{2}$. La similarité globale vaut alors : $Sim(taux\_sel, salinite) = 0.5 \times 0.83 + 0.5 \times 0.5 = 0.67$.

Supposons maintenant que le pair SAED exprime la requête suivante :

Quels sont le nom, le taux de salinité et l'humidité des sols sur lesquels le riz est cultivé ?

Le pair SAED ne connaissant que son schéma, la requête est exprimée sur son propre schéma avec son propre vocabulaire :

(R1)
$< nom\_sol, taux\_sel, humidite >$
$< sol, culture >$
$< nomc = "riz" >$

La requête (R1) est ensuite envoyée au super-pair du domaine *Agriculture*. Dans un premier temps elle est réécrite au sein du domaine. Ainsi d'après la matrice de correspondance SP/P (Agriculture/SAED et Agriculture/ISRA), la requête est réécrite avec le vocabulaire du pair ISRA comme suit :

(R2)
$< nomS, taux\_sel, hygrometrie >$
$< terre, culture >$
$< nomc = "riz" >$.

La requête (R2) peut maintenant être reformulée avec le langage d'interrogation du pair ISRA et les résultats envoyés au super-pair *Agriculture*.

D'autre part, la matrice de correspondance SP/SP (Agriculture/Pédologie) permet de réecrire la requête (R1), avec le vocabulaire du domaine *Pédologie*, en une requête (R3) :

(R3)
$< nomTerre, taux\_sel, infiltration >$
$< sol >$

Celle-ci est évaluée dans le domaine et les résultats sont ensuite envoyés au Super-pair *Agriculture* qui peut enfin composer les résultats et les envoyer au pair SAED ayant initié la requête.

6. Travaux connexes

A ce jour plusieurs projets scientifiques ont permis la mise en place d'un certain nombre de PDMS dont le but principal est le partage de sources de données hétérogènes et distribuées à grande échelle. Ces systèmes reposent principalement sur les connaissances figurant dans les schémas. Nous pouvons les subdiviser en différentes catégories selon les deux critères suivants :

– une ontologie est commune à tous les nœuds ;

– chaque nœud déclare sa propre ontologie ;

– la connexion nécessite ou pas la mise en place de correspondances sémantiques.

	Sans correspondances	Avec correspondances
Ontologie commune	Bibster Edutella	
Plusieurs ontologies	PeerDB XPeer	Piazza PEPSINT Hyperion SEWASIE

Tableau 3. *Classification de quelques PDMS existant*

Dans la suite nous présentons quelques PDMS suivant les critères énumérés ci-dessus.

Bibster[11] permet l'échange de données bibliographiques entre chercheurs. Il est basé sur deux ontologies permettant de structurer automatiquement les données des pairs, se passant ainsi du problème de découverte des correspondances. Les ontologies interviennent dans le stockage des données, la reformulation et le routage des requêtes mais aussi dans la présentation des résultats. La sélection des pairs pertinents est basée sur des description d'expertise des pairs, induisant la formation d'un réseau Pair-à-Pair sémantique indépendant de la topologie Pair-à-Pair existante. Les requêtes sont formulées en termes des deux ontologies et routées de façon intelligente en fonction des descriptions d'expertises connues par le pair actuel. Cependant Bibster ne constitue pas une archi-

tecture flexible car fournissant une application spécifique ne fonctionnant que pour le domaine bibliographique. De ce fait donc avec le changement des méta-données, la réutilisabilité est compromise.

Edutella[25] fournit une infrastructure Pair-à-Pair supportant les méta-données RDF. Il est basé sur le framework JXTA et permet l'échange de données d'un domaine commun qui est ici le domaine éducatif. Le réseau Edutella est basé sur une topologie super-pair qui est similaire à SenPeer et dans laquelle les pairs sont organisés en hyper-cube pour router les requêtes. Les requêtes sont échangées dans un format commun d'échange de requêtes appelé RDF-QEL. A noter que Edutella ne supporte pas directement les sources de données XML , et donc les sources de données RDF doivent être sérialisées dans un format XML. Contrairement à SenPeer le modèle de données est commun.

PeerDB[26] 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) similairement à SenPeer. Contrairement à SenPeer il n'y a pas de processus classique de découverte de correspondances entre les schémas entre pairs. 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 de requêtes. L'utilisateur doit décider quelles requêtes ont un sens et quelles requêtes doivent être exécutées. Notons aussi que la version actuelle de PeerDB ne supporte que les données relationnelles

XPeer [29] est basé sur une architecture hybride, comme SenPeer, permettant le partage de données XML concernant n'importe quel domaine. L'intégration se fait sans schéma global, ce qui réduit considérablement les tâches d'administration. Chaque pair exporte la description des données à partager sous forme d'une arborescence qui est automatiquement inférée des données. Les pairs sont organisés logiquement en groupes sur la base de critères de similarité de schémas. Le schéma d'un super-pair est construit sans activité d'intégration, ce qui fait que l'assistance humaine n'est pas nécessaire. Les requêtes sont écrites dans un sous ensemble de XQuery. Cependant l'applicabilité de XPeer concerne seulement les situations pour lesquelles la découverte des correspondances peut être évitée ou être faite en dehors du système Pair-à-Pair.

Piazza[12] 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 interéssés 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 la reformulation des requêtes est faite par un nœud central.

Hyperion[2] propose une architecture pour un PDMS dans lequel la publication de chaque pair est une base de données relationnelle et dans lequel il n'y a pas de schéma global. L 'échange d'informations entre pairs est possible grâce à la définition de tables de correspondance comme dans SenPeer et d'expressions de correspondances qui stockent les correspondances sémantiques entre éléments des schémas des pairs. Un gestionnaire de requêtes utilise les tables et les expressions de correspondances pour réécrire une requête exprimée sur le schéma d'un pair spécifique sur les schémas des pairs liés. Cependant, les tables de correspondance, bien que plus faciles à établir qu'un processus complet de découverte de correspondances entre schémas, sont jusqu'ici créées manuellement par des spécialistes du domaine ce qui peut être coûteux en temps. Contrairement à SenPeer, seules les sources de données relationnelles sont supportées par Hyperion.

APPA (ATLAS Peer-to-Peer Architecture)[1] est un système Pair-à-Pair de gestion de données distribuées ayant une architecture indépendante du type de réseau Pair-à-Pair (non structuré, DHT, Super-pair, etc.). Cette architecture repose sur des services organisés par niveaux (service de réseau Pair-à-Pair, services de base, services avancés). Les services avancés permettent le partage de données sémantiquement riches incluant la gestion des schémas, la réplication, le traitement des requêtes, etc. Les données partagées sont au format XML et le langage d'interrogation est XQuery. De plus chaque pair a la possibilité de manipuler ses données XML localement à travers un adaptateur si besoin. Le partage de données sémantiquement riches se fait de façon décentralisé. APPA suppose que les pairs désireux de partager leurs données s'accordent sur une description de schéma commune (CSD). Le schéma d'un pair est exprimé comme une vue sur le CSD. Les requêtes sont exprimées en terme du schéma local et non du schéma commun. De plus APPA considère que les correspondances entre schémas sont maintenues jusqu'à ce que le partage de données ne soit plus souhaité. Le système APPA est en cours d'implémentation avec le réseau générique JXTA.

PEPSINT[9] permet d'intégrer sémantiquement des sources de données XML et RDF hétérogènes dans une environnement Pair-à-Pair. Le système est basé sur une architecture Pair-à-Pair hybride composé d'un super-pair unique et d'un ensemble de pairs. Le super-pair en question contient une ontologie RDF globale (construite en utilisant l'approche GAV). Les pairs abritent les schémas locaux et les sources de données locales. Cette ontologie sert non seulement de point de contrôle central sur l'ensemble des pairs du réseau

mais aussi de médiateur pour la réécriture des requêtes d'un pair à un autre. De façon analogue à SenPeer, chaque pair connecté au réseau est indexé par le super-pair grâce à un certain nombre de correspondances sémantiques stockées dans une table de correspondance qui indique les équivalences entre éléments des schémas. Les correspondances sont établies grâce à un processus classique d'appariement de schémas. Durant cette opération l'ontologie globale est étendue par intégration des schémas locaux. Les requêtes sont reformulées par le super-pair en utilisant des compositions de correspondances du pair initial vers le super-pair et du super-pair vers les autres pairs. Un défi majeur à relever pour PEPSINT est la tolérance aux fautes. En effet avec l'utilisation d'un super-pair central sollicité pour tout traitement de requêtes le système ne fonctionne plus en cas d'indisponibilité de ce dernier.

SEWASIE[3] se propose de permettre aussi bien l'échange de données struturées, semi-structurées que non structurées. Chaque pair contient une information spécifique à propos des domaines concernés, mais seulement une couche de cette connaissance doit être exportée vers les autres pairs à travers un langage standard pour représenter la structure de ces sources. La connaissance en question est représentée à travers une ontologie. La connection entre les pairs repose alors sur l'échange de méta-données XML. L'intégration est possible grâce à la création d'une vue virtuelle globale des sources de données et d'un certain nombre de correspondances entre cette vue et les sources intégrées. Un gestionnaire de requêtes décompose la requête en tenant compte des correspondances entre la vue virtuelle globale et les sources pouvant répondre à la requête, envoie les requêtes, collecte les résultats et les envoie au pair interrogateur.

7. Conclusion

Nous avons présenté dans cet article le système SenPeer permettant le partage décentralisé de données entre plusieurs experts de différents domaines de compétences publiant des données environnementales liées à la mise en valeur de la vallée du fleuve Sénégal. Les données publiées par les pairs sont basées sur l'un des modèles de données suivants : relationnel, objet ou XML. Chaque pair dispose de sa propre interface d'accès et de son propre langage d'interrogation.

Dans le but d'un partage efficace de ces données, une seule couche est exportée grâce à un réseau sémantique appelé $sGraph$. Ce réseau sémantique constitue un modèle pivot interne qui a pour but de venir à bout de la diversité des modèles de données et de faciliter plus tard la génération des correspondances sémantiques entre pairs. Un enrichissement sémantique des modèles internes grâce à des mots-clés introduits à la conception des schémas guide la découverte des correspondances sémantiques qui sont stockées dans des matrices de correspondance.

L'organisation en domaines sémantiques grâce à la topologie super-pair permet le regroupement des pairs d'un domaine d'intérêt commun. Ceci permet d'une part de dimunier les tâches quant à la découverte des correspondances sémantiques puiqu'elle est faite entre groupes de pairs et non entre n'importe quel couple de pairs du réseau pair-à-pair. D'autre part, cette organisation permet de n'envoyer les requêtes qu'aux pairs susceptibles d'y répondre évitant ainsi la propagation aveugle des requêtes à tous les pairs.

Nos travaux futurs vont consister à :

– trouver un format commun d'échanges de requêtes en se basant sur le modèle pivot interne $sGraph$

– mettre en place le processus d'interrogation et notamment la sélection des pairs pertinents

– définir des algorithmes de décomposition, de réécriture et de routage de requêtes

8. Bibliographie

[1] AKBARINIA R., MARTINS V., PACITTI E., VALDURIEZ P. « Replication and Query Processing in the APPA Data Management System » *Distributed Data & Structures 6 (WDAS) : Records of the 6th International Meeting (Lausanne, Switzerland)* : Carleton Scientific, 2004.

[2] ARENAS M., KANTERE V., KEMENTSIETSIDIS A., KIRINGA I., MILLER R. J. , MYLOPOULOS J., « The Hyperion Project : From Data Integration to Data Coordination », *SIGMOD Record 32(3), page 53-58*, 2003.

[3] BERGAMASCHI S. , GUERRA F. , VINCINI M., « A peer-to-peer information system for the semantic web », *proceedings of the International Workshop on Agents and Peer-to-Peer Computing (AP2PC03)* Melbourne, Australia, July 14.

[4] BERNSTEIN P. A. , MELNIK S., PETROPOULOS M., QUIX C., « Industrial-strength schema matching », *SIGMOD Rec.* vol. 33, n° 4, 2004.

[5] DO H., RAHM E., PETROPOULOS M., QUIX C., « COMA - A system for flexible combination of schema matching approaches », *Proc. 28th Conference on Very Large Databases (VLDB), Hongkong, August*, 2002.

[6] EQUIPE BDISIC. PROJET SIC-WEB SÉNÉGAL, « Compte rendu du Workshop des 10 et 11 juin 2004, Université Gaston Berger de Saint-Louis du Sénégal », 2004.

[8] CASTANO S. , FERRARA A. , MONTANELLI S., QUIX C., « H-MATCH : an Algorithm for Dynamically Matching Ontologies in Peer-based Systems », *Proc. of the 1st VLDB Int. Workshop on Semantic Web and Databases (SWDB 2003) Berlin, Germany, September*, 2003.

[9] CRUZ I. F. , XIAO H. , HSU F. , « Peer-to-Peer Semantic Integration of XML and RDF Data Sources », *Third International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2004), July*, 2004.

[11] HAASE P. , BROEKSTRA J. , EHRIG M., MENKEN M. , MIKA P., PLECHAWSKI M. , PYSZLAK P., SCHNIZLER B., SIEBES R. , STAAB S., TEMPICH C., « Bibster - A Semantics-

Based Bibliographic Peer-to-Peer System », *Proceedings of the International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 9-11* , 2004.

[12] 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.

[16] HERSCHEL S. , HEESE R., « Humboldt Discoverer : A semantic P2P index for PDMS », *International Workshop Data Integration and the Semantic Web, Porto, Portugal, June* , 2005.

[17] JXTA, « http ://www.jxta.org/ ».

[18] KAZAA, « www.kazaa.com ».

[19] LEVENSTEIN A., « Binary Codes Capable of Correcting Deletions, Insertions and Reversals Sov », *Phys. Dohl, Vol.10, P707-710,* 1966.

[20] LUMINEAU N., DOUCET A., GANÇARSKI S., « Thematic Schema Building for Mediation-based P2P Architecture », *International Workshop On Database Interoperability (InterDB2005), Namur, Belgique* , 2005.

[22] MAEDCHE A. , STAAB S., « Measuring Similarity between Ontologies. », *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW), Siguenza, Spain , pages 251-263,* 2002.

[24] NAPSTER, « www.napster.com ».

[25] NEJDL W., WOLF B., QU C., DECKER S., SINTEK M. , NAEVE A., NILSSON M. , PALMER M., RISCH T. , « Edutella : A P2P networking infrastructure based on RDF », *in 11th International World Wide Web Conference(WWW2002), Hawaii, USA , May,* 2002.

[26] NG W. S., OOI B. C., TAN L. , ZHOU A., « PeerDB : A P2P-based System for Distributed DataSharing », *Proceedings of the 19th International Conference on Data Engineering, Bangalore, India, page 633-644,* 2003.

[28] RODRIGUEZ A. , EGENHOFER M. , RUGG R. , « Assessing Semantic Similarities Among Geospatial Feature Class Definitions », *Interoperating Geographic Information Systems, Second International Conference, Interop '99, Zurich, Switzerland, Lecture Notes in Computer Science, Vol. 1580, Springer-Verlag, pp. 189-202, March* 1999.

[29] SARTIANI C. , MANGHI P. , GHELLI G. , CONFORTI G. , « XPeer : A Self-organizing XML P2P Database System », *Proceedings of the First EDBT Workshop on P2P and Databases (P2P&DB 2004), Crete, Greece July,* 2004.

[30] TVERSKY A., EGENHOFER M. , RUGG R. , CONFORTI G. , « Features of similarity », *Psychological Review, 84, 327-352,* 1977.